The Robustness of the p -Norm Algorithms

CLAUDIO GENTILE

gentile@dsi.unimi.it

CIRI, Università dell'Insubria, Via Ravasi, 2, 21100—Varese, Italy

Editor: Lisa Hellerstein

Abstract. We consider two on-line learning frameworks: binary classification through linear threshold functions and linear regression. We study a family of on-line algorithms, called p -norm algorithms, introduced by Grove, Littlestone and Schuurmans in the context of deterministic binary classification. We show how to adapt these algorithms for use in the regression setting, and prove worst-case bounds on the square loss, using a technique from Kivinen and Warmuth. As pointed out by Grove, et al., these algorithms can be made to approach a version of the classification algorithm Winnow as p goes to infinity; similarly they can be made to approach the corresponding regression algorithm EG in the limit. Winnow and EG are notable for having loss bounds that grow only logarithmically in the dimension of the instance space. Here we describe another way to use the p -norm algorithms to achieve this logarithmic behavior. With the way to use them that we propose, it is less critical than with Winnow and EG to retune the parameters of the algorithm as the learning task changes. Since the correct setting of the parameters depends on characteristics of the learning task that are not typically known *a priori* by the learner, this gives the p -norm algorithms a desirable robustness. Our elaborations yield various new loss bounds in these on-line settings. Some of these bounds improve or generalize known results. Others are incomparable.

Keywords: on-line learning, loss bounds, learning rate, dual norms, on-line model selection

1. Introduction

In this paper, we consider on-line learning from labeled examples. We consider a family of algorithms that are suitable both for learning linear functions and for learning binary classification, with only minor modifications needed to convert from one task to the other. For both tasks, in each on-line trial the learning algorithm observes an *instance* chosen from $\mathcal{R}^n$ and is required to predict the value of a *label* associated with that instance. When learning linear functions (this case is also referred to as *linear regression*) the label to be predicted is a real number; for the classification task the label is chosen from $\{-1, +1\}$. In each trial the learner maintains a weight vector representing its current hypothesis. Combined with the current instance, this weight vector determines the learner's prediction. In linear regression the prediction is a real number, while in binary classification the prediction lies in $[-1, +1]$. In the special case when the prediction is ± 1 we say we are in the *deterministic* binary classification setting. When the algorithm's prediction spans the whole interval $[-1, +1]$ we say we are in the *randomized* binary classification setting. After the learner makes its prediction, the value of the label is revealed and the learner suffers a *loss*, measuring the discrepancy between the prediction and the actual label. Then the learner *updates* its weights.

We are interested in comparing the performance of the algorithms with the best performance obtainable with a fixed weight vector. It is sometimes convenient to think of the data

for each trial as being generated by first choosing the instance and then choosing the label using such a weight vector. The label may then be corrupted with noise. When thinking this way, we refer to the weight vector as a *target vector*. If the target vector perfectly determines the labels we say that the learning task is noise-free with respect to that target. We say that attributes whose weight is 0 in a target vector are *irrelevant* with respect to that target. (An *attribute* is a component of an instance.)

The algorithms that we consider belong to a general family of algorithms that was developed by Grove, Littlestone, and Schuurmans (2001), who studied binary classification, and also, in separate work, by Kivinen and Warmuth (2001), who consider (generalized) linear regression. These algorithms are called *quasi-additive* by Grove et al., and *general additive* by Kivinen and Warmuth. The analysis of these algorithms in these papers yields bounds on the cumulative loss that they suffer.

This family of algorithms includes a number of well-known algorithms. For instance, in the classification setting it includes the classical Perceptron algorithm (Rosenblatt, 1962; Block, 1962; Novikov, 1962) and algorithms in the “Winnow family”,¹ such as the Zero-threshold Winnow. In the binary *expert* framework the version that we call Zero-threshold Winnow is equivalent to the Weighted Majority algorithm; (see Littlestone, 1989; Littlestone & Warmuth, 1994; Vovk, 1990; Cesa-Bianchi et al., 1997). In the regression (linear function) setting the general additive algorithms include the Widrow-Hoff algorithm, also called the least mean square rule (Widrow & Hoff, 1960; Cesa-Bianchi, Long, & Warmuth, 1996; Kivinen & Warmuth, 1997) and Kivinen and Warmuth’s “EG family” of algorithms, such as $EG^\pm$ and EGU (Kivinen & Warmuth, 1997). Perceptron and Widrow-Hoff update their weights additively, adding some multiple of the current instance to the current weight vector. Thus we call them *additive* algorithms. Winnow and EG perform multiplicative updates. Thus we call them *multiplicative* algorithms.

Here we focus on a particular sub-family of the general additive algorithms, the *p*-norm Perceptron algorithms, that was introduced by Grove et al. They discuss it only in the context of deterministic binary classification. One of the main results of the present paper is the demonstration that it can be adapted for use in the linear regression setting. We will refer to these algorithms for classification and regression collectively as *p*-norm algorithms.

Grove, Littlestone, and Schuurmans (2001) show that the *p*-norm Perceptron algorithms for binary classification actually can interpolate smoothly between the Perceptron at $p = 2$ and Weighted Majority as $p \rightarrow \infty$ (if appropriate care is taken in how the limit is approached). It is straightforward to see that this also holds for the regression case, where the *p*-norm algorithms can interpolate between Widrow-Hoff and EG.

Perhaps the most notable characteristic of the multiplicative algorithms is that their cumulative loss grows only logarithmically with the number of irrelevant attributes of the target weight vector. With appropriate care, the *p*-norm algorithms approach the multiplicative algorithms in the limit as p goes to infinity, and so in that limit their bounds are also logarithmic in n . However, to obtain this logarithmic behavior in the limit, the initial weight vector must be an appropriate non-zero weight vector, and the ratio of the initial weight vector to the factor used to scale the updates (this factor is usually called the *learning rate*) must grow in proportion to p and be sufficiently large. The ratio that suffices depends on properties of the learning task that are not necessarily visible to the learner, and thus

estimates of the appropriate ratio are needed. Though in many cases the accuracy of these estimates turns out in practice not to be too critical, the multiplicative algorithms' learning can entirely break down in some cases if the update factor is too large.²

The second main result of this paper is that the logarithmic behavior can be obtained with the p -norm algorithms in a different way, for finite p and zero initial weight vector. To do this, we choose p to be an appropriate constant multiple of $\log n$. In the case of deterministic binary classification, this lets us choose the learning rate entirely arbitrarily. With zero initial weight, the choice of the learning rate does not affect the behavior of the algorithm at all. If the initial weight vector and the target vector are both positive, then one can improve the performance somewhat through tuning the learning rate, but the p -norm algorithm with this logarithmic choice of p will still do well as the learning rate goes to infinity (this corresponds to letting the initial weights go to zero). Thus one need not fear the breakdown in learning that can occur with Winnow. In the case of randomized binary classification, we have recently shown in Auer and Gentile (2000) how it is possible to obtain a quasi-optimal logarithmic behavior by scaling the learning rate as a function of the current cumulative loss of the algorithm.

The p -norm algorithm with logarithmic p is *robust* in this regard; that is, there is a way to safely choose its parameters that depends only on information available to the learner, such as the dimension n of the instance space and the cumulative loss of the algorithm.

These considerations are the main theme of this paper. They motivate the study of the p -norm algorithms and allow us to prove various interesting new results related to the on-line settings we are concerned with.

In the regression setting we provide the first loss bound for the p -norm algorithms to date (Theorem 7). In the classification setting we give an extension of the analysis of Grove, Littlestone, and Schuurmans (2001) to the non-linearly separable case (Theorem 8). The bounds we prove are in terms of a generic pair of dual norms, which determine the interpolation ability of the p -norm algorithms.³

Then we consider the p -norm algorithm with logarithmic p and we compare it to various known algorithms in the Winnow and EG family. In particular, in the linear regression setting we show (Corollary 9) that such a p -norm algorithm learns linear functions with a bound similar to the one proven for the EGU algorithm by Kivinen and Warmuth (1997).

In the case of EGU, it is not clear how to extend their analysis to handle negative instances and labels; our p -norm bound does not have this limitation. One might look at the p -norm algorithm with logarithmic p as a new basic on-line regression technology. Indeed, one can easily extend our result to more general regression frameworks, such as the generalized linear regression model of Helmbold, Kivinen, and Warmuth (1999), Kivinen and Warmuth (2001) and Herbster and Warmuth's shifting framework (1998a, 1998b).

In the case of deterministic binary classification we use the p -norm algorithm with logarithmic p to learn r -of- k threshold functions Littlestone (1988, 1989b, 1991). (This class includes monotone k -literal disjunctions and k -literal conjunctions as 1-of- k and k -of- k threshold functions, respectively.) Here we prove new bounds, which are in general incomparable to the bounds proven for Winnow by Littlestone (1991) and Auer and Warmuth (1998). In the special case $k = r = 1$ this gives a bound in the (binary) expert framework (Littlestone, 1989b; Vovk, 1990; Littlestone & Warmuth, 1994; Cesa-Bianchi et al., 1997).

This bound improves on a result obtained by Cesa-Bianchi, Helmbold, and Panizza (1998), which was essentially the best robust bound to date. In the case of randomized binary classification we adapt the p -norm algorithm to learn with respect to an *infinite* pool of experts. This learning problem might be seen as an on-line model selection problem with a countably infinite set of models. This adaptation requires combining the techniques developed in Auer and Gentile (2000) with the ability to analyze a p -norm algorithm where p increases with time (Theorem 15 and Corollary 16). We improve on results in Littlestone and Warmuth (1994) and partially solve a problem left open in Cesa-Bianchi et al. (1997).

The next section contains the major definitions used throughout the paper. Section 3 defines the p -norm algorithms for linear regression and deterministic binary classification. In Section 4 we prove our basic loss bounds. Section 5 uses the results in Section 4 to prove robust bounds for the p -norm algorithm with logarithmic p . The randomized classification algorithm for an infinite pool of experts requires a somewhat special treatment. Therefore its introduction is postponed till Section 5.3.

2. Preliminaries

In this section we recall some basic preliminaries and introduce the main notation used throughout the paper.

Learning proceeds in trials. In the t -th trial, the algorithm receives an *instance* $\mathbf{x}_t$ from the environment and produces a *prediction* $\hat{y}_t$. The instances belong to a given *instance space* $\mathcal{X} \subseteq \mathcal{R}^n$. Then the environment discloses the true label y_t associated with $\mathbf{x}_t$ and the algorithm suffers a loss $L(y_t, \hat{y}_t)$, measuring how far the prediction $\hat{y}_t$ is from the label y_t .

In this paper, this prediction will always be a function of $\mathbf{w}_t \cdot \mathbf{x}_t$, where $\mathbf{w}_t$ represents the current weight vector of the algorithm. The weight vector represents the learner's current hypothesis and is contained in some suitable *weight space* $\mathcal{W} \subseteq \mathcal{R}^n$. The algorithm uses the true label to update its weight vector. We call a sequence $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_t, y_t), \dots, (\mathbf{x}_T, y_T))$ of instances and labels observed by an algorithm in a run a *trial sequence*.

The mathematical model we use in the analysis of the algorithms is a generalization of a learning model introduced by Littlestone (1988, 1989), Littlestone and Warmuth (1994) and Angluin (1988). We measure the performance of an on-line algorithm A on S by the cumulative loss $L_A(S)$ the algorithm A suffers on S : $L_A(S) = \sum_{t=1}^T L(y_t, \hat{y}_t)$. Our goal is to bound the amount by which this loss on an arbitrary sequence exceeds some measure of the performance of a fixed weight vector. The measure that we use depends on whether we are considering the regression or the classification setting.

