Appears in
Proceedings of the Tenth Annual Conference on Computational Learning Theory (COLT-97),
Nashville, July 1997.

General convergence results for linear discriminant updates

Adam J. Grove **Nick Littlestone** **Dale Schuurmans***
NEC Research Institute
4 Independence Way
Princeton NJ 08540, USA
{grove,nickl,dale}@research.nj.nec.com

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 much of this class, including both Perceptron and Winnow but also many novel algorithms. Our proof introduces a generic *measure of progress* that seems to capture much of when and how these algorithms converge. Using this measure, we develop a simple general technique for proving mistake bounds, which we apply to the new algorithms as well as existing algorithms. When applied to known algorithms, our technique "automatically" produces close variants of existing proofs (and we generally obtain the known bounds, to within constants)—thus showing, in a certain sense, that these seemingly diverse results are fundamentally isomorphic.

1 Introduction

Many iterative mistake-driven algorithms have been proposed for learning linear discriminant concepts, including the famous *Perceptron* algorithm [Ros62,MP69,DH73] and Littlestone's *Winnow* family of algorithms [Lit88,Lit89,Lit91]. This is an important and well studied collection of algorithms that have very interesting properties and practical applications. In this paper we define a general class of algorithms and provide a unified theoretical analysis, which not only covers both Perceptron and Winnow but also many new algorithms.

The algorithms we discuss represent linear discriminant concepts by keeping a current weight vector $\tilde{w} \in \mathbb{R}^n$ and classifying examples $\tilde{x} \in \mathbb{R}^n$ into classes labeled ± 1 according to the rule $\tilde{x} \mapsto$ *sign*$(\tilde{w} \cdot \tilde{x})$.[1] These algorithms work in an on-line fashion by observing a sequence of training examples $(\tilde{x}_1, y_1), (\tilde{x}_2, y_2), \ldots$, and updating the current weight vector $\tilde{w}$ if and only if they make a mistake on a particular example (i.e., whenever *sign*$(\tilde{w} \cdot \tilde{x}_i) \neq y_i$ for a given example $(\tilde{x}_i, y_i)$). 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; see Section 2.

A striking fact about these algorithms is that they all *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 exist two parallel separating hyperplanes 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 [MP69,DH73]) and Littlestone's proofs for members of the Winnow family [Lit88] [Lit89] 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 first postulates a function of the weight vector, which we call a *measure of progress*, and then proves that it (eventually) makes constant progress towards a solution after each mistake. However, aside from this broad similarity, the proofs

*Primary affiliation: Institute for Research in Cognitive Science, University of Pennsylvania, 3401 Walnut Street, Suite 400A, Philadelphia, PA 19104-6228

[1]Note that often the class of linear discriminant concepts are taken to be those defined by a vector $\tilde{u} \in \mathbb{R}^n$ and a constant $c \in \mathbb{R}$, with the label being $+1$ iff $\tilde{x} \cdot \tilde{u} \geq c$. However our assumption, that $c = 0$, is not restrictive. To cope with $c \neq 0$ one can add an $n+1$'st attribute to all training examples which has the constant value 1. We can imagine similarly extending $\tilde{u}$, by adding an $n+1$'st attribute with value $-c$. This "equivalent" concept has threshold 0 as we require.

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 that includes 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 Perceptron and Winnow as special cases, but also includes many new algorithms that have not been previously investigated.

The central contribution of this paper is the introduction of a general approach for constructing measures of progress that can be applied to quasi-additive algorithms. We outline the motivating ideas of this approach briefly in Section 3. As will be apparent there, our basic approach can be completed using one of several slightly different elaborations. The simplest and most general of these variations is used in Section 4 to prove a general convergence result for quasi-additive algorithms. This result has the benefit of being simple and general, however it has the drawback that the specific mistake bounds it yields can be somewhat loose in particular cases. In later sections we present some refinements of our technique which, when applicable, can be used to prove tighter bounds.

Interestingly, in the cases of 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. For example, for Perceptron, two different instantiations of our approach correspond to two of the most famous proofs of Percetron convergence. One of these instantiations, when applied to the Winnow family, leads to almost exactly the same measures of progress used by Littlestone in [Lit89]. Our results thus show that, in a certain sense, all these previous proofs are closely related.

In Section 6 we introduce a new family of quasi-additive algorithms, which we call p-norm Perceptron algorithms, whose convergence is assured by the results in Section 4. However, we then prove specific mistake bounds for this family, revealing two interesting facts. First, as the parameter p is varied, one can "interpolate" between the additive Perceptron algorithm and multiplicate algorithms in a flexible and principled fashion. Second, the bounds for these algorithms depend on particular products of *conjugate norms* (see Section 6) varying with the parameter p, offering the (yet to be explored) possibility that the new algorithms may be superior to both Perceptron and Winnow in certain contexts.

Finally, note that we discuss related work in more detail in Section 7.

2 Quasi-additive algorithms

An on-line mistake-driven algorithm is determined by its *update* rule; i.e., how it revises $\tilde{w}$ if 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$(\tilde{w}, (\tilde{x}, y))$**:**

If *sign*$(\tilde{w} \cdot \tilde{x}) \neq y$ then
$$\tilde{w} \;:=\; \tilde{w} + a\, y\, \tilde{x}$$

Here the parameter a is a positive constant.

Littlestone has analyzed three other algorithms for learning linear discriminant concepts, which we collectively refer to as the *Winnow* family. On the surface, these might seem very different from Perceptron. But as we now show, there is a fundamental similarity. To see this, consider the balanced version of Winnow described in [Lit89,Lit95] (also called Balanced Weighted Majority). As it is normally presented, it maintains two weight vectors, $\tilde{w}^+$ and $\tilde{w}^-$, and updates each separately:[2]

BalancedWinnow$(\tilde{w}^+, \tilde{w}^-, (\tilde{x}, y))$**:**

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

(Here the parameter α is greater than 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 $a = \log(\alpha)$), but then computes the final weight vector as a function of this sum: $f(z) = e^z - e^{-z}$ $(= 2\sinh(z))$.[3]

BalancedWinnow$(\tilde{w}, \tilde{z}, (\tilde{x}, y))$**:**

If *sign*$(\tilde{w} \cdot \tilde{x}) \neq y$ then
$$\tilde{z} \;:=\; \tilde{z} + \log(\alpha)\, y\, \tilde{x}$$
$$\tilde{w} \;:=\; 2\sinh(\tilde{z}) \;\text{ (i.e., } w_i = 2\sinh(z_i) \text{ for all } i)$$

Thus, Balanced Winnow is just like Perceptron aside from the transformation $f(z) = 2\sinh(z)$. This motivates a natural class of "generalized" Perceptron algorithms. The idea is to distinguish the cumulative sum of mistake vectors $\tilde{z}$ from the final weight vector $\tilde{w}$ 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 $\tilde{z}$.