A main distinguishing feature of this learning model is the absence of probabilistic assumptions about the trial sequence S . In other words, we assume that S is generated by an adversarial environment. It is well-known that results in this learning model can be converted into results for probabilistic frameworks; (see e.g., Littlestone, 1989a; Helmbold & Warmuth, 1995; Forster & Warmuth, 2000).

In this paper we focus on three specific on-line learning problems: linear regression with square loss, deterministic binary classification using linear threshold functions with threshold 0, and randomized binary classification in the (infinite) expert framework. Such problems have been widely studied in the last decade; see, e.g., Littlestone (1988, 1989b,

1991), Vovk (1990, 1997), Littlestone and Warmuth (1994), Cesa-Bianchi et al. (1996, 1997), Kivinen and Warmuth (1997, 1999, 2001), Yamanishi (1998), Bylander (1997), Grove, Littlestone, Schuurmans (2001), Freund and Schapire (1999), Gentile and Warmuth (1999), and references therein.

In the linear regression setting the label $y_t \in \mathcal{R}$ and the prediction $\hat{y}_t$ is the linear function $\hat{y}_t = \mathbf{w}_t \cdot \mathbf{x}_t$. Here we measure the loss of the algorithm with the square loss $L(y_t, \hat{y}_t) = \frac{1}{2}(y_t - \hat{y}_t)^2 \in \mathcal{R}$. We compare this loss with the least square loss that could be incurred by predicting with a fixed weight vector chosen from some *comparison class*.

In the deterministic binary classification setting the label y_t belongs to $\{-1, +1\}$ and the algorithm predicts with the thresholded linear function⁵ $\hat{y}_t = \text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t)$. Here we measure the loss of A with the discrete loss $L(y_t, \hat{y}_t) = \frac{1}{2}|y_t - \hat{y}_t| \in \{0, 1\}$. Thus the loss is one when a prediction mistake occurs, i.e., when $\hat{y}_t \neq y_t$, while the cumulative loss is just the number of mistakes the algorithm makes. In the randomized binary classification setting the label y_t is still in $\{-1, +1\}$, but the algorithm is allowed to make randomized predictions, i.e., it is allowed to output a prediction value $\hat{y}_t$ lying in $[-1, +1]$. The discrepancy between the prediction $\hat{y}_t$ and the label y_t is measured through the absolute loss $L(y_t, \hat{y}_t) = \frac{1}{2}|y_t - \hat{y}_t| \in [0, 1]$. This loss might be interpreted as the *probability* of a prediction mistake. The cumulative loss would then be the expected number of mistakes. We compare either the cumulative discrete loss or the cumulative absolute loss with what a fixed weight vector can achieve, but the way we measure the performance of the fixed weight vector is not necessarily in terms of the number of mistakes it would make; see Section 4.2.

Norms. For $\mathbf{w} = (w_1, \dots, w_n) \in \mathcal{R}^n$ and $p \geq 1$ we denote by $\|\mathbf{w}\|_p$ the p -norm of $\mathbf{w}$, i.e., $\|\mathbf{w}\|_p = (\sum_{i=1}^n |w_i|^p)^{1/p}$ (also, $\|\mathbf{w}\|_\infty = \lim_{p \rightarrow \infty} (\sum_{i=1}^n |w_i|^p)^{1/p} = \max_i |w_i|$). We say that q is *dual* to p if $\frac{1}{p} + \frac{1}{q} = 1$ holds. For instance, the 1-norm is dual to the ∞ -norm and the 2-norm is self-dual. It is well-known that for any $\mathbf{w} \in \mathcal{R}^n$ we have $\|\mathbf{w}\|_{p_1} \leq \|\mathbf{w}\|_{p_2}$ whenever $p_1 \geq p_2$. For the rest of this paper we assume that p and q are some pair of dual values, with $p \geq 2$.

3. The p -norm algorithms

We define the p -norm algorithms using the framework of the general additive algorithms. For clarity, we work with two state parameters. One is the weight vector of the algorithm and the other is a vector obtained from the weight vector by applying some mapping $\mathbf{f}$, termed a *link function* by Jagota and Warmuth (1998). (In this paper $\mathbf{f}$ is bijective. Thus the two parameters are redundant; either can be used to give a complete description of the algorithm's state.) We write $\boldsymbol{\theta} = \mathbf{f}(\mathbf{w})$. The general additive algorithms are characterized by making additive updates in the $\boldsymbol{\theta}$ parameter space. We define the link function $\mathbf{f}$ for the p -norm algorithms as follows (a p -indexing for $\mathbf{f}$ is understood): $\mathbf{f} : \mathcal{R}^n \rightarrow \mathcal{R}^n$, $\mathbf{f} = (f_1, \dots, f_n)$, where

$$f_i(\mathbf{w}) = \frac{\text{sign}(w_i) |w_i|^{q-1}}{\|\mathbf{w}\|_q^{q-2}}, \quad \mathbf{w} = (w_1, \dots, w_n) \in \mathcal{R}^n. \quad (1)$$

We have the following lemma.

Lemma 1. *The link function $\mathbf{f}$ defined in (1) enjoys the following properties⁶:*

1. $\mathbf{f}$ is bijective, its (unique) inverse is $\mathbf{f}^{-1} : \mathcal{R}^n \rightarrow \mathcal{R}^n$, $\mathbf{f}^{-1} = (f_1^{-1}, \dots, f_n^{-1})$, where

$$f_i^{-1}(\theta) = \frac{\text{sign}(\theta_i) |\theta_i|^{p-1}}{\|\theta\|_p^{p-2}}, \quad \theta = (\theta_1, \dots, \theta_n) \in \mathcal{R}^n, \quad (2)$$

i.e., $\mathbf{f}^{-1}$ is obtained from $\mathbf{f}$ by replacing q with p .

2. Both $\mathbf{f}$ and $\mathbf{f}^{-1}$ are gradients of scalar functions. In particular, $\mathbf{f}$ is the gradient of $P_{\mathbf{f}}(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|_q^2$, while $\mathbf{f}^{-1}$ is the gradient of $P_{\mathbf{f}^{-1}}(\theta) = \frac{1}{2} \|\theta\|_p^2$.
3. $\mathbf{f}$ maps vectors with a given q -norm to vectors with the same p -norm, i.e., for all $\mathbf{w} \in \mathcal{R}^n$ we have $\|\mathbf{f}(\mathbf{w})\|_p = \|\mathbf{w}\|_q$.
4. For all $\mathbf{w} \in \mathcal{R}^n$ the identity $\mathbf{w} \cdot \mathbf{f}(\mathbf{w}) = \|\mathbf{w}\|_q^2$ holds.

Proof: Straightforward, recalling that $\frac{1}{p} + \frac{1}{q} = 1$. □

In order to prove loss bounds for our algorithms, we follow a well-established methodology; see e.g. Littlestone (1988, 1989b), Littlestone and Warmuth (1994), Helmbold, Kivinen, and Warmuth (1999), Cesa-Bianchi et al. (1996, 1997), Grove, Littlestone, and Schuurmans (2001), Kivinen and Warmuth (1997, 2001), Jagota and Warmuth (1998), Bylander (1997), Gentile and Warmuth (1999) and Auer and Gentile (2000). We define a measure of progress based on the link function under consideration. Let $\mathbf{f}$ be as in (1) and $P_{\mathbf{f}}$ be as in part 2 of Lemma 1. Our measure of progress $d_{\mathbf{f}}(\cdot, \cdot)$ is the so-called *Bregman divergence* (see Bregman, 1967; Censor and Lent, 1981) associated with $\mathbf{f}$:

$$d_{\mathbf{f}}(\mathbf{u}, \mathbf{w}) = P_{\mathbf{f}}(\mathbf{u}) - P_{\mathbf{f}}(\mathbf{w}) - (\mathbf{u} - \mathbf{w}) \cdot \mathbf{f}(\mathbf{w}). \quad (3)$$

The Bregman divergence $d_{\mathbf{f}}(\mathbf{u}, \mathbf{w})$ is thus the difference between the strictly convex function $P_{\mathbf{f}}(\mathbf{u})$ and its first-order Taylor expansion around $\mathbf{w}$. Observe that from the strict convexity of $P_{\mathbf{f}}$ it follows that $d_{\mathbf{f}}(\mathbf{u}, \mathbf{w}) \geq 0$ with equality holding if and only if $\mathbf{u} = \mathbf{w}$. By parts 2 and 4 of Lemma 1 it is easy to check that $d_{\mathbf{f}}(\mathbf{u}, \mathbf{w})$ becomes⁷

$$d_{\mathbf{f}}(\mathbf{u}, \mathbf{w}) = \frac{1}{2} \|\mathbf{u}\|_q^2 + \frac{1}{2} \|\mathbf{w}\|_q^2 - \mathbf{u} \cdot \mathbf{f}(\mathbf{w}). \quad (4)$$

The following lemma is a fundamental projection property of Bregman divergences. We will use it in Section 5.3. This lemma has also been used in on-line learning by Herbster and Warmuth (1998b) and Auer and Gentile (2000).

Lemma 2 (Bregman, 1967; Censor & Lent, 1981). *Let $\mathcal{W}$ be a closed and convex subset of $\mathcal{R}^n$, let $\mathbf{w} \in \mathcal{R}^n$, and let $\mathbf{w}' = \arg\min_{\mathbf{u} \in \mathcal{W}} d_{\mathbf{f}}(\mathbf{u}, \mathbf{w})$ be the projection of $\mathbf{w}$ onto $\mathcal{W}$ w.r.t. the Bregman divergence $d_{\mathbf{f}}$ given in (3). Then for any $\mathbf{u} \in \mathcal{W}$ the following holds:*

$$d_{\mathbf{f}}(\mathbf{u}, \mathbf{w}) \geq d_{\mathbf{f}}(\mathbf{u}, \mathbf{w}') + d_{\mathbf{f}}(\mathbf{w}', \mathbf{w}) \geq d_{\mathbf{f}}(\mathbf{u}, \mathbf{w}').$$

Table 1. The p -norm algorithm for linear regression with square loss (“regression”) and deterministic binary classification with threshold 0 (“classification”).

Initialization: Initial weight vector $\mathbf{w}_1 \in \mathcal{R}^n$; learning rate $\eta > 0$.	
For $t = 1, 2, \dots, T$:	
• Get instance $\mathbf{x}_t \in \mathcal{X}$.	
• Predict with $\hat{y}_t = \begin{cases} \mathbf{w}_t \cdot \mathbf{x}_t \in \mathcal{R} & \text{(regression);} \\ \text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t) \in \{-1, +1\} & \text{(classification).} \end{cases}$	
• Get label $y_t \in \begin{cases} \mathcal{R} & \text{(regression);} \\ \{-1, +1\} & \text{(classification).} \end{cases}$	
Incur loss $L(y_t, \hat{y}_t) = \begin{cases} \frac{1}{2}(y_t - \hat{y}_t)^2 \in \mathcal{R} & \text{(regression);} \\ \frac{1}{2} y_t - \hat{y}_t \in \{0, 1\} & \text{(classification).} \end{cases}$	
• Update weights as follows: $\mathbf{w}_{t+1} = \mathbf{f}^{-1}(\mathbf{f}(\mathbf{w}_t) - \eta(\hat{y}_t - y_t)\mathbf{x}_t)$.	

In Section 5.3 we will be projecting onto $\mathcal{W} = \{\mathbf{w} \in \mathcal{R}^n : \|\mathbf{w}\|_q \leq 1\}$, for some finite n . An easy Kuhn-Tucker analysis shows that in such a case $\mathbf{w}' = \mathbf{w} / \max\{1, \|\mathbf{w}\|_q\}$. We also have the following simple lemma, taken from Auer and Gentile (2000).