General$_f(\tilde{w}, \tilde{z}, (\tilde{x}, y))$**:**

If *sign*$(\tilde{w} \cdot \tilde{x}) \neq y$ then
$$\tilde{z} \;:=\; \tilde{z} + a\, y\, \tilde{x}$$
$$\tilde{w} \;:=\; f(\tilde{z}) \;\text{ (i.e., } w_i = f(z_i) \text{ for all } i)$$

(To completely specify the algorithm we also need to specify the initial value of $\tilde{z}$. Unless otherwise specified, we will assume that the initial value is $\tilde{0}$.) Choosing different transformations f yield different algorithms. We call this general family of algorithms *quasi-additive*, as they essentially involve an additive update at their core.

In particular, we can express the other algorithms in the Winnow family in essentially this fashion. For example,

[2] To compare this with the presentation in [Lit89], note that the parameter β there is here $\frac{1}{\alpha}$.

[3] Recall that the hyperbolic sine function sinh is defined by $\sinh x = \frac{e^x - e^{-x}}{2}$.

the algorithm called *Weighted Majority* in [LW89,Lit89] is equivalent[4] to another quasi-additive procedure defined by choosing $f(z) = e^z$ and setting $a = -(\log \beta)/2$.

Finally, the *Fixed Threshold* variant of the Winnow algorithm [Lit88,Lit89] can also be expressed in quasi-additive form, if we make one minor extension. In general, we could 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). More deeply understanding this generalized notion of quasi-additivity, and the somewhat related issue of understanding how to analyze "fixed threshold" algorithms in general, remain largely open questions for us at the moment. Thus, in the rest of the paper, we only consider quasi-additive 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, and assume in the remainder of the paper, is that f is a continuous monotonically increasing function defined on all of $\mathbb{R}$ and not everywhere negative. But as we show in the Section 4, a few simple and broad conditions on f suffice to guarantee *convergence* for $\textbf{General}_f$. 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 giving this convergence result, however, we first give an overview of our general technique for analyzing quasi-additive functions.

3 Measures of progress

Perhaps the main contribution of this paper is to suggest a particular style of analysis for quasi-additive algorithms. In this section we discuss some of the key motivating ideas behind such analyses, giving an informal description of our general framework. Later sections will describe various concrete instantiations of these ideas.

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 $\tilde{u} \in \mathbb{R}^n$ be a fixed *target vector*, that is, a vector such that for all $(\tilde{x}, y) \in S$ we have $sign(\tilde{u} \cdot \tilde{x}) = y$.

The mistake bounds we prove for any algorithm will depend on, among other factors, a quantity we call the *gap*. In this paper, we define the gap as a function of both $\tilde{u}$ and S:

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

(We sometimes omit the subscripts and write just δ if the context is clear.) Very roughly speaking, the gap bounds how close the examples are to the separating hyperplane defined by $\tilde{u}$.

We now outline the basic approach and key ideas we use to construct plausible measures of progress for quasi-additive algorithms. The first, and simpler, of our two main observations is the following. (Note that this is a standard technique, and so is not original to us.) Consider the dot product $L = \tilde{u} \cdot \tilde{z}$, and consider how L changes when we make a single update to $\tilde{z}$ by adding $a\, y\, \tilde{x}$ to $\tilde{z}$ for some example $(\tilde{x}, y) \in S$. Then we have

$$\Delta_L \;=\; (\tilde{u}\cdot\tilde{z}_{new} - \tilde{u}\cdot\tilde{z}) \;=\; \tilde{u}\cdot(\tilde{z}_{new} - \tilde{z}) \;=\; \tilde{u}\cdot(a\, y\, \tilde{x}) \;\geq\; a\, \delta_{\tilde{u},S}$$

using the definition of the gap together with the positivity of a. Thus, L makes steady progress each time we update $\tilde{z}$, and after m updates we have $L \geq m\, a\, \delta_{\tilde{u},S}$. So we have shown that L must increase monotonically as we continue to make mistakes. The simplicity of this argument is perhaps the main consequence of the additivity aspect of quasi-additive algorithms. Of course, this property remains true if we instead define L as $L = q(\tilde{u})\, \tilde{u} \cdot \tilde{z} + r(\tilde{u})$ where q and r are scalar functions of $\tilde{u}$ and $q(\tilde{u}) > 0$; we use this generality in later sections.

All that remains is to find some function $H(\tilde{z})$ such that for some $q(\tilde{u})$ and $r(\tilde{u})$, we are able to show (1) for all $\tilde{z}$, we have $H(\tilde{z}) \geq L$, and (2) that $H(\tilde{z})$ eventually grows more slowly than L (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:

$$M \;=\; H(\tilde{z}) - L \;=\; H(\tilde{z}) - q(\tilde{u})\, \tilde{u} \cdot \tilde{z} - r(\tilde{u}).$$

On the one hand, M is non-negative by assumption. On the other hand, after sufficiently many mistakes the second term is assumed to 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. All our proofs rely on this argument.

To complete this construction we only need supply a suitable function H. Here we introduce the second key idea of this paper, and perhaps the most important. Given the original transformation function f defining our quasi-additive algorithm, we can construct suitable choices for H in the following way. First define g to be an indefinite integral of f. In particular, let t_0 be that point such that $f(t_0) = 0$, or $-\infty$ if no such point exists. (Recall that f is monotonically increasing, so t_0 is unique.) Then we define

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

for all x. (All our later results will include sufficient restrictions to ensure that g exists.) Now let $G(\tilde{z}) = \sum_{i=1}^{n} g(z_i)$. (Note that $G(\tilde{z}) \geq 0$ since, by construction, g is a positive function.) Finally, consider any function H which depends on $\tilde{z}$ only through G; i.e., $H(\tilde{z}) = \psi(G(\tilde{z}))$ for some function ψ that is increasing on $\mathbb{R}^+$.

We propose that such an H is a plausible function to use in our measure of progress. To see this, recall that the

[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.

main issue is ensuring that $H(\tilde{z})$ does not grow too fast as we update $\tilde{z}$. But consider how an H constructed as above changes as we update $\tilde{z}$ on a mistake vector $a\, y\, \tilde{x}$: For a first-order approximation we have:

$$\begin{aligned}
\Delta_H &\approx \sum_{i=1}^{n} \frac{\partial H}{\partial z_i} \Delta_{z_i} \\
&= \sum_{i=1}^{n} \psi'(G(\tilde{z})) f(z_i)\, a\, y\, x_i \\
&= \psi'(G(\tilde{z}))\; a\, y \sum_{i=1}^{n} w_i x_i \\
&\leq 0
\end{aligned}$$

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 when we update by $\tilde{x}$ the change in $\tilde{z}$ is simply $a\, y\, x_i$. The penultimate step simply uses the definition of the quasi-additive rule **General**$_f$, i.e., that $f(z_i)$ is the current weight vector w_i. The final step uses the assumption that ψ is an an increasing function (so its derivative is positive) and, most critically, the fact that $\tilde{x}$ is a vector on which a mistake has just been made. The latter means that $y\,(\tilde{w} \cdot \tilde{x}) \leq 0$ (for otherwise, the prediction using $\tilde{w}$ would not have been mistaken). Thus, at least to a first-order approximation, H *decreases* after an update. This motivates our interest in considering functions H that depend on $\tilde{z}$ only through G. It will also be the *only* use we ever make of the mistake-driven nature of the algorithms.

A geometric intuition might be helpful. For any $\tilde{z}$, let $h = H(\tilde{z})$ and consider the surface passing through $\tilde{z}$ defined by $B_h = \{\tilde{z}' : H(\tilde{z}') = h\}$. We see, again using the first derivative, that the tangent plane to this surface at $\tilde{z}$ is defined by $\{\tilde{z}' : (\tilde{z}' - \tilde{z}) \cdot \tilde{w} = 0\}$; i.e., is defined using the current weight vector. When we move away from $\tilde{z}$ in the direction $\tilde{x}$ (where $\tilde{x}$ is a sample 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 B_h. If $\tilde{x}$ is small enough, or if B_h is not too sharply curved at $\tilde{z}$, this will correspond to a decrease in H.

The intuitive significance of this is it gives us good reason to hope that H will not increase too fast with $\tilde{z}$; recall that this is one of the criteria we seek. Of course, our proofs will have to be more careful than this, and we will see that this is where much of the difficulty arises. It is also one place where the choice of H (which amounts to the choice of ψ) is significant.

In this section we have not given a specific measure of progress, let alone a complete analysis, because ψ, r, and q have not been specified. We make various nontrivial suggestions about these functions in later sections. However, in the next section we make rather straightforward choices. This will serve to illustrate more concretely how the approach just outlined can be made to work and, in the process, provides an extremely general 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, essentially 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 $k \in \mathbb{R}^+$, $\|\tilde{x}\|_k$ denotes the k-norm of $\tilde{x}$, that is $\left(\sum_{i=1}^{n} |x_i|^k\right)^{1/k}$; we also use $\|\tilde{x}\|_\infty = \max_i |x_i|$. Let $\|S\|_k$ denote $\sup_{\tilde{x} \in S} \|\tilde{x}\|_k$.

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*

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

for some $c \geq 0$ (where f' denotes the derivative of f).

Then there exist functions $m_{\delta,\beta}$ and $a_{\delta,\beta} > 0$ such that **General**$_f$*, if run with parameter $a = a_{\delta,\beta}$, makes at most $m_{\delta,\beta}$ mistakes when trained on* any *set of examples S such that for some $\tilde{u}$, $\delta = \delta_{\tilde{u},S}/\|\tilde{u}\|_1$ is greater than 0 and $\beta = \|S\|_\infty$ is finite.*

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 $\tilde{u} \geq 0$, and redefine c as*

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

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.

First we note that the condition $\lim_{z \to -\infty} z^2 f(z) = 0$ from Case 2 can be used to establish the existence of g (i.e., 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 $f(x) > 0$ 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 to 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 1 and 2, $g^{(-1)}$ is well defined on $\mathbb{R}^+$, is increasing, and is continuously differentiable except possibly at 0.

Consider the following measure of progress, defined by taking $\psi = g^{(-1)}$, $q(\tilde{u}) = 1/\|\tilde{u}\|_1$, and $r(\tilde{u}) = 0$:

$$M_f \;=\; g^{(-1)}\left(\sum_{i=1}^{n} g(z_i)\right) - \frac{\tilde{u}\cdot\tilde{z}}{\|\tilde{u}\|_1} \;= H - L$$

where H denotes $g^{(-1)}(\sum_{i=1}^{n} g(z_i)) = g^{(-1)}(G(\tilde{z}))$. It is clear that, without loss of generality, we can assume that $\|\tilde{u}\|_1 = 1$ (in which case L is simply $\tilde{u}\cdot\tilde{z}$); this avoids some notational clutter.

As already discussed, the proof hinges on three claims: (1) M_f is non-negative, (2) L increases by a fixed amount (in fact, by at least $a\delta_{\tilde{u},S}$) at each step and (3) after s steps, where s does not depend on the particular sequence of examples, $H(\tilde{z})$ either decreases, or increases by at most $a\delta_{\tilde{u},S}/2$. Together these clearly suffice to yield a bound on the number of updates that can performed, and thus on the number of mistakes.

Proof of claim (1). We wish to show that $H(\tilde{z}) \geq L$. 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 that $H \geq \max_{i=1}^{n} |z_j|$. But using the assumption that $\sum_{i=1}^{n} |u_i| = 1$ we clearly have $\max_{i=1}^{n} |z_j| \geq \tilde{u}\cdot\tilde{z}$, so we are done.

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

Proof of claim (2). We have already shown, in Section 3, that $L = \tilde{u}\cdot\tilde{z}$ grows by at least $a\,\delta_{\tilde{u},S}$ 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 z_i changes by $y\,a\,x_i$, we are interested in $\Delta_H = H(\tilde{z} + y\,a\,\tilde{x}) - H(\tilde{z})$. In particular, we need to show that (eventually) Δ_H must be small. By Taylor's theorem there is some point $\tilde{\zeta}$ between $\tilde{z}$ and $\tilde{z} + y\,a\,\tilde{x}$ such that

$$\Delta_H = \sum_{i=1}^{n} \left.\frac{\partial H}{\partial z_i}\right|_{\tilde{z}} y\,a\,x_i + \frac{1}{2!}\sum_{i=1}^{n}\sum_{j=1}^{n} \left.\frac{\partial H}{\partial z_i\,\partial z_j}\right|_{\tilde{\zeta}} y^2\,a^2\,x_i\,x_j.$$

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

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). Thus, we can bound Δ_H simply by the second-order term. Analyzing this term comprises the remainder of the proof.