Lemma 3. Let $\mathcal{W} = \{\mathbf{w} \in \mathcal{R}^n : \|\mathbf{w}\|_q \leq 1\}$ and $\mathbf{u}, \mathbf{w} \in \mathcal{W}$. Then $d_{\mathbf{f}}(\mathbf{u}, \mathbf{w}) \leq 2$.

Proof: The assertion follows from (4) and Hölder’s inequality on the term $-\mathbf{u} \cdot \mathbf{f}(\mathbf{w})$, for $-\mathbf{u} \cdot \mathbf{f}(\mathbf{w}) \leq |\mathbf{u} \cdot \mathbf{f}(\mathbf{w})| \leq \|\mathbf{u}\|_q \|\mathbf{f}(\mathbf{w})\|_p = \|\mathbf{u}\|_q \|\mathbf{w}\|_q \leq 1$, where the equality uses the identity $\|\mathbf{f}(\mathbf{w})\|_p = \|\mathbf{w}\|_q$ (Lemma 1, part 3). $\square$

Table 1 describes the p -norm algorithm for linear regression with square loss and for deterministic binary classification with threshold 0. The randomized classification algorithm will be illustrated in Section 5.3. The algorithm in Table 1 maintains a vector of n weights. It starts from $\mathbf{w}_1$, and in the generic trial t it is required to predict the value of the unknown label y_t of the instance $\mathbf{x}_t$. In the regression problem $y_t \in \mathcal{R}$ and the algorithm predicts through the linear combination $\hat{y}_t = \mathbf{w}_t \cdot \mathbf{x}_t$, while in the deterministic classification problem $y_t \in \{-1, +1\}$ and the algorithm predicts through the thresholded linear combination $\hat{y}_t = \text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t) \in \{-1, +1\}$. When the label y_t is received, the algorithm incurs a loss $L(y_t, \hat{y}_t)$. Such a loss is the square loss $L(y_t, \hat{y}_t) = \frac{1}{2}(y_t - \hat{y}_t)^2 \in \mathcal{R}$ in the linear regression setting and the discrete loss $L(y_t, \hat{y}_t) = \frac{1}{2}|y_t - \hat{y}_t| \in \{0, 1\}$ in the binary classification setting. Then the algorithm updates the weights as indicated.

One can work conveniently in terms of either the $\mathbf{w}$ parameter, as we primarily do here, or the $\boldsymbol{\theta}$ parameter. Grove, Littlestone, and Schuurmans (2001) work in terms of the $\boldsymbol{\theta}$ parameter. Note also that since there is a one-to-one correspondence among $\mathbf{f}$, $\mathbf{f}^{-1}$, $P_{\mathbf{f}}$, and $P_{\mathbf{f}^{-1}}$, one can take any one of these as primary in defining a particular algorithm. The name *p-norm algorithm* comes from thinking in terms of starting with $P_{\mathbf{f}^{-1}}(\boldsymbol{\theta})$, which for the p -norm algorithm is a power of the p -norm of $\boldsymbol{\theta}$. In the zero-threshold deterministic classification case studied by Grove et al., it turns out that we can actually start with $P_{\mathbf{f}^{-1}}(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|_p^s$

for any $s \geq 1$. One obtains equivalent algorithms and the same bounds. The choice of s does affect the normalization of the weights in the algorithm, but this does not matter in that case, since the prediction depends only on the sign of $\mathbf{w}_t \cdot \mathbf{x}_t$. In the regression case, as well as in the randomized classification case, the normalization does make a difference, and different choices of s would lead to different algorithms. However, it turns out that we do not have this flexibility; we need to choose $s = 2$ in such cases to obtain our loss bounds. (The analysis depends on bounding the second-order remainder term in the Taylor expansion of $P_{f^{-1}}$, and it is only for $s = 2$ that this term remains bounded. In the analysis for the zero-threshold deterministic classification, one has additional flexibility that obviates this constraint.)

When $p = 2$ the resulting link function $\mathbf{f}$ is the identity function. In this case the regression algorithm of Table 1 corresponds to the Widrow-Hoff rule, while the classification algorithm turns to the Perceptron algorithm. If p is moderately large the p -norm algorithm tends to behave as a multiplicative algorithm. We will see in Section 5 that when p is logarithmic in n the p -norm algorithm exhibits a clean “logarithmic behavior” and is easier to analyze than the other known robust multiplicative algorithms.

4. Loss bounds for the p -norm algorithms

The overall methodology to prove loss bounds follows a pattern from Kivinen and Warmuth (1997, 2001), Grove, Littlestone, and Schuurmans (2001), Jagota and Warmuth (1998), and Gentile and Warmuth (1999). We first bound the second-order remainder in the Taylor expansion of $P_{f^{-1}}$ (or of P_f). Then we derive one-trial loss bounds for the regression and the classification algorithms. This is done through the following three technical lemmas.

Lemma 4. *Let $\mathbf{w}, \mathbf{x} \in \mathcal{R}^n$, $a \in \mathcal{R}$ and $p \geq 2$. Set $\mathbf{w}' = \mathbf{f}^{-1}(\mathbf{f}(\mathbf{w}) + a\mathbf{x})$, $\boldsymbol{\theta} = \mathbf{f}(\mathbf{w})$ and $\boldsymbol{\theta}' = \mathbf{f}(\mathbf{w}')$. Then the following holds:*

$$d_f(\mathbf{w}, \mathbf{w}') = \frac{1}{2} \|\boldsymbol{\theta}'\|_p^2 - \frac{1}{2} \|\boldsymbol{\theta}\|_p^2 - a \mathbf{w} \cdot \mathbf{x} \leq \frac{a^2}{2} (p-1) \|\mathbf{x}\|_p^2.$$

Proof: Recall the notation $P_f(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|_q^2$, $P_{f^{-1}}(\boldsymbol{\theta}) = \frac{1}{2} \|\boldsymbol{\theta}\|_p^2$. From (4), Lemma 1, parts 3 and 4 and the relationship $\boldsymbol{\theta}' = \boldsymbol{\theta} + a\mathbf{x}$ we have

$$\begin{aligned} d_f(\mathbf{w}, \mathbf{w}') &= P_f(\mathbf{w}) + P_f(\mathbf{w}') - \mathbf{w} \cdot \boldsymbol{\theta}' \\ &= P_{f^{-1}}(\boldsymbol{\theta}) + P_{f^{-1}}(\boldsymbol{\theta}') - \mathbf{w} \cdot \boldsymbol{\theta} - a \mathbf{w} \cdot \mathbf{x} \\ &= P_{f^{-1}}(\boldsymbol{\theta}') - P_{f^{-1}}(\boldsymbol{\theta}) - a \mathbf{w} \cdot \mathbf{x}. \end{aligned}$$

This proves the equality of the lemma. This equality says that $d_f(\mathbf{w}, \mathbf{w}')$ equals the second-order remainder term in the Taylor expansion of $P_{f^{-1}}(\boldsymbol{\theta}')$ around $\boldsymbol{\theta}$. The inequality of the lemma is taken from the proof of Theorem 7.1 in Grove, Littlestone, and Schuurmans (2001), where the authors essentially bound from above the same second-order term. $\square$

Lemma 5. *Let $\mathbf{u}, \mathbf{w} \in \mathcal{R}^n, \mathbf{x} \in \mathcal{R}^n$, with $\|\mathbf{x}\|_p \leq X_p$ and set $\mathbf{w}' = \mathbf{f}^{-1}(\mathbf{f}(\mathbf{w}) - \eta(\hat{y} - y)\mathbf{x})$, with $\eta = \frac{c}{(1+c)(p-1)X_p^2}$, $c \geq 0$, $y \in \mathcal{R}$ and $\hat{y} = \mathbf{w} \cdot \mathbf{x}$. Then the following inequality holds:*

$$\begin{aligned} & \frac{c}{1+c} \frac{1}{(p-1)X_p^2} \frac{1}{2}(y - \hat{y})^2 - \frac{c}{(p-1)X_p^2} \frac{1}{2}(y - \mathbf{u} \cdot \mathbf{x})^2 \\ & \leq d_{\mathbf{f}}(\mathbf{u}, \mathbf{w}) - d_{\mathbf{f}}(\mathbf{u}, \mathbf{w}'). \end{aligned}$$

Proof sketch: Set $\boldsymbol{\theta} = \mathbf{f}(\mathbf{w})$ and $\boldsymbol{\theta}' = \mathbf{f}(\mathbf{w}')$. We have

$$\begin{aligned} & d_{\mathbf{f}}(\mathbf{u}, \mathbf{w}) - d_{\mathbf{f}}(\mathbf{u}, \mathbf{w}') \\ & = \frac{1}{2}\|\boldsymbol{\theta}\|_p^2 - \frac{1}{2}\|\boldsymbol{\theta}'\|_p^2 + \mathbf{u} \cdot (\boldsymbol{\theta}' - \boldsymbol{\theta}) \\ & \geq \eta(\hat{y} - y)\mathbf{w} \cdot \mathbf{x} + \mathbf{u} \cdot (\boldsymbol{\theta}' - \boldsymbol{\theta}) - \frac{\eta^2}{2}(y - \hat{y})^2(p-1)X_p^2 \\ & = \eta(\hat{y} - y)(\mathbf{w} \cdot \mathbf{x} - \mathbf{u} \cdot \mathbf{x}) - \frac{\eta^2}{2}(y - \hat{y})^2(p-1)X_p^2, \end{aligned}$$

where the first equality follows from (4) and Lemma 1, part 3, the inequality derives from Lemma 4, and the second equality is obtained by applying the identity $\boldsymbol{\theta}' = \boldsymbol{\theta} - \eta(\hat{y} - y)\mathbf{x}$. Thus, in order to prove the claim, it suffices to show that

$$\begin{aligned} & \frac{c}{1+c} \frac{1}{(p-1)X_p^2} \frac{1}{2}(y - \hat{y})^2 - \frac{c}{(p-1)X_p^2} \frac{1}{2}(y - \mathbf{u} \cdot \mathbf{x})^2 \\ & \leq \eta(\hat{y} - y)(\mathbf{w} \cdot \mathbf{x} - \mathbf{u} \cdot \mathbf{x}) - \frac{\eta^2}{2}(y - \hat{y})^2(p-1)X_p^2, \end{aligned}$$

with $\eta = \frac{c}{(1+c)(p-1)X_p^2}$, $c \geq 0$. But this is formally equivalent to Lemma 5.1 in (Kivinen & Warmuth, 1997), where X^2 there is here $(p-1)X_p^2/2$ and b there is here $\frac{c}{(p-1)X_p^2}$. Following their proof gives the desired inequality. $\square$

Lemma 6. *Let $\mathbf{u}, \mathbf{w}, \mathbf{x} \in \mathcal{R}^n$, $a \in \mathcal{R}$ and set $\mathbf{w}' = \mathbf{f}^{-1}(\mathbf{f}(\mathbf{w}) + a\mathbf{x})$. Then the following equality holds:*

$$a(\mathbf{u} \cdot \mathbf{x} - \mathbf{w} \cdot \mathbf{x}) = d_{\mathbf{f}}(\mathbf{u}, \mathbf{w}) - d_{\mathbf{f}}(\mathbf{u}, \mathbf{w}') + d_{\mathbf{f}}(\mathbf{w}, \mathbf{w}').$$

Proof: We have:

$$\begin{aligned} a(\mathbf{u} \cdot \mathbf{x} - \mathbf{w} \cdot \mathbf{x}) & = (\mathbf{u} - \mathbf{w}) \cdot (\mathbf{f}(\mathbf{w}') - \mathbf{f}(\mathbf{w})) \\ & = d_{\mathbf{f}}(\mathbf{u}, \mathbf{w}) - d_{\mathbf{f}}(\mathbf{u}, \mathbf{w}') + d_{\mathbf{f}}(\mathbf{w}, \mathbf{w}'), \end{aligned}$$

where the first equality follows from the definition of $\mathbf{w}'$ and the second immediately derives from the general definition of $d_{\mathbf{f}}(\cdot, \cdot)$ given in (3). $\square$

In the rest of this section we treat linear regression and deterministic binary classification. It is worth noting the different role played by the learning rate η in the two settings. In the

regression setting η is a fundamental scaling factor (Theorem 7). In the classification setting, on the other hand, if the algorithm starts with $\mathbf{w}_1 = \mathbf{0}$ then η turns out to be immaterial (Theorem 8).