Consider any differentiable function $\psi(G(\tilde{z}))$ (of course, in this proof $\psi = g^{(-1)}$). It is easy to show, using the chain rule, that:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \left.\frac{\partial\psi(G(\tilde{z}))}{\partial z_i\,\partial z_j}\right|_{\tilde{\zeta}} x_i x_j =$$
$$\psi'(G(\tilde{\zeta}))\,\sum_{i=1}^{n} f'(\zeta_i)x_i^2 + \psi''(G(\tilde{\zeta}))\,(\sum_{i=1}^{n} f(\zeta_i)x_i)^2.$$

We can use the standard inverse-function differentiation rule to evaluate the derivatives of $\psi = g^{(-1)}$, obtaining $\psi'(G(\tilde{z})) = \frac{1}{f(g^{(-1)}(G(\tilde{z})))}$ and $\psi''(G(\tilde{z})) = \frac{-f'(g^{(-1)}(G(\tilde{z})))}{f(g^{(-1)}(G(\tilde{z})))^3}$. Note that (and using the assumption that f is differentiable) all these derivatives exist except perhaps, in Case 1, when $\zeta = \tilde{0}$ (because $f(0) = 0$ and this is the only point where $g^{(-1)}(G(\tilde{\zeta})) = 0$). 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:

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

The expression $f(g^{(-1)}(G(\tilde{z})))$ 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,

$$\Delta_H \;\leq\; \frac{a^2\,\sum_{i=1}^{n} f'(\zeta_i)x_i^2}{2f(g^{(-1)}(\sum_i g(\zeta_i)))} \;\triangleq\; Q_f,$$

where we denote this upper bound on Δ_H by Q_f.

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

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

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 $|\zeta_i| \leq \|\tilde{\zeta}\|_\infty$ for all i, and the fact that f' is even (since f is odd). Also for the third step recall that $\beta = \|S\|_\infty = \sup_{\tilde{x}\in S}\|\tilde{x}\|_\infty$ by definition.

But now we can obtain a lower bound on $\|\zeta\|_\infty$ as follows. Clearly, if we have made m updates then $\tilde{u}\cdot\tilde{z}$

will be at least $a\,\delta_{\tilde{u},S}\,m$. We have already shown that $\tilde{u}\cdot\tilde{z} \le \|\tilde{z}\|_\infty$, so that after m updates we have $\|\tilde{z}\|_\infty \ge a\,\delta_{\tilde{u},S}\,m$. Since $\tilde{\zeta}$ is between $\tilde{z}$ and $\tilde{z}+y\,a\,\tilde{x}$, we know that $\|\tilde{\zeta}\|_\infty \ge \|\tilde{z}\|_\infty - a\|\tilde{x}\|_\infty \ge a\delta_{\tilde{u},S}m - a\beta$. (Note that after the first $\beta/\delta_{\tilde{u},S}$ mistakes we must have $\|\tilde{\zeta}\|_\infty > 0$, 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:

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

In particular, if $c \ne 0$ there is some fixed value $z_{2c} > 0$ such that for all $z > z_{2c}$, we have $\sup_{0\le z'<z} \frac{f'(z')}{f(z)} \le 2c$. Suppose $\|\tilde{\zeta}\|_\infty > z_{2c}$. Then, if we choose any a less than $\frac{\delta_{\tilde{u},S}}{2n\beta^2 c}$, we have $Q_f < a\,\delta_{\tilde{u},S}$ as required. From our earlier bound, we also know that $\|\tilde{\zeta}\|_\infty > z_{2c}$ will hold whenever $m > \frac{z_{2c}+a\beta}{a\delta_{\tilde{u},S}}$, so we are done. In the case where $c=0$ we can make $\sup_{0\le z'<z} \frac{f'(z')}{f(z)}$ 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 $\tilde{u}\cdot\tilde{z} \le \max_i z_i$, and thus as m grows we know that the largest *positive* component of $\tilde{z}$ grows without bound. (In Case 1, we only knew this about $\|\tilde{z}\|_\infty$.) 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. □

We now observe that Theorems 4.1 and 4.2 immediately apply to standard algorithms, and thus provide a unified proof of their convergence. For example, the Perceptron algorithm, which is defined by the identity transformation $f(z) = z$ in our framework, 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 of these existing cases.

Of course, this convergence result is interesting because we can apply it to many *new* algorithms. In fact, many other functions f have the properties needed to define convergent quasi-additive algorithms. In Section 6 below we consider some new families of algorithms that are covered by our results. These algorithms demonstrate an interesting "interpolating" property between the Perceptron and Winnow algorithms.

Before studying these new algorithms, we present some refinements of our basic technique that yield tighter mistake bounds in specific cases. Notice that the theorems in this section only demonstrate basic convergence, and do not necessarily provide tight bounds on the number of mistakes the algorithms actually make. That is, although there are particular mistake bounds implicit in the general proof, these bounds have not been optimized in any way, and turn out to be looser than necessary in many cases. In the next section, we consider ways of improving our bounds for specific cases of known algorithms. The key to this will be to reconsider some choices made out of convenience in Theorems 4.1 and 4.2 (e.g., the choices $q(\tilde{u}) = 1/\|\tilde{u}\|_1$, $r(\tilde{u}) = 0$) and show how they can be improved in specific cases.

5 Deriving mistake bounds

To obtain good bounds, we now introduce a more refined instantiation of the ideas from Section 3. Given $H(\tilde{z}) = \psi(G(\tilde{z}))$, recall that we would like to find functions $q(\tilde{u})$ and $r(\tilde{u})$ such that $H(\tilde{z}) \ge q(\tilde{u})\tilde{u}\cdot\tilde{z} + r(\tilde{u})$ for all $\tilde{z}$. There is a general technique for doing this that applies when H is a *convex* function of $\tilde{z}$. A useful geometric intuition about convexity is gained by thinking about the graph of H, i.e., the set $\{(\tilde{z},y) : y = H(\tilde{z})\}$ in $\mathbb{R}^{n+1}$. Given convexity, the tangent plane to this graph considered at some point $\tilde{v}$ will lie entirely below the graph. The tangent at a point $\tilde{v}$ is defined by

$$\left\{ (\tilde{z},y) :\ y = H(\tilde{v}) + \sum_{i=1}^n \left.\frac{\partial H(\tilde{z})}{\partial z_i}\right|_{\tilde{v}} (z_i - v_i) \right\}.$$