4.1. Linear regression

The theorem that follows gives the dual norms bound for the p -norm algorithms. It follows fairly directly from a combination of results from Grove, Littlestone, and Schuurmans (2001), and from Kivinen and Warmuth (1997), together with the observation that to obtain the regression algorithms one needs to define the measure of progress through a squared p -norm, as discussed in the previous section.

Theorem 7. *Let $\mathcal{X} = \{\mathbf{x} \in \mathcal{R}^n : \|\mathbf{x}\|_p \leq X_p\}$ and $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_T, y_T)) \in (\mathcal{X} \times \mathcal{R})^T$. For any $\mathbf{u} \in \mathcal{W} = \mathcal{R}^n$ the p -norm regression algorithm with $\eta = \frac{c}{(1+c)(p-1)X_p^2}$, $c > 0$ achieves the following loss bound:*

$$\sum_{i=1}^T \frac{1}{2} (y_i - \hat{y}_i)^2 \leq (1+c) \sum_{i=1}^T \frac{1}{2} (y_i - \mathbf{u} \cdot \mathbf{x}_i)^2 + \frac{1+c}{c} (p-1) X_p^2 d_{\mathbf{f}}(\mathbf{u}, \mathbf{w}_1).$$

In particular, when $\mathbf{w}_1 = \mathbf{0}$ we have:

$$\sum_{i=1}^T \frac{1}{2} (y_i - \hat{y}_i)^2 \leq (1+c) \sum_{i=1}^T \frac{1}{2} (y_i - \mathbf{u} \cdot \mathbf{x}_i)^2 + \frac{1+c}{2c} (p-1) X_p^2 \|\mathbf{u}\|_q^2.$$

Proof: We sum over all trials $t = 1, \dots, T$ the inequality of Lemma 4, drop the non-negative term $d_{\mathbf{f}}(\mathbf{u}, \mathbf{w}_{T+1})$ and solve the resulting inequality for $\sum_{i=1}^T \frac{1}{2} (y_i - \hat{y}_i)^2$. This yields the first bound. The second bound immediately follows from the first one and the fact that $d_{\mathbf{f}}(\mathbf{u}, \mathbf{0}) = \frac{1}{2} \|\mathbf{u}\|_q^2$ (see (4)). $\square$

4.2. Deterministic binary classification

Theorem 8 below gives a bound on the number of mistakes of the p -norm algorithm for deterministic binary classification. The theorem extends to the non-separable case a result Grove, Littlestone, and Schuurmans (2001) give for the linearly separable case only. The bound is in terms of the *deviation* $D_{\gamma}(\mathbf{u}; (\mathbf{x}_t, y_t))$ of a linear threshold classifier $\mathbf{u}$ with threshold 0 on example $(\mathbf{x}_t, y_t)$ w.r.t. margin $\gamma > 0$ (e.g., Freund & Schapire, 1999), defined as

$$D_{\gamma}(\mathbf{u}; (\mathbf{x}_t, y_t)) = \max\{0, \gamma - y_t \mathbf{u} \cdot \mathbf{x}_t\}.$$

Here γ is a free parameter. Notice that if $y_t \mathbf{u} \cdot \mathbf{x}_t > 0$, i.e., if $\mathbf{u}$ classifies $(\mathbf{x}_t, y_t)$ correctly then we can pick $\gamma < y_t \mathbf{u} \cdot \mathbf{x}_t$, thereby making $D_{\gamma}(\mathbf{u}; (\mathbf{x}_t, y_t)) = 0$.

Theorem 8. *Let $\mathcal{X} = \mathcal{R}^n$, $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_T, y_T)) \in (\mathcal{X} \times \{-1, +1\})^T$, $\mathcal{M}$ be the set of trials where the classification algorithm makes a mistake, when it is run on S ,*

and $X \geq X_{p,\mathcal{M}} = \max_{t \in \mathcal{M}} \|\mathbf{x}_t\|_p$. For any $\eta > 0$ and any $\mathbf{u} \in \mathcal{W} = \mathcal{R}^n$ the p -norm classification algorithm achieves the following bound on the number of mistakes, holding for any $\gamma > 0$:

$$|\mathcal{M}| \leq \frac{1}{\gamma} \sum_{t \in \mathcal{M}} D_\gamma(\mathbf{u}; (\mathbf{x}_t, y_t)) - \frac{\mathbf{u} \cdot \mathbf{f}(\mathbf{w}_1)}{2\eta\gamma} + \frac{B}{2\gamma^2} + \frac{1}{2\gamma^2} \sqrt{4\gamma B \sum_{t \in \mathcal{M}} D_\gamma(\mathbf{u}; (\mathbf{x}_t, y_t)) + B^2 - \frac{2\gamma B \mathbf{u} \cdot \mathbf{f}(\mathbf{w}_1)}{\eta} + \frac{\|\mathbf{w}_1\|_q^2 U^2 \gamma^2}{\eta^2}}, \quad (5)$$

where $B = (p-1) X^2 U^2$ and $U \geq \|\mathbf{u}\|_q$. Note that when $\mathbf{w}_1 = \mathbf{0}$, the bound is independent of η .

Proof: First note that the expression under the radical sign can be written as

$$4\gamma B \sum_{t \in \mathcal{M}} D_\gamma(\mathbf{u}; (\mathbf{x}_t, y_t)) + \left(B - \frac{\gamma \mathbf{u} \cdot \mathbf{f}(\mathbf{w}_1)}{\eta} \right)^2 + \frac{\gamma^2}{\eta^2} (\|\mathbf{f}(\mathbf{w}_1)\|_p^2 U^2 - (\mathbf{u} \cdot \mathbf{f}(\mathbf{w}_1))^2).$$

By Hölder's inequality, $\|\mathbf{f}(\mathbf{w}_1)\|_p U \geq \|\mathbf{f}(\mathbf{w}_1)\|_p \|\mathbf{u}\|_q \geq |\mathbf{u} \cdot \mathbf{f}(\mathbf{w}_1)|$. Thus the expression under the radical sign is non-negative. Observe that if $|\mathcal{M}| \leq \frac{1}{\gamma} \sum_{t \in \mathcal{M}} D_\gamma(\mathbf{u}; (\mathbf{x}_t, y_t)) - \frac{\mathbf{u} \cdot \mathbf{f}(\mathbf{w}_1)}{2\eta\gamma}$ then (5) trivially holds. Hence we continue by assuming $|\mathcal{M}| > \frac{1}{\gamma} \sum_{t \in \mathcal{M}} D_\gamma(\mathbf{u}; (\mathbf{x}_t, y_t)) - \frac{\mathbf{u} \cdot \mathbf{f}(\mathbf{w}_1)}{2\eta\gamma}$. We use Lemma 6 with $a = -\eta(\hat{y}_t - y_t)$ and $\mathbf{u}$ replaced by $\lambda \mathbf{u}$, where $\mathbf{u} \in \mathcal{R}^n$ and λ is a positive scalar to be optimized. The weights are changed only in trials $t \in \mathcal{M}$. In such trials $\hat{y}_t - y_t = -2y_t$ and $y_t \mathbf{w}_t \cdot \mathbf{x}_t \leq 0$. This yields:

$$2\lambda y_t \mathbf{u} \cdot \mathbf{x}_t \leq \frac{1}{\eta} (d_{\mathbf{f}}(\lambda \mathbf{u}, \mathbf{w}_t) - d_{\mathbf{f}}(\lambda \mathbf{u}, \mathbf{w}_{t+1}) + d_{\mathbf{f}}(\mathbf{w}_t, \mathbf{w}_{t+1})).$$

Set now for brevity

$$\hat{\gamma}_{\mathbf{u}, \mathcal{M}} = \frac{1}{|\mathcal{M}|} \sum_{t \in \mathcal{M}} y_t \mathbf{u} \cdot \mathbf{x}_t.$$

We sum over $t \in \mathcal{M}$ the previous inequality, use Lemma 4 with $\|\mathbf{x}_t\|_p \leq X_{p,\mathcal{M}}$, and apply the definition of $\hat{\gamma}_{\mathbf{u}, \mathcal{M}}$. We obtain:

$$\begin{aligned} 2\lambda \hat{\gamma}_{\mathbf{u}, \mathcal{M}} |\mathcal{M}| &\leq \frac{1}{\eta} (d_{\mathbf{f}}(\lambda \mathbf{u}, \mathbf{w}_1) - d_{\mathbf{f}}(\lambda \mathbf{u}, \mathbf{w}_{T+1}) + 2\eta^2 (p-1) X_{p,\mathcal{M}}^2 |\mathcal{M}|) \\ &\leq \frac{1}{\eta} \left(\frac{\lambda^2 \|\mathbf{u}\|_q^2 + \|\mathbf{w}_1\|_q^2}{2} - \lambda \mathbf{u} \cdot \mathbf{f}(\mathbf{w}_1) + 2\eta^2 (p-1) X_{p,\mathcal{M}}^2 |\mathcal{M}| \right), \end{aligned}$$

where in the second inequality we have dropped the non-negative term $d_f(\lambda \mathbf{u}, \mathbf{w}_{T+1})$ and used Eq. (4) for $d_f(\lambda \mathbf{u}, \mathbf{w}_1)$. Note also that the inequality $\gamma - \frac{1}{|\mathcal{M}|} \sum_{t \in \mathcal{M}} D_\gamma(\mathbf{u}; (\mathbf{x}_t, y_t)) \leq \hat{\gamma}_{\mathbf{u}, \mathcal{M}}$ holds for any positive γ and immediately derives from the definitions of $D_\gamma(\mathbf{u}; (\mathbf{x}_t, y_t))$ and $\hat{\gamma}_{\mathbf{u}, \mathcal{M}}$. Thus, for any given value of $\lambda > 0$, a bound on the size of $\mathcal{M}$ is given by the infimum of the $M > 0$ for which

$$\begin{aligned} & \frac{1}{\eta} \left(\frac{\lambda^2 \|\mathbf{u}\|_q^2 + \|\mathbf{w}_1\|_q^2}{2} - \lambda \mathbf{u} \cdot \mathbf{f}(\mathbf{w}_1) + 2\eta^2 (p-1) X_{p, \mathcal{M}}^2 M \right) \\ & - 2\lambda \gamma M + 2\lambda \sum_{t \in \mathcal{M}} D_\gamma(\mathbf{u}; (\mathbf{x}_t, y_t)) \end{aligned} \quad (6)$$

is negative. Furthermore, we obtain a valid bound if instead of (6), we work with the expression obtained by substituting U for $\|\mathbf{u}\|_q$ and X for $X_{p, \mathcal{M}}$, for any $U \geq \|\mathbf{u}\|_q$ and $X \geq X_{p, \mathcal{M}}$; we refer to the resulting expression as (6'). The minimum of (6') as a function of λ occurs at

$$\lambda = \frac{\mathbf{u} \cdot \mathbf{f}(\mathbf{w}_1) - 2\eta \sum_{t \in \mathcal{M}} D_\gamma(\mathbf{u}; (\mathbf{x}_t, y_t)) + 2M\gamma\eta}{U^2}$$

which is positive by our assumption that $|\mathcal{M}| > \frac{1}{\gamma} \sum_{t \in \mathcal{M}} D_\gamma(\mathbf{u}; (\mathbf{x}_t, y_t)) - \frac{\mathbf{u} \cdot \mathbf{f}(\mathbf{w}_1)}{2\eta\gamma}$. At the minimum as a function of M , the value of (6') equals

$$\frac{\|\mathbf{w}_1\|_q^2}{2\eta} + 2\eta(p-1)X^2M - \frac{(\mathbf{u} \cdot \mathbf{f}(\mathbf{w}_1) - 2\eta \sum_{t \in \mathcal{M}} D_\gamma(\mathbf{u}; (\mathbf{x}_t, y_t)) + 2M\gamma\eta)^2}{2\eta U^2}. \quad (7)$$

This is quadratic in M with negative leading coefficient, and it is positive at $M = 0$ since $U\|\mathbf{w}_1\|_q \geq \|\mathbf{u}\|_q \|\mathbf{f}(\mathbf{w}_1)\|_p \geq \mathbf{u} \cdot \mathbf{f}(\mathbf{w}_1)$ by Hölder's inequality. Thus the infimum of the M that make it negative is the larger of the roots of the equation obtained by setting it to 0. This gives us the bound of the theorem. $\square$

From the proof above the reader can see that Expression (7) is the same as the expression derived at the same stage in the equivalent analysis contained in the proof of Theorem 7.1 by Grove, Littlestone, and Schuurmans (2001), except for our inclusion of an extra term for the noise. In particular, without noise the equivalent of the $\mathbf{u} \cdot \mathbf{z}_0$ occurring in Grove, Littlestone, and Schuurmans (2001) would be $\mathbf{u} \cdot \mathbf{f}(\mathbf{w}_1)$; with noise, this gets replaced with $\mathbf{u} \cdot \mathbf{f}(\mathbf{w}_1) - 2\eta \sum_{t \in \mathcal{M}} D_\gamma(\mathbf{u}; (\mathbf{x}_t, y_t))$. The route to obtain this expression that we have chosen here helps to show the close relationship between the bounds for the regression and the classification cases.

A mistake bound for the (2-norm) Perceptron algorithm in the non-separable case has also been given by Freund and Schapire (1999). This bound uses the deviation *squared* to measure the margins, and thus is not comparable to (5) when $p = q = 2$.

We will apply Theorem 8 in Section 5.2 to proving robust bounds for learning r -of- k threshold functions with the p -norm algorithms.

5. Logarithmic behavior

In this section we show that for suitable choices of p (i.e., of the link function $\mathbf{f}$) the p -norm algorithm has loss bounds that grow logarithmically in n . Such loss bounds are typically associated with the multiplicative algorithms.

This section is organized as follows. In Section 5.1 we apply Theorem 7 to prove a bound on the regression algorithm. This bound is similar to bounds proven for algorithms in the EG-family (Kivinen & Warmuth, 1997). In Section 5.2 we give interesting applications of Theorem 8. Here we have also included a discussion of logarithmic behavior for the deterministic binary classification case. It seems easier to illustrate this point in the classification setting than in the regression setting. Finally, in Section 5.3 we introduce and analyze the randomized binary classification algorithm for an infinite pool of experts. Here we make heavy use of the logarithmic properties of the p -norm link function $\mathbf{f}$.

5.1. Linear regression

We have the following simple consequence of Theorem 7.

Corollary 9. *Let $\mathcal{X} = \{\mathbf{x} \in \mathcal{R}^n : \|\mathbf{x}\|_\infty \leq X_\infty\}$ and $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_T, y_T)) \in (\mathcal{X} \times \mathcal{R})^T$. For any $\mathbf{u} \in \mathcal{W} = \mathcal{R}^n$ the p -norm regression algorithm with⁸ $p = 2 \ln n$, $n \geq 3$, $\mathbf{w}_1 = \mathbf{0}$ and $\eta = \frac{c}{2e(1+c)(\ln n)X_\infty^2}$, $c > 0$ achieves the following loss bound:*

$$\sum_{i=1}^T \frac{1}{2} (y_i - \hat{y}_i)^2 \leq (1+c) \sum_{i=1}^T \frac{1}{2} (y_i - \mathbf{u} \cdot \mathbf{x}_i)^2 + \frac{e(1+c)}{c} (\ln n) X_\infty^2 \|\mathbf{u}\|_1^2. \quad (8)$$

Proof: We apply the second bound of Theorem 7. We observe that if $\|\mathbf{x}_t\|_\infty \leq X_\infty$ then $\|\mathbf{x}_t\|_p^2 = (\sum_i |x_{t,i}|^p)^{\frac{2}{p}} \leq (n X_\infty^p)^{\frac{2}{p}} = n^{\frac{2}{p}} X_\infty^2$. Hence we can upper bound the factor $(p-1) X_p^2 \|\mathbf{u}\|_q^2$ occurring in the bound of Theorem 7 by $(p-1) X_p^2 \|\mathbf{u}\|_q^2 \leq (p-1) n^{\frac{2}{p}} X_\infty^2 \|\mathbf{u}\|_q^2 \leq (p-1) n^{\frac{2}{p}} X_\infty^2 \|\mathbf{u}\|_1^2$. Next, we want to pick the best p . A derivative argument shows that the minimal value of $(p-1) n^{\frac{2}{p}}$ as a function of $p \geq 2$ is achieved by

$$p = \begin{cases} \ln n + \sqrt{\ln^2 n - 2 \ln n}, & \text{for } n \geq e^2; \\ 2, & \text{for } n < e^2. \end{cases}$$

Setting $p = 2 \ln n$ is therefore suboptimal, but it both simplifies the calculations and it differs from the optimal one by lower order terms only. (The condition $n \geq 3$ is needed here to insure $p \geq 2$.) With this setting we have $(p-1) n^{\frac{2}{p}} = (2 \ln n - 1) e < 2e \ln n$. Using the second bound of Theorem 7 with $\eta = \frac{c}{2e(1+c)(\ln n)X_\infty^2}$ gives (8). $\square$

Bounds similar to (8) have also been proven by Kivinen and Warmuth (1997) for the EG $^\pm$ and the EGU algorithms (Theorems 5.11 and 5.16 therein, respectively), and by Bylander (1997) for a “multiplicative” version of the BEG algorithm (Theorem 7 therein). We point

out that all such bounds are comparatively less robust than (8). In particular, $\text{EG}^\pm$ requires the prior knowledge of (an upper bound on) $\|\mathbf{u}\|_1$. EGU is useful when the 1-norm $\|\mathbf{u}\|_1$ of the comparison vector $\mathbf{u}$ is unknown, but the involved analysis shown in Kivinen and Warmuth (1997) holds only in the special case where both the labels y_t and the components $x_{t,i}$ of the instances $\mathbf{x}_t$ are non-negative and have a known upper bound. Finally, the “multiplicative” BEG requires relevant side information to tune the learning rate η and, in addition, the bound proven for it is worse than (8). (However, for the sake of comparison, observe that our square loss $\frac{1}{2}(y_t - \hat{y}_t)^2$ is half the square loss employed by Kivinen and Warmuth, and Bylander.)

5.2. Deterministic binary classification

For the deterministic binary classification case we have the following.

Corollary 10. *Let $\mathcal{X} = \mathcal{R}^n$, $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_T, y_T)) \in (\mathcal{X} \times \{-1, +1\})^T$, $\mathcal{M}$ be the set of trials where the classification algorithm makes a mistake, when it is run on S , and $X_{\infty, \mathcal{M}} = \max_{t \in \mathcal{M}} \|\mathbf{x}_t\|_\infty$. Suppose that $p = c \ln n$ for some $c > 0$ such that $p \geq 2$ and that we choose the initial weight vector $\mathbf{w}_1$ such that $\mathbf{f}(\mathbf{w}_1) = (\alpha, \dots, \alpha)$ for some $\alpha > 0$. Then for any $\eta > 0$ and any $\mathbf{u} \in \mathcal{W} = \mathcal{R}^n$ the p -norm classification algorithm achieves the following bound on the number of mistakes, holding for any $\gamma > 0$:*

$$|\mathcal{M}| \leq \frac{1}{\gamma} \sum_{t \in \mathcal{M}} D_\gamma(\mathbf{u}; (\mathbf{x}_t, y_t)) - \frac{\mathbf{u} \cdot \mathbf{f}(\mathbf{w}_1)}{2\eta\gamma} + \frac{B}{2\gamma^2} \\ + \frac{1}{2\gamma^2} \sqrt{4\gamma B \sum_{t \in \mathcal{M}} D_\gamma(\mathbf{u}; (\mathbf{x}_t, y_t)) + B^2 - \frac{2\gamma B \mathbf{u} \cdot \mathbf{f}(\mathbf{w}_1)}{\eta} + \frac{\alpha^2 e^{2/c} \|\mathbf{u}\|_1^2 \gamma^2}{\eta^2}},$$

where $B = (c \ln n - 1)e^{2/c} X_{\infty, \mathcal{M}}^2 \|\mathbf{u}\|_1^2 = O((\ln n) X_{\infty, \mathcal{M}}^2 \|\mathbf{u}\|_1^2)$.

Proof: Note that $\|\mathbf{u}\|_q \leq \|\mathbf{u}\|_1$ and for any $\mathbf{z}$, $\|\mathbf{z}\|_p \leq n^{1/p} \|\mathbf{z}\|_\infty$. Thus $X_{p, \mathcal{M}}$ in the statement of Theorem 8 is upper bounded by $n^{1/p} X_{\infty, \mathcal{M}}$ and we can set $U = \|\mathbf{u}\|_1$ and $X = n^{1/p} X_{\infty, \mathcal{M}}$ in that theorem. We also have $\|\mathbf{w}_1\|_q = \|\mathbf{f}(\mathbf{w}_1)\|_p = \alpha n^{1/p}$. Choose $p = c \ln n$ for some $c > 0$ such that $p \geq 2$. Then the quantity B of Theorem 8 becomes $B = (c \ln n - 1)n^{2/(c \ln n)} X_{\infty, \mathcal{M}}^2 \|\mathbf{u}\|_1^2 = (c \ln n - 1)e^{2/c} X_{\infty, \mathcal{M}}^2 \|\mathbf{u}\|_1^2$. Furthermore, $\|\mathbf{w}_1\|_q = \alpha e^{1/c}$. This leads to the desired bound. $\square$

If, as we vary n , we fix the other parameters appearing in the above bound then the bound grows as $O(\log n)$. Furthermore, the bound continues to grow logarithmically if η varies as $1/p$ or if instead α grows linearly with p .

There are thus two known ways that the p -norm algorithms lead to bounds that are logarithmic in n . One is to let p grow in proportion to $\log n$, with an arbitrary starting weight vector. The other, discussed in Grove, Littlestone, and Schuurmans (2001) is to let p go to infinity with a non-negative starting weight vector, non-negative components of $\mathbf{u}$, and a sufficiently small value of the ratio η/α . This ratio must decrease as $1/p$ and also must

Figure 1. Loss bounds as a function of p and of the scaling of the initial weight vector, for two different non-negative target vectors $\mathbf{u}$, for deterministic binary classification. Here $\theta_1 = \mathbf{f}(\mathbf{w}_1) = (\alpha, \dots, \alpha)$. Thus when $\alpha = 0$ the initial weights are 0. We hold η fixed (at 1). (Varying α is equivalent to fixing the initial weight vector at a non-zero value and varying η . The case $\alpha = 0$ corresponds to $\eta = \infty$.) The figure shows contour plots of the loss bounds as a function of p (for $p \geq 2$) and α for two different target vectors $\mathbf{u}$. Here $n = 1000$, the attributes are all ± 1 , the target in the left-hand plot is $\mathbf{u} = (1, 0, \dots, 0)$ and the other is $\mathbf{u} = (1, 1, 1, 0, \dots, 0)$. There is no noise.

decrease as the size of the *average margin* of $\mathbf{u}$ over the learner's mistaken trials decreases. (The average margin occurs explicitly in the analysis by Gentile and Warmuth (1999) and does also appear in the proof of Theorem 8, where it is denoted by $\hat{\gamma}_{\mathbf{u}, \mathcal{M}}$.) This margin can be assumed to be γ in the noise-free case, but also in general depends on the amount of noise, which is not necessarily known to the learner.