It follows, then, that

$$H(\tilde{z}) - H(\tilde{v}) - \sum_{i=1}^n \left.\frac{\partial H(\tilde{z})}{\partial z_i}\right|_{\tilde{v}} (z_i - v_i) \quad \ge \quad 0$$

for *any* (fixed) point $\tilde{v}$. Since we are interested in $H(\tilde{z})$ of the form $\psi(G(\tilde{z}))$, and we know that the derivative simplifies to $\psi'(G(\tilde{z}))f(z_i)$, the above expression can be rewritten as:

$$\begin{aligned} H(\tilde{z}) \quad &- \quad \psi'(G(\tilde{v}))\sum_{i=1}^n f(v_i) z_i \\ &- \quad \left(H(\tilde{v}) - \psi'(G(\tilde{v}))\sum_{i=1}^n f(v_i) v_i \right). \end{aligned}$$

Corresponding to any target vector $\tilde{u}$, we choose a point $\tilde{v}$ such that $f(v_i) = u_i$. Substituting $f^{(-1)}(u_i)$ for v_i in the above expression we obtain a measure of progress in the desired form

$$M \quad = \quad H(\tilde{z}) - q(\tilde{u})\,\tilde{u}\cdot\tilde{z} - r(\tilde{u})$$

for appropriate q and r. The resulting measure of progress is guaranteed to be non-negative (given convexity of H) and, as we now show, has the ability to give better bounds than the cruder but more general analysis used in Section 4.

These new measure of progress can be used in proofs that follow an extremely similar structure to that given in Section 4. In this section, we continue to take $\psi = g^{(-1)}$. Recall that in Section 4 we used a very general argument to bound the increase in the measure of progress M_f by $\Delta_{M_f} \le Q_f - a\,\delta_{\tilde{u},S}/\|\tilde{u}\|_1$. The same basic argument still applies when $q(\tilde{u})$ is not $1/\|\tilde{u}\|_1$, showing that in general we have the bound $\Delta_{M_f} \le Q_f - a\,q(\tilde{u})\,\delta_{\tilde{u},S}$. As in Section 4, we will first want to show that after some number of mistakes m_1, we must have $Q_f < \frac{a\,q(\tilde{u})\,\delta_{\tilde{u},S}}{2}$. It should then be clear that after the first m_1 mistakes the actual value of M_f is monotonically decreasing. Once this has been achieved, the next step is to find some upper bound B on the value of M_f. Given this, we know we can only make at most

$$m_2 = \frac{2B}{a\,q(\tilde{u})\,\delta_{\tilde{u},S}}$$

further mistakes (otherwise the measure of progress would become negative). Thus, the total $m_1 + m_2$ gives our mistake bound. Note that $r(\tilde{u})$ can affect the bound because it influences B.

This completes our outline of a general approach to proving mistake bounds for convex H, much of which (including the definition of M_f) is fairly "automatic". The rest of the details, notably the analysis of Q_f, tends to be specific to the particular function f. However, obtaining a reasonable bound often just involves algebraic manipulation.

To illustrate the technique in full, we work through two example in detail. First, we consider the Weighted Majority algorithm. In this case $f(z) = e^z$, thus

$$M_{WM} = \log \sum_{i=1}^{n} e^{z}_{i} - \frac{\tilde{u}\cdot\tilde{z}}{\|\tilde{u}\|_1} + \sum_{i=1}^{n} \frac{u_i}{\|\tilde{u}\|_1} \log \frac{u_i}{\|\tilde{u}\|_1}.$$

We assume $\tilde{u} \ge 0$. To see how this follows from the above formulation, note that $\psi = \log$ and $G(\tilde{v}) = \sum_{i=1}^{n} e^{v_i} = \sum_{i=1}^{n} u_i = \|\tilde{u}\|_1$ where the penultimate step follows by the choice of $v_i = f^{(-1)}(u_i)$, and the final step uses the positivity of $\tilde{u}$ (which is necessary for Weighted Majority to converge). We note, without proof, that $\log \sum_{i=1}^{n} e^{z_i}$ is a convex function of $\tilde{z}$. Next, we consider

$$Q_{WM} = \frac{a^2 \sum_{i=1}^{n} e^{\zeta_i} x_i^2}{2\sum_{i=1}^{n} e^{\zeta_i}}.$$

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

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

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_{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 [Lit89] in an equally short argument.

Next, we consider the Perceptron Algorithm. Here $f(z) = z$, the identity function, so $g(z) = z^2/2$, $g^{(-1)}(z) = \sqrt{2z}$, and $H = \sqrt{\sum_i z_i^2} = \|\tilde{z}\|_2$. I.e., H is simply the standard Euclidean norm of $\tilde{z}$. We note that all norms are convex functions. It is easy to see that, using the tangent-plane approach, we have $q(\tilde{u}) = 1/\|\tilde{u}\|_2$ and $r(\tilde{u}) = 0$, and thus we obtain the measure of progress $M_{Percept} = \|\tilde{z}\|_2 - \frac{\tilde{u}\cdot\tilde{z}}{\|\tilde{u}\|_2}$.[6] (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.) Thus, using the fact that $f'(z) = 1$ in this case, we are led to analyze

$$Q_{Percept} \;=\; \frac{a^2 \sum_{i=1}^{n} x_i^2}{2\|\tilde{\zeta}\|_2} \;\le\; \frac{a^2\,\|S\|_2^2}{2\|\tilde{\zeta}\|_2}.$$

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

$$\begin{aligned} m_1 + m_2 \;&=\; \frac{\|\tilde{u}\|_2(t + a\,\|S\|_2)}{a\,\delta} + \frac{2\|\tilde{u}\|_2(t + 2\,a\,\|S\|_2)}{a\,\delta} \\ &=\; \frac{2\|\tilde{u}\|_2^2\|S\|_2^2 + 4\delta\|\tilde{u}\|_2\|S\|_2}{\delta^2} \end{aligned}$$

[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] Note that the positivity of this measure of progress could also have been deduced directly, using the Cauchy-Schwarz inequality.

$$= O\left(\frac{\|\tilde{u}\|_2^2\|S\|_2^2}{\delta_{\tilde{u},S}^2}\right)$$

where we use the fact that by the Cauchy-Schwartz inequality, $\delta \leq \|\tilde{u}\|_2\|S\|_2$. Up to a constant factor, this is just the classic result (see [Pap61,Blo62,Nil65,MP69,DH73]). The similarity is even deeper: our measure of progress is in fact very closely related to that used in [Pap61,MP69]. 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.

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.

6 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 6.1, which proves particular 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. [KW95]), 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 extremely interesting from a practical point of view.