Figure 1 gives two examples of how the bound of Theorem 8 varies as p is varied and as the scaling of the initial weight vector is varied for particular values of the other parameters. The region where the bounds are logarithmic in n lies in the valleys in those plots. We can see that when $\alpha = 0$ the minimum of the loss bounds occurs in both cases when p is roughly 10. The bounds decrease somewhat from there as p grows if α also grows at the right rate. The correct rate depends on $\mathbf{u}$ and differs in the two plots. Thus the best value of α (or, equivalently, the best multiplier η if α is fixed) for a given p depends on the unknown target $\mathbf{u}$. Note also that the bound rises steeply as p decreases to 2.

From results in Grove, Littlestone, and Schuurmans (2001) we can calculate the values we would get in the limiting case as p goes to infinity with α varying optimally as a function of p . This limiting bound is roughly 14 in the case of the left-hand plot, and roughly 105 for the right-hand plot. One can also show that in these cases, the limiting bound is the best bound. Thus at the contour interval (of the light-weight contours) of the plots, no lower contour would appear in either plot no matter how much one extended them.

We next present a specific application of Theorem 8. We consider the problem of learning r -of- k threshold functions over the boolean domain $\mathcal{X} = \{0, 1\}^n$. For $r \leq k \leq n$, an r -of- k threshold function $\mathbf{g}$ over $\{0, 1\}^n$ is specified by k *relevant* attributes $x_{i_1}, x_{i_2}, \dots, x_{i_k}$, where $1 \leq i_j \leq n$. The function $\mathbf{g}$ is +1 on instance $\mathbf{x}_t \in \mathcal{X}$ if and only if at least r of the k relevant attributes of $\mathbf{x}_t$ are 1, $\mathbf{g}$ is -1 otherwise.⁹ When $r = 1$ we have the class of monotone k -literal disjunctions, while $r = k$ yields the class of monotone k -literal conjunctions.

These comparison classes have been studied, for example, by Littlestone (1988, 1991) and Auer and Warmuth (1998). These authors show that a multiplicative algorithm, such as the Winnow algorithm, substantially outperforms an additive algorithm such as the Perceptron algorithm, when the target function has few relevant attributes (i.e., when k is small compared to n) and the instances have sufficiently many non-zero attributes. When $k = r = 1$ the r -of- k threshold functions reduce to n single variables or *experts* (see, e.g., Littlestone, 1989b; Vovk, 1990; Littlestone & Warmuth, 1994; Cesa-Bianchi et al., 1997).

Following Littlestone (1991) and Auer and Warmuth (1998), we define the *attribute error* $a_{g,t}$ of function g on example $(\mathbf{x}_t, y_t) \in \mathcal{X} \times \{-1, +1\}$ as the minimal number of components of the boolean vector $\mathbf{x}_t$ that need to be flipped to make g classify $(\mathbf{x}_t, y_t)$ correctly.

Throughout the rest of this section $a_{g,\mathcal{M}}$ will denote $\sum_{t \in \mathcal{M}} a_{g,t}$, where $\mathcal{M}$ is a given sequence of trials. Observe that when $k = r = 1$ (the binary expert setting) the attribute error $a_{g,\mathcal{M}}$ represents the number of prediction *mistakes* of expert g in trial sequence $\mathcal{M}$.

Remark 1. For simplicity, in parts 1 and 2 of the following corollary we apply Theorem 8 with $\mathbf{w}_1 = \mathbf{0}$. As a matter of fact, as figure 1 shows, better choices for $\mathbf{w}_1$ exist. These lead to some improvement in the constants. A careful tuning of $\mathbf{w}_1$ is done only in part 3. Typically, a good choice is to let $\mathbf{w}_1$ have all positive components. On the other hand, $\mathbf{w}_1 = \mathbf{0}$ has the advantage of yielding an algorithm that handles target weights of both signs equally well, without needing the doubling of the number of attributes typically done for the multiplicative algorithms (Littlestone, 1988, 1989).

Corollary 11. *Let $\mathcal{X} = \{0, 1\}^n$, $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_T, y_T)) \in (\mathcal{X} \times \{-1, +1\})^T$ and $\mathcal{M}$ be the set of trials where the classification algorithm makes a mistake, when it is run on S . For any r -of- k threshold function g the p -norm classification algorithm achieves the following mistake bounds:*

1. *If $p = 2 \ln(n+1)$, $n \geq 2$, $\mathbf{w}_1 = \mathbf{0}$ and $\eta > 0$ then*

$$|\mathcal{M}| \leq 2a_{g,\mathcal{M}} + e(k + |k - 2r + 1|)^2 \ln n + 2\sqrt{e} \sqrt{a_{g,\mathcal{M}}(k + |k - 2r + 1|)^2 \ln n + \frac{e}{4}(k + |k - 2r + 1|)^4 \ln^2 n}.$$

2. *If $p = 2 \ln \frac{n+|k-2r+1|}{k+|k-2r+1|}$, $ek + (e-1)|k-2r+1| \leq n$, $\mathbf{w}_1 = \mathbf{0}$ and $\eta > 0$ then*

$$|\mathcal{M}| \leq 2a_{g,\mathcal{M}} + \frac{B}{2} + \frac{1}{2} \sqrt{8a_{g,\mathcal{M}}B + B^2},$$

where

$$B = 2e(k + |k - 2r + 1|)^2 \ln \frac{n + |k - 2r + 1|}{\sqrt{e}(k + |k - 2r + 1|)}.$$

3. *If $p = 2 \ln n$, $n \geq 3$, $k = r = 1$ (the binary expert setting), $\mathbf{w}_1$ is such that $\mathbf{f}(\mathbf{w}_1) = (1, \dots, 1)$ and $\eta = 1/(2(p-1))$ then*

$$|\mathcal{M}| \leq 2a_{g,\mathcal{M}} + (e-2) \ln n + 2\sqrt{e} \sqrt{a_{g,\mathcal{M}} \ln n + \frac{e}{4} \ln^2 n}.$$

Proof: Here we adopt the notation introduced in this section and apply Theorem 8 with $\gamma = 1$. We observe that when $\mathbf{w}_1 = \mathbf{0}$ in (5) the tightest bound results when B is as small as possible. Thus in parts 1 and 2 we represent r -of- k threshold functions and select p such that B is small. In addition, in part 3 we need to take care of the ratio $\mathbf{f}(\mathbf{w}_1)/\eta$.

1. We represent an r -of- k threshold function g on $\mathcal{X}$ as a linear threshold function $\mathbf{u}_g$ with threshold 0 on $\mathcal{X}' = \{-1, +1\}^{n+1}$. First we map $\mathbf{x}_t \in \mathcal{X}$ into $\mathbf{x}'_t \in \mathcal{X}'$: we turn to -1 the components of $\mathbf{x}_t$ of value 0, and we pad the resulting vector with an extra component of value $+1$. Then we let $\mathbf{v}_g$ be an n -dimensional $\{0, 1\}$ -vector where the components set to 1 correspond to the relevant attributes of g . Vector $\mathbf{u}_g$ is obtained by padding $\mathbf{v}_g$ with an $n+1$ 'st component of value $k-2r+1$. It is not hard to see that g is $+1$ on $\mathbf{x}_t$ if and only if $\mathbf{u}_g \cdot \mathbf{x}'_t \geq 1$ and that g is -1 on $\mathbf{x}_t$ if and only if $\mathbf{u}_g \cdot \mathbf{x}'_t \leq -1$. Also, with this representation we have $a_{g,t} = D_1(\mathbf{u}_g; (\mathbf{x}'_t, y_t))/2$, and thus

$$2a_{g,\mathcal{M}} = \sum_{t \in \mathcal{M}} D_1(\mathbf{u}_g; (\mathbf{x}'_t, y_t)). \quad (9)$$

Then we apply Theorem 8 on the trial sequence $((\mathbf{x}'_1, y_1), \dots, (\mathbf{x}'_T, y_T))$ and the comparison class $\{\mathbf{u}_g\}$. By the above representation we can write $X_{p,\mathcal{M}} = (n+1)^{1/p}$ and $\|\mathbf{u}_g\|_q \leq \|\mathbf{u}_g\|_1 = k + |k-2r+1|$. Therefore in this case B is at most $(p-1)(n+1)^{2/p}(k+|k-2r+1|)^2$. We need to minimize B as a function of $p \geq 2$. As in the proof of Corollary 9, setting $p = 2 \ln(n+1)$ is slightly sub-optimal, but it simplifies the algebra. (The hypothesis $n \geq 2$ is needed to insure $p \geq 2$.) This choice of p gives $(p-1)(n+1)^{2/p} = e(2 \ln(n+1) - 1) < 2e \ln n$, when $n \geq 2$. Hence

$$B \leq 2e(k + |k-2r+1|)^2 \ln n.$$

Using this upper bound for B in (5) with $\mathbf{w}_1 = \mathbf{0}$ along with (9) proves part 1.

2. Here we slightly modify the representation of r -of- k threshold functions we adopted in part 1. We “spread” the threshold $k-2r+1$ that was previously put into the $n+1$ -st component of $\mathbf{u}_g$ into $|k-2r+1|$ threshold components. If $k-2r+1 \geq 0$ these threshold components have value 1, while if $k-2r+1 \leq -1$ they have value -1 . Accordingly, we map $\mathbf{x}_t \in \mathcal{X}$ into $\mathbf{x}'_t \in \{-1, +1\}^{n+|k-2r+1|}$ by turning to -1 the components of $\mathbf{x}_t$ of value 0, and by padding the resulting vector with $|k-2r+1|$ ones. Obviously, (9) still holds. Now in Theorem 8 we can write $X_{p,\mathcal{M}} = (n+|k-2r+1|)^{1/p}$ and $\|\mathbf{u}_r\|_q = (k+|k-2r+1|)^{1/q}$. Hence in this case

$$B = (p-1)(n+|k-2r+1|)^{2/p}(k+|k-2r+1|)^{2/q}.$$

Again, in order to make B small, setting $p = 2 \ln \frac{n+|k-2r+1|}{k+|k-2r+1|}$ is a good trade-off between optimality and simplicity. (The assumption $e k + (e-1)|k-2r+1| \leq n$ prevents $p < 2$.) The reader can verify that our choice of p yields

$$B = 2e(k + |k-2r+1|)^2 \ln \frac{n + |k-2r+1|}{\sqrt{e}(k + |k-2r+1|)}.$$

Using such B in (5) with $\mathbf{w}_1 = \mathbf{0}$ along with (9) gives part 2.