The *p-norm Perceptron* algorithms are those quasi-additive algorithms for which $H(\tilde{z})$ is a p-norm, i.e., $\|\tilde{z}\|_p$ for some $2 \leq p < \infty$. Since the p-norm is $\left(\sum_{i=1}^n g(z_i)\right)^{1/p}$ for $g(z) = |z|^p$, the corresponding transformation function $f_p(z) = p\,\mathrm{sign}(z)|z|^{p-1}$ allows us to express the p-norm Perceptron algorithm in the quasi-additive form $\mathbf{General}_{f_p}$. That is, the weight w_i is given by $p\,\mathrm{sign}(z_i)|z_i|^{p-1}$. Note that since the prediction made by the algorithm only depends on the sign of $\tilde{w} \cdot \tilde{x}$, we could equivalently consider weights $w_i = \mathrm{sign}(z)|z|^{p-1}$.

We will use q to denote the conjugate exponent of p, that is, the value q such that $\frac{1}{p} + \frac{1}{q} = 1$. An interesting and important property of conjugate exponents is *Hölder's inequality*, which says that $\tilde{u} \cdot \tilde{v} \leq \|\tilde{u}\|_p\|\tilde{v}\|_q$ whenever p and q are conjugates.[7] 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 many algorithms.

The following theorem gives mistake bounds for this family of algorithms. In addition to the dependence on conjugate norms, note that we have explicitly included the influence of the initial $\tilde{z}$ vector $\tilde{z}_0$ (in previous sections, we assumed that the $\tilde{z}_0$ 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 6.1 *Let $2 \leq p < \infty$, and $\tilde{u}$ be a target vector for some set of examples S.*

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

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

for any update coefficient $a > 0$.

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

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

Proof Since norms are convex, we can use the tangent-plane construction to derive the measure of progress

$$M_p = \|\tilde{z}\|_p - \frac{\tilde{u} \cdot \tilde{z}}{\|\tilde{u}\|_q}.$$

(We omit the straightforward details of the derivation.) Note that the q-norm arises automatically out of this construction.[8]

To obtain a mistake bound, we must obtain an upper bound on $\|\tilde{z}\|_p$ as a function of $\tilde{z}_0$ and of the number of mistakes that have been made, m. It turns out that a bound in a nice form can be obtained by working with $\|\tilde{z}\|_p^2$ instead of $\|\tilde{z}\|_p$. We can then take the square root of this bound to obtain the desired bound on $\|\tilde{z}\|_p$. In the following, let $\xi(\tilde{z}) = \|\tilde{z}\|_p^2$. We also use β_p as shorthand for $\|S\|_p$, and tend to write $\delta_{\tilde{u},S}$ as just δ.

We perform a second order Taylor expansion on ξ about $\tilde{z}$. Note that ξ actually has the form $\psi(G(\tilde{z}))$ discussed in

[7] The inequality also holds for the pair $\|\cdot\|_1$ and $\|\cdot\|_\infty$, which are also considered conjugate.

[8] An alternative way of deriving this measure of progress (i.e., of proving positivity) would be to appeal to Hölder's inequality directly.

Section 3 (taking $\psi(z) = z^{2/p}$), so our earlier observations about the derivatives of such functions apply. First, since we only update when a mistaken prediction is made, the first-order term in the Taylor expansion is non-positive and so we can ignore it. Second, we can use the bound on the second-order term that was introduced in the proof of Theorem 4.1. This shows that:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2 \xi}{\partial z_i \partial z_j}\bigg|_{\tilde{z}} x_i x_j$$

$$\begin{aligned} &\leq 2(p-1)\left(\sum_{i=1}^{n}|z_i|^p\right)^{\frac{2}{p}-1}\sum_{i=1}^{n}|z_i|^{p-2}x_i^2 \\ &= 2(p-1)\|\tilde{z}\|_p^{2-p}\sum_{i=1}^{n}|z_i|^{p-2}x_i^2. \end{aligned}$$

(For the inequality, we use the fact that ψ'' is non-positive for $p >= 2$.) Thus, after updating by $\tilde{x}$, we have

$$\begin{aligned} \Delta_\xi &= \xi(\tilde{z} + a\,\tilde{x}) - \xi(\tilde{z}) \\ &\leq \frac{(p-1)a^2\sum_{i=1}^{n}|\zeta_i|^{p-2}x_i^2}{\|\tilde{\zeta}\|_p^{p-2}} \end{aligned}$$

for some $\tilde{\zeta}$ between $\tilde{z}$ and $\tilde{z} + a\tilde{x}$. The sum in the numerator can be viewed as a dot product (between vectors whose components are ζ_i^{p-2} and x_i^2 respectively), and 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:

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

This is a convenient form because the dependence on $\tilde{\zeta}$ has vanished completely; this is the major advantage of considering the function ξ in our analysis.

Thus, after m mistakes, $\xi(\tilde{z}) \leq \xi(\tilde{z}_0) + m(p-1)a^2\beta_p^2$, and so:

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

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

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

Thus, the greatest m at which this quantity is non-negative is a bound on the number of mistakes we can make.

For this expression to be non-negative, we must have either $\tilde{u}\cdot\tilde{z}_0 + m\,a\,\delta \leq 0$ or $\|\tilde{z}_0\|_p^2 + m(p-1)a^2\beta_p^2 \geq \left(\frac{\tilde{u}\cdot\tilde{z}_0 + ma\delta}{\|\tilde{u}\|_q}\right)^2$. But since the latter condition holds when $\tilde{u}\cdot\tilde{z}_0 + ma\delta = 0$, it follows that the former condition can never lead to the largest value of of m. Thus we concentrate on the second (quadratic) condition, which can be reformulated as:

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

Since this formula is quadratic with a positive leading coefficient, this condition cannot hold when m exceeds its larger root. Hence, this root will give us the mistake bound we seek.

Solving the equation is straightforward, but is simplified using the abbreviations $U = \frac{\tilde{u}\cdot\tilde{z}_0}{\|\tilde{u}\|_q\beta_p}$, $Z = \frac{\|\tilde{z}_0\|_p}{\beta_p}$, $D = \frac{\delta}{\|\tilde{u}\|_q\beta_p}$, 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 $\tilde{z}_0 = 0$), 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 $\tilde{u}\cdot\tilde{z}_0 = \|\tilde{z}_0\|_p\|\tilde{u}\|_q$, 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 $\tilde{u}$.

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

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

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

The bounds in Theorem 6.1 depend on p and therefore one might think that they would necessarily become worse as p increases. This is indeed a problem when $\tilde{z}_0 = \tilde{0}$. However, things can change if we choose different initial vectors $\tilde{z}_0$. Consider, in particular, the case where $\tilde{z}_0 = (1, 1, 1, \ldots, 1)$ (which we denote by $\tilde{1}$). As we now show, the bounds above can have a finite limit even as $p \to \infty$.

Corollary 6.2 *Fix $\tilde{z}_0 = \tilde{1}$, and suppose that all components of $\tilde{u} \neq \tilde{0}$ are non-negative. Let p go to infinity. With the update coefficient, a, set as specified in part (b) of Theorem 6.1 the bounds stated in that theorem converge to:*

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

Proof We can write the bound from Theorem 6.1 as

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

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

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

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

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

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

$$\begin{aligned}
&\frac{d\, n^{2-2/q}\|\tilde{u}\|_q^2}{d\, q} \\
&= (2/q^2)(\ln n) n^{2-2/q}\|\tilde{u}\|_q^2 \;+\; n^{2-2/q}\frac{d \ln \|\tilde{u}\|_q^2}{d q}\|\tilde{u}\|_q^2 \\
&= \|\tilde{u}\|_q^2 \left(\begin{array}{l} (2/q^2)(\ln n) n^{2-2/q} \\ +\;\; n^{2-2/q}\left[\frac{-2}{q^2}\ln \sum_1^n u_i^q + \frac{2}{q}\frac{\sum_1^n (\ln u_i) u_i^q}{\sum_1^n u_i^q}\right] \end{array}\right).
\end{aligned}$$

Thus the limit is just

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

which leads to the stated result. □

Note that bounds from this corollary are exactly the same as those derived in Section 5 for the Weighted Majority algorithm. Of course this is not a coincidence, as we now explain. The p-norm Perceptron rule with initial vector $\tilde{z}_0 = \tilde{1}$ is clearly equivalent to using a quasi-additive rule defined by $f(z) = p\ sign(1+z)\ |1+z|^{p-1}$ and beginning with $\tilde{z}_0 = \tilde{0}$. 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 $f(z) = p\ sign(1 + \frac{z}{p-1})\ |1 + \frac{z}{p-1}|^{p-1}$. 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 so 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 $\tilde{u}$ 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.

7 Related work

We are aware of two other explicitly proposed schemes to unify additive (Perceptron-like) algorithms with multiplicative algorithms such as Winnow. Littlestone [Lit97] 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. 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 closer connection with Kivinen and Warmuth's [KW97b] and other related papers. Their algorithms, like ours, base predictions on $\tilde{w} \cdot \tilde{x}$ for some weight vector $\tilde{w}$ and instance $\tilde{x}$, but their predictions can be some continuous function ϕ of $\tilde{w} \cdot \tilde{x}$ and the loss can be a continuous function L of the true label y and $\phi(\tilde{w} \cdot \tilde{x})$. (The choice of these functions is discussed in [HKW96].) This is sometimes referred to as a *regression* problem, as compared to the classification problem we consider.

Kivinen and Warmuth give a very general construction for deriving various learning algorithms, which we now briefly describe. Let $d(\tilde{w}, \tilde{w}')$ be some *distance* function between weight vectors (not necessarily symmetric). Suppose we see an example $(\tilde{x}, y)$, and let $\tilde{w}$ be the current weight vector. We wish to update to a new weight vector $\tilde{w}'$. Kivinen and Warmuth's intuition is that we should balance a conservativeness factor, $d(\tilde{w}', \tilde{w})$, measuring how little we change the old weight, with a correctiveness factor based on $L(y, \tilde{w} \cdot \tilde{x})$, which measures well the new vector would do on the current example. We can choose $\tilde{w}'$ to approximately minimize the sum $d(\tilde{w}', \tilde{w}) + \eta\, L(y, \tilde{w}' \cdot \tilde{x})$, where η is a parameter we can choose to control the trade-off. It turns out that when d is Euclidean distance and L is square loss, this reconstructs the *gradient descent* algorithm (also called the Widrow-Hoff algorithm), which is the regression analogue of Perceptron: at each update, the weights are modified by adding some multiple of $\tilde{x}$. When d is relative entropy, one obtains an algorithm called Exponentiated Gradient, which is a multiplicative scheme and is related to Winnow. Kivinen and Warmuth also propose a particular general analysis technique, based on a measure of progress constructed from d and L.

The obvious question concerns the relationship of this work to ours. In their approach one can vary both d and L to obtain different algorithms. The loss function L is important to the difference between classification and regression problems. Note that if one constructs a loss function that is 1 if $y \neq sign(\tilde{w} \cdot \tilde{x})$, and otherwise 0, then one is essentially in the classification setting we consider. But this loss function is not continuous and thus does not fit into the class of loss functions that are discussed in [KW97b,HKW96]. However, Warmuth has been exploring a generalization of their approach that does include classification problems [War97]. Conversely, we hope that our framework could be extended to give new understanding of regression problems.

In our framework, we work with two types of variables that are in some sense dual to each other: the $\tilde{z}$ vectors, which change in proportion to the instances when mistakes are made, and the weight vectors, used in making predictions. The weight vectors are functions of the $\tilde{z}$ vectors, related by the gradient of our H function. Our definition of the quasi-additive family, based on varying a function f that maps the $\tilde{z}$ vectors to the weights, has the effect of restricting the class of H functions that we consider. Focusing on this particular family has allowed us to obtain explicit general convergence results (Theorems 4.1 and 4.2), which help identify which of the algorithms suggested by the framework will converge.

The distance function d used in [KW97b] also relates dual parameters (in their setting, the precise definition of the parameter corresponding to $\tilde{z}$ depends on the loss that is being considered); Warmuth and Kazoury plan to discuss this duality, and a motivation based on the exponential family of distributions, in [KW97a]. In their setting the update rule changes as the distance function is varied. In [KW97b], they discuss distance extremely generally, and then study certain specific examples; a class corresponding to our quasi-additive algorithms does not arise, and they do not include any general convergence result analogous to ours. However, in recent (and independent) research, Warmuth and others have been studying a family of algorithms much more closely related to our quasi-additive algorithms, although in somewhat different settings, and they have identified a class of distance functions that corresponds to this family of algorithms [KW97c,KW97a,WJ97]. So it is reasonable to conjecture that two approaches will end up being fairly closely related to each other.

Our final observation about Kivinen and Warmuth's work is to point out that they have previously observed the surprising role of products of dual norms (over $\tilde{u}$ and S, just as in Section 6), in the regression case. They prove a lower bound involving such norms and conjecture that there exist algorithms with upper bounds corresponding to any dual pair.[9]

Aside from the regression case, another different but seemingly related learning model is the so-called *expert* case (e.g., [LW89,Vov90,CFH+93,CHP96]), 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

[9] Note, though, that Kivinen and Warmuth were also concerned with the particular constants occurring in their bounds, so even if we were to ignore the difference between regression and classification, our results would not answer their conjecture. (Note that the conjecture appeared in the earlier (STOC) version of their paper .)

interesting to see if our analysis adds any additional insight.