3. We set $k = r = 1$ in the representation conventions of part 2. This yields $B = 2e \ln \frac{n}{\sqrt{e}}$. In applying (5) we note that $\frac{u \cdot f(\mathbf{w}_1)}{\eta} = 2(p-1) = 2(2 \ln n - 1)$ and that $\|\mathbf{w}_1\|_q^2 = n^{2/p} = e$. The resulting expression under the radical sign of (5) equals $16e a_{g,\mathcal{M}} \ln \frac{n}{\sqrt{e}} + 4e^2 \ln^2 \frac{n}{\sqrt{e}}$. The whole bound can be written as

$$|\mathcal{M}| \leq 2a_{g,\mathcal{M}} + (e-2) \ln \frac{n}{\sqrt{e}} + \sqrt{4e a_{g,\mathcal{M}} \ln \frac{n}{\sqrt{e}} + e^2 \ln^2 \frac{n}{\sqrt{e}}},$$

which implies the claimed inequality. $\square$

We now compare Corollary 11 to the state of the art in deterministic binary classification. Generally speaking, the bounds in parts 1 and 2 are not comparable to the corresponding bounds in Littlestone (1991), Theorem 5 therein and Auer and Warmuth (1998), the deterministic parts of Theorems 3, 4, and 5 therein. The difference between parts 1 and 2 of Corollary 11 lies in the fact that in part 2 we allow p to depend on k and r . In both cases we do not need to tune η as a function of $a_{g,\mathcal{M}}$. Unlike Littlestone's and Auer and Warmuth's, our bounds have the constant 2 in front of $a_{g,\mathcal{M}}$. However, we have a quadratic dependence on k for any r , whereas Littlestone and Auer and Warmuth's bounds are linear in k . (On the other hand, notice that the existing Winnow bounds for learning r -of- k threshold functions are linear in rk . Hence they are quadratic in k if, say, $r = k/2$.) We do not know whether there is a variant of the p -norm algorithm that has a linear dependence on k . We leave that as an open question.

Part 3 has slightly worse constants than those achieved by algorithms, such as the Binomial Weighting (Cesa-Bianchi et al., 1996) and the tuned Weighted Majority algorithms (Vovk, 1990; Cesa-Bianchi et al., 1997), that assume the prior knowledge of an upper bound K on the number of mistakes of the best expert. Bounds for algorithms that do not need K are provided by Cesa-Bianchi et al. (1997) and Cesa-Bianchi, Helmbold, and Panizza (1998). In the first paper the authors exploit a sophisticated doubling trick to repeatedly re-estimate the best η for Weighted Majority. But the constants they get are worse than those in part 3. In the second paper the authors elaborate an elegant bayesian interpretation of the problem of learning in the binary expert framework. The algorithm they analyze, though "asymptotically equivalent" to the bound proven for the Binomial Weighting algorithm, yields a bound which is actually worse than ours, unless $\ln n \gg a_{u,\mathcal{M}}$. Moreover, we observe that the p -norm bounds have a far easier derivation.

5.3. Randomized binary classification against an infinite set of experts

In this section we assume the comparison class is made up of a countably infinite set of experts. The labels y_t are binary ± 1 , but both the algorithm and the experts can output predictions lying in $[-1, +1]$. Prediction performance is measured by the absolute loss.

A *prior* distribution $\{W_i, i = 1, 2, \dots\}$, with $W_{i+1} \leq W_i$ for any i , is defined over the experts. The prior value measures the relative importance of experts, and might also be used to rank experts according to their intrinsic (descriptive) complexity. As a typical example, if we have an infinite sequence of increasingly complex prediction algorithms indexed by

the positive integers, we can view each algorithm as an expert and define a prior distribution $\{W_i, i = 1, 2, \dots\}$ such that the complexity of the i -th algorithm is $O(\ln \frac{1}{W_i})$. This might be seen as an on-line model selection problem with countably many models.¹⁰

We are looking for cumulative loss bounds of the form

$$L_A(S) \leq \min_{i=1,2,\dots} L_i(S) + c \ln \frac{1}{W_i} + g\left(L_i(S), \ln \frac{1}{W_i}\right), \quad (10)$$

where $L_A(S)$ is the cumulative absolute loss of algorithm A on trial sequence S , $L_i(S)$ is the cumulative absolute loss of the i -th expert on S , c is a positive constant, and g is some function which increases less than linearly in the two arguments.

We formalize the above as a problem of learning infinite-dimensional linear threshold functions. We set the comparison class $\mathcal{W}$ to $\{\mathbf{u} \in \mathcal{R}^\infty : \|\mathbf{u}\|_1 \leq 1\}$ (this includes the set of unit vectors, i.e., the set of “experts”), the instance space $\mathcal{X}$ to $[-1, +1]^\infty$, and assume that vector $\mathbf{u} \in \mathcal{W}$ predicts on instance $\mathbf{x}_t \in \mathcal{X}$ by the (infinite) dot product $\mathbf{u} \cdot \mathbf{x}_t \in [-1, 1]$. Vector $\mathbf{u}$ incurs loss $D_{\mathbf{u},t} = D_1(\mathbf{u}; (\mathbf{x}_t, y_t)) = \max\{0, 1 - y_t \mathbf{u} \cdot \mathbf{x}_t\}$. The quantity $\frac{1}{2} D_{\mathbf{u},t}$ turns out to be the absolute loss of $\mathbf{u}$ on example $(\mathbf{x}_t, y_t)$. For instance, if $\mathbf{u}$ is the i -th unit vector and $\mathbf{x}_t \in \mathcal{X}$, then the t -th prediction of the i -th expert is $x_{t,i} = \mathbf{u} \cdot \mathbf{x}_t$, and $\frac{1}{2} D_{\mathbf{u},t} = \frac{1}{2} |y_t - x_{t,i}| \in [0, 1]$ is exactly the absolute loss suffered by the i -th expert in the t -th trial.

Since we cannot keep track of vectors with infinitely many components, we resort to the standard finitization technique (e.g., Littlestone & Warmuth, 1994) of keeping an increasing initial segment of “active” experts. We call the set of active experts in a given trial the *active pool* of experts. Finitization is performed as follows. We are given a non-decreasing sequence of positive integers $\{N_t\}_{t \geq 1}$. The quantity N_t is the size of the active pool in trial t . If the learning algorithm incurs no (absolute) loss in trial t then $N_{t+1} = N_t$, otherwise $N_{t+1} \geq N_t$. The growth rate of the sequence $\{N_t\}_{t \geq 1}$ will be related to both the prior distribution $\{W_i\}$ and the current cumulative loss of the learning algorithm A . In every trial t we turn the instance $\mathbf{x}_t = (x_{t,1}, x_{t,2}, \dots) \in \mathcal{X}$ into $\mathbf{x}'_t = (x'_{t,1}, x'_{t,2}, \dots) \in \mathcal{X}$, where $x'_{t,i} = x_{t,i}$ if $i \leq N_t$ and $x'_{t,i} = 0$ otherwise. Therefore when expert i is not in the active pool in trial t we assume that its prediction is 0. In other words, we consider only the predictions of experts $i = 1, 2, \dots, N_t$, and disregard those of experts $i = N_t + 1, N_t + 2, \dots$.

Before describing the algorithm, we need to introduce some short-hand notation. We allow the link function $\mathbf{f}$ to change over time. We use the subscript t for $\mathbf{f}$, $\mathbf{f}^{-1}$ and the dual norm values p and q . We also use the short-hand $d_t(\mathbf{u}, \mathbf{w})$ for $d_{\mathbf{f}_t}(\mathbf{u}, \mathbf{w})$. Hence (recall (4))

$$d_t(\mathbf{u}, \mathbf{w}) = \frac{1}{2} \|\mathbf{u}\|_{q_t}^2 + \frac{1}{2} \|\mathbf{w}\|_{q_t}^2 - \mathbf{u} \cdot \mathbf{f}_t(\mathbf{w}),$$

where q_t and $\mathbf{f}_t$ depend on time.

The algorithm is given in Table 2, where we denote by l_t the absolute loss the algorithm incurs in trial t and by L_t the cumulative absolute loss of the algorithm up to trial t , i.e., $L_t = \sum_{i=1}^t l_i$, with $L_0 = 0$. The algorithm is a so-called self-confident algorithm (Auer & Gentile, 2000), since it is also aimed at learning on the fly the best learning rate η_t . As in Auer and Gentile (2000), η_t scales with the current cumulative loss of the algorithm. The

Table 2. The p -norm algorithm for randomized binary classification against an infinite set of experts.

Input parameter sequence: $\{N_t\}_{t \geq 1}$ nondecreasing sequence of positive integers.

Initialization: Initial weight vector $\mathbf{w}_1 = \mathbf{0}$.

For $t = 1, 2, \dots, T$:

- Get instance $\mathbf{x}_t \in [-1, +1]^\infty$;
- Turn $\mathbf{x}_t$ into $\mathbf{x}'_t \in [-1, +1]^\infty$ such that $x'_{t,i} = \begin{cases} x_{t,i} & \text{if } i \leq N_t, \\ 0 & \text{if } i > N_t; \end{cases}$
- Let

$$p_t = 2 \ln N_t, \\ \alpha_t = \sqrt{\frac{(p_t-1)e}{(p_t-1)e + L_{t-1} + 1}};$$

- Predict with $\hat{y}_t = \begin{cases} 1 & \text{if } \mathbf{w}_t \cdot \mathbf{x}'_t \geq 1 - \alpha_t, \\ \frac{\mathbf{w}_t \cdot \mathbf{x}'_t}{1 - \alpha_t} & \text{if } \mathbf{w}_t \cdot \mathbf{x}'_t \in (\alpha_t - 1, 1 - \alpha_t), \\ -1 & \text{if } \mathbf{w}_t \cdot \mathbf{x}'_t \leq \alpha_t - 1; \end{cases}$

- Get label $y_t \in \{-1, +1\}$;

Incur loss $l_t = L(y_t, \hat{y}_t) = \frac{1}{2} |y_t - \hat{y}_t| \in [0, 1]$;

- **If** $l_t > 0$ **then** update weights as follows:

$$\mathbf{w}_t^m = \mathbf{f}_t^{-1}(\mathbf{f}_t(\mathbf{w}_t) + \eta_t y_t \mathbf{x}'_t), \\ \mathbf{w}_{t+1} = \mathbf{w}_t^m / \max\{1, \|\mathbf{w}_t^m\|_{q_t}\},$$

where

$$\eta_t = \frac{2\alpha_t}{(p_t-1)e} = \frac{2}{\sqrt{(p_t-1)e}} \frac{1}{\sqrt{(p_t-1)e + L_{t-1} + 1}}, \\ q_t = \frac{p_t}{p_t-1}.$$

Figure 2. The finite slope sigmum function. Here $\alpha_t \in (0, 1)$.

prediction function $\hat{y}_t = \hat{y}_t(\mathbf{w}_t \cdot \mathbf{x}'_t)$ is a finite slope sigmum function whose knots $\pm(1 - \alpha_t)$ change over time. Figure 2 gives a graphical representation.

The algorithm maintains (in principle) an infinite-dimensional weight vector. The starting weight vector is $\mathbf{w}_1 = \mathbf{0}$. In the generic trial t the algorithm turns the instance $\mathbf{x}_t$ into $\mathbf{x}'_t$ as indicated. Then it predicts the label $y_t \in \{-1, +1\}$ associated with $\mathbf{x}_t$ (or $\mathbf{x}'_t$) through

the finite slope signum function, as specified in Table 2. Only the first N_t components of $\mathbf{x}'_t$ can be nonzero. When the label y_t is received, the algorithm incurs absolute loss $l_t = \frac{1}{2}|y_t - \hat{y}_t| \in [0, 1]$. Finally, the algorithm updates its weights as indicated. The update has two steps. The first step computes $\mathbf{w}_t^m$ by the conventional update of the p -norm algorithms, as in Section 3 (notice, however, that both the link function $\mathbf{f}_t$ and the learning rate η_t depend on time). The second step computes $\mathbf{w}_{t+1}$ by *projecting* $\mathbf{w}_t^m$ onto $\mathcal{W}$ w.r.t. $d_t(\cdot, \cdot)$. The weights are not updated if $l_t = 0$. Observe that in any trial t only a finite number of components of $\mathbf{w}_t$ are nonzero. In particular, only the first N_t components of $\mathbf{w}_{t+1}$ can be nonzero. Thus all vectors occurring in Table 2 can actually be considered finite-dimensional.