8 Future work

There remain many intriguing lines for future research. One direction is in looking for other new algorithms suggested by our analysis, and then complementing a theoretical analysis (in the style of Section 6) with appropriate empirical work. The need for the latter is particularly great in the case of the p-norm Perceptron algorithms, because of the obvious question as to whether the "blending" of Perceptron and Weighted Majority that they represent has any practical use.

There are also many open theoretical questions. We have already noted some of the extensions to the basic model that interest us. We are in the process of trying to understand the best way of understanding *fixed threshold* algorithms (roughly speaking, instead of comparing $\tilde{w} \cdot \tilde{z}$ with 0, such algorithms compare the dot product against some fixed value θ). Some early results give us reason to be optimistic that these can also be analyzed using the techniques presented in this paper, although the extension does not seem to be entirely trivial. Another obvious question is to consider the *regression* model discussed in Section 7; not only is this an inherently important problem but may also help us understand the connection with [KW97b] more completely.

We suspect that the analysis suggested in this paper has not yet reached its final form, even for the straightforward case of pure quasi-additive classification algorithms. Here we briefly describe just one of the directions we are pursuing. The tangent-plane construction we have used is appropriate for differentiable convex functions (although even then it may not be the "best" possible). However, it can be generalized to allow the construction of measures of progress from any H whose graph can be bounded below by planes in directions corresponding to all desired target weight vectors. Neither convexity nor differentiability is essential for this, though differentiability may be useful at other stages (in particular, in proving that progress is made at updates). The theory and language of *Legendre-Fenchel transforms* [Ell85] (also called conjugate functions in convex analysis [Roc70]) seems particularly useful here. Briefly, the Legendre-Fenchel transform of $H(\tilde{z})$ is the function $H^*(\tilde{u})$ equal to the largest value such that $H(\tilde{z}) - \tilde{u} \cdot \tilde{z} + H^*(\tilde{u})$ is non-negative for all $\tilde{z}$. In particular, this is exactly the correction we need to obtain a non-negative measure of progress. This definition may seem somewhat empty, but in fact there is a rich theory of these transformations and the associated duality relationship, and our preliminary research suggests that this may lead to a general and more powerful understanding of our results.

We close this section by hinting at way in which Legendre-Fenchel transforms have helped us. In Section 3 we formulated our approach allowing for an arbitrary increasing function ψ of $G(\tilde{z})$, and yet we only ever used $\psi = g^{(-1)}$ in later proofs. Can one get better bounds using other ψ? Interestingly, if one analyzes Perceptron with ψ as the identity function (so that $H(\tilde{z}) = \|\tilde{z}\|_2^2$), one can also prove a mistake bound. The analysis (whose details we omit, but which is related to the proof of Theorem 6.1) turns out to essentially isomorphic to *another* standard proof of Perceptron convergence—see [DH73]. This other proof gives the same bound. We have been able to show, using Legendre-Fenchel transforms as an important tool, that in this and various other cases, varying ψ from $g^{(-1)}$ (subject to certain conditions) in fact cannot lead to better bounds. We thus conjecture that these two classic proofs are related an even deeper sense than simply being particular instantiations of our framework. Of course, we would like to extend these observations and obtain a more thorough understanding of the role of ψ in general.

9 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 analyzing 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, our technique gives the same or very similar results to 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.

Acknowledgements

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

References

[Blo62] H. D. Block. The perceptron: a model for brain functioning. *Reviews of Modern Physics*, 34(1):123–135, 1962.

[CFH$^+$93] Nicolo Cesa-Bianchi, Yoav Freund, David P. Helmbold, David Haussler, Robert E Schapire, and Manfred K Warmuth. How to use expert advice. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on the Theory of Computing*, pages 382–391, 1993.

[CHP96] Nicolo Cesa-Bianchi, David P. Helmbold, and Sandra Panizza. On bayes methods for on-line boolean prediction. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, pages 314–324, 1996.

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

[Ell85] R. Ellis. *Entropy, Large Deviations, and Statistical Mechanics*. Springer-Verlag, New York, 1985.

[HKW96] D. Helmbold, J. Kivinen, and M. Warmuth. Worst-case loss bounds for single neurons. In *Proceedings of NIPS-8*, pages 309–315, 1996.

[KW95] Jyrki Kivinen and Manfred K. Warmuth. The perceptron algorithm vs. winnow: linear vs. logarithmic mistake bounds when few input variables are relevant. In *Proceedings of the 1995 Workshop on Computational Learning Theory*, pages 289–296, 1995.

[KW97a] Katy Kazoury and Manfred K. Warmuth. Relative loss bounds and the exponential family of distributions. (unpublished manuscript), 1997.

[KW97b] J. Kivinen and M. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997. An earlier version appeared in *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing* (STOC 1995).

[KW97c] Jyrki Kivinen and Manfred Warmuth. Relative loss bounds for multiclass regression problems. (unpublished manuscript), 1997.

[Lit88] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear threshold algorithm. *Machine Learning*, 2(4):285–318, 1988.

[Lit89] N. Littlestone. *Mistake bounds and linear-threshold learning algorithms*. PhD thesis, University of California, Santa Cruz, 1989. Technical Report UCSC-CRL-89-11.

[Lit91] N. Littlestone. Redundant noisy attributes, attribute errors, and linear-threshold learning using Winnow. In *Proceedings COLT-91*, pages 147–156, 1991.

[Lit95] N. Littlestone. Comparing several linear-threshold learning algorithms on tasks involving superfluous attributes. In *Proceedings ML-95*, pages 353–361, 1995.

[Lit97] N. Littlestone. An apobayesian relative of winnow. In *Advances in Neural Information Processing Systems 9*, 1997.

[LW89] N. Littlestone and M. Warmuth. The weighted majority algorithm. In *Proceedings FOCS-89*, pages 256–261, 1989.

[MP69] M. L. Minsky and S. A. Papert. *Perceptrons*. MIT Press, Cambridge, MA, 1969.

[Nil65] N. J. Nilsson. *Learning Machines*. Morgan Kaufmann, San Mateo, CA, 1965.

[Pap61] S. Papert. Some mathematical models of learning. In *Proceedings of the Fourth London Symposium on Information Theory*, 1961.

[Roc70] R. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[Ros62] F. Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, Washington, D. C., 1962.

[Vov90] Volodimir G. Vovk. Aggregating strategies. In *Proceedings of the 1990 Workshop on Computational Learning Theory*, pages 371–383, 1990.

[War97] Manfred Warmuth. Personal communication, 1997.

[WJ97] Manfred K. Warmuth and Arun Jagota. Continuous time non-linear gradient descent: Convergence and relative loss bounds. (unpublished manuscript), 1997.