Analyzing this algorithm needs some care, since both the learning rate and the link function change from trial to trial. We are required to extend the self-confident technique described by Auer and Gentile (2000) to handle this more general situation. The analysis is based on the following three technical lemmas. Lemma 12 is taken from Auer and Gentile (2000). Its proof is repeated here for completeness. Lemmas 13 and 14 will be proven in the appendix. Lemma 13 will be used to bound differences arising from time-varying Bregman divergences. The bounds so obtained are in turn bounded from above through Lemma 14. These bounds are meaningful only when $|b - a|$ there is small.

Lemma 12. *Let $l_{j_1}, l_{j_2}, \dots, l_{j_T} \geq 0$ and $\delta > 0$. Then*

$$\sum_{t=1}^T \frac{l_{j_t}}{\sqrt{\delta + \sum_{i=1}^t l_{j_i}}} \leq 2 \left(\sqrt{\delta + \sum_{t=1}^T l_{j_t}} - \sqrt{\delta} \right).$$

Proof: Let $l_{j_0} = \delta$ and $L_t = \sum_{i=0}^t l_{j_i}$, $t = 0, \dots, T$. In the inequality $\frac{1}{2}x \leq 1 - \sqrt{1-x}$, $x \leq 1$, set $x = \frac{l_{j_t}}{L_t}$ and then multiply both terms of the resulting inequality by $\sqrt{L_t}$. This yields

$$\frac{1}{2} \frac{l_{j_t}}{\sqrt{L_t}} \leq \sqrt{L_t} - \sqrt{L_{t-1}}.$$

Summing over $t = 1, \dots, T$ gives the desired inequality. $\square$

Lemma 13. *Let a, b, A, B be positive real numbers with $a, b > 1$. Then*

$$\begin{aligned} & \max_{0 \leq x \leq 1} (x^{a-1} A - x^{b-1} B) \\ & \leq \max \left\{ 0, A - B, \frac{B^{\frac{a-1}{a-b}}}{A^{\frac{b-1}{a-b}}} \left(\left(\frac{a-1}{b-1} \right)^{\frac{a-1}{b-a}} - \left(\frac{a-1}{b-1} \right)^{\frac{b-1}{b-a}} \right) \right\}. \end{aligned}$$

Lemma 14. *Let $\mathbf{w} \in \mathcal{R}^n$ and let a, b be real numbers with $2 > b > a > 1$.*

1. *If $4(\ln n) \frac{b-a}{ab} \leq 1$ holds then*

$$\max_{\mathbf{w}: \|\mathbf{w}\|_b \leq 1} (\|\mathbf{w}\|_a^2 - \|\mathbf{w}\|_b^2) \leq 4(\ln n) \frac{b-a}{ab}.$$

2.

$$\max_{\mathbf{w}: \|\mathbf{w}\|_b \leq 1} (\|\mathbf{w}\|_b^{2-b} - \|\mathbf{w}\|_a^{2-a}) \leq e^{-1} \frac{b-a}{2-b}.$$

3.

$$\max_{\mathbf{w}: \|\mathbf{w}\|_b \leq 1} (\|\mathbf{w}\|_a^{2-a} - \|\mathbf{w}\|_b^{2-b}) \leq n^{b-a} (\ln n) (b-a).$$

4. Let $A = \|\mathbf{w}\|_a^{2-a}$ and $B = \|\mathbf{w}\|_b^{2-b}$. Then

$$\max_{\mathbf{w}: \|\mathbf{w}\|_b \leq 1} \frac{B^{\frac{a-1}{a-b}}}{A^{\frac{b-1}{a-b}}} \leq n^{b-1}.$$

5.

$$\left(\frac{a-1}{b-1}\right)^{\frac{a-1}{b-a}} - \left(\frac{a-1}{b-1}\right)^{\frac{b-1}{b-a}} \leq \frac{b-a}{a-1}.$$

We are now ready to prove Theorem 15, the main result of this subsection. We have tried to keep the proof as simple as possible, without paying any special attention to minimizing constants. The bound contained in Theorem 15 is in implicit form, since N_t depends on the loss L_t . For technical reasons, we have also included a trial parameter t_0 . A loss bound in explicit form will be provided by Corollary 16, where the role of t_0 is clarified.

Theorem 15. Let $\mathcal{X} = [-1, +1]^\infty$, $\{N_t\}_{t \geq 1}$ be a sequence of positive integers such that $N_{t+1} \geq N_t$, $N_1 \geq 4$ and $N_{t+1} \leq e^{-1} N_t^2$. Let $\mathcal{W} = \{\mathbf{u} \in \mathcal{R}^\infty : \|\mathbf{u}\|_1 \leq 1\}$ and $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_T, y_T)) \in (\mathcal{X} \times \{-1, +1\})^T$. Then for any $\mathbf{u} \in \mathcal{W}$ and any $t_0 \in \{0, 1, \dots, T\}$ the algorithm in Table 2 has cumulative absolute loss L_T satisfying the following inequality:

$$\begin{aligned} L_T &\leq \frac{1}{2} D_{\mathbf{u}, (t_0+1) \dots T} + L_{t_0} + \sqrt{112 p_T \left(\frac{1}{2} D_{\mathbf{u}, (t_0+1) \dots T} + L_{t_0} \right)} + 135 p_T + 2e \\ &= \frac{1}{2} D_{\mathbf{u}, (t_0+1) \dots T} + L_{t_0} + O(\sqrt{\ln N_T (D_{\mathbf{u}, (t_0+1) \dots T} + L_{t_0})} + \ln N_T), \end{aligned} \quad (11)$$

where

$$\begin{aligned} L_{t_0} &= \sum_{t=1}^{t_0} l_t = \sum_{t=1}^{t_0} \frac{1}{2} |y_t - \hat{y}_t|, \\ D_{\mathbf{u}, (t_0+1) \dots T} &= \sum_{t=t_0+1, l_t > 0}^T D_{\mathbf{u}, t}, \\ D_{\mathbf{u}, t} &= D_1(\mathbf{u}; (\mathbf{x}'_t, y_t)) = \max\{0, 1 - y_t \mathbf{u} \cdot \mathbf{x}'_t\} = |y_t - \mathbf{u} \cdot \mathbf{x}'_t|, \\ p_T &= 2 \ln N_T. \end{aligned}$$

Proof: The proof is a more sophisticated version of the proof of Theorem 9 in Auer and Gentile (2000). We denote by $\mathcal{M}$ the set of trials where the algorithm incurs a nonzero

loss. Let us focus on a single trial $t \in \mathcal{M}$ and let $\mathbf{w}_t^m$ be as in Table 2. We apply Lemma 6 with $\mathbf{w} = \mathbf{w}_t$, $\mathbf{w}' = \mathbf{w}_t^m$, $\mathbf{x} = \mathbf{x}'_t$, $a = \eta_t y_t$, and we upper bound the last term $d_t(\mathbf{w}_t, \mathbf{w}_t^m)$ by Lemma 4 (recall that $|x'_{t,i}| \leq 1$ for $i \leq N_t$ and $x'_{t,i} = 0$ if $i > N_t$): $d_t(\mathbf{w}_t, \mathbf{w}_t^m) \leq \frac{\eta_t^2}{2}(p_t - 1)N_t^{2/p_t} = \frac{\eta_t^2}{2}(p_t - 1)e$. This yields

$$y_t (\mathbf{u} \cdot \mathbf{x}'_t - \mathbf{w}_t \cdot \mathbf{x}'_t) \leq \frac{1}{\eta_t} \left(d_t(\mathbf{u}, \mathbf{w}_t) - d_t(\mathbf{u}, \mathbf{w}_t^m) + \frac{\eta_t^2}{2}(p_t - 1)e \right).$$

From the definition of $D_{u,t}$ it follows that $y_t \mathbf{u} \cdot \mathbf{x}'_t \geq 1 - D_{u,t}$. Then, by Lemma 2, we lower bound¹¹ $d_t(\mathbf{u}, \mathbf{w}_t^m)$ through $d_t(\mathbf{u}, \mathbf{w}_{t+1})$ and rearrange:

$$1 - y_t \mathbf{w}_t \cdot \mathbf{x}'_t - \frac{\eta_t}{2}(p_t - 1)e \leq D_{u,t} + \frac{1}{\eta_t} (d_t(\mathbf{u}, \mathbf{w}_t) - d_t(\mathbf{u}, \mathbf{w}_{t+1})).$$

Next, observe that $\frac{\eta_t}{2}(p_t - 1)e = \alpha_t$. Now, as in Theorem 9 of Auer and Gentile (2000), one can easily show that for any trial t such that $l_t > 0$ the LHS of the last inequality is at least $2(1 - \alpha_t)l_t$. This gives us the one-trial loss bound

$$2(1 - \alpha_t)l_t \leq D_{u,t} + \frac{1}{\eta_t}(d_t(\mathbf{u}, \mathbf{w}_t) - d_t(\mathbf{u}, \mathbf{w}_{t+1})), \quad (12)$$

holding for any t such that $l_t > 0$ and any $\mathbf{u} \in \mathcal{W}$. Our aim is to obtain telescoping sums from (12). The RHS of (12) is treated as follows.

$$\begin{aligned} \frac{1}{\eta_t}(d_t(\mathbf{u}, \mathbf{w}_t) - d_t(\mathbf{u}, \mathbf{w}_{t+1})) &= \frac{1}{\eta_t} d_t(\mathbf{u}, \mathbf{w}_t) - \frac{1}{\eta_{t+1}} d_{t+1}(\mathbf{u}, \mathbf{w}_{t+1}) \\ &\quad + \frac{1}{\eta_{t+1}} (d_{t+1}(\mathbf{u}, \mathbf{w}_{t+1}) - d_t(\mathbf{u}, \mathbf{w}_{t+1})) \\ &\quad + \left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t} \right) d_t(\mathbf{u}, \mathbf{w}_{t+1}). \end{aligned}$$

Now, the first difference in the RHS of the last equation is telescoping, the second one will be upper bounded in a moment, while the third one is upper bounded by $2(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t})$ (from Lemma 3 and the fact that $\eta_{t+1} \leq \eta_t$).

We need to upper bound the second difference. We will make use of Lemmas 13 and 14. From the alternate definition of d_t given in (4) we have

$$\begin{aligned} &d_{t+1}(\mathbf{u}, \mathbf{w}_{t+1}) - d_t(\mathbf{u}, \mathbf{w}_{t+1}) \\ &= \frac{1}{2} \|\mathbf{w}_{t+1}\|_{q_{t+1}}^2 - \frac{1}{2} \|\mathbf{w}_{t+1}\|_{q_t}^2 + \frac{1}{2} \|\mathbf{u}\|_{q_{t+1}}^2 - \frac{1}{2} \|\mathbf{u}\|_{q_t}^2 \\ &\quad + \mathbf{u} \cdot \mathbf{f}_t(\mathbf{w}_{t+1}) - \mathbf{u} \cdot \mathbf{f}_{t+1}(\mathbf{w}_{t+1}) \\ &\leq \frac{1}{2} \|\mathbf{w}_{t+1}\|_{q_{t+1}}^2 - \frac{1}{2} \|\mathbf{w}_{t+1}\|_{q_t}^2 + \frac{1}{2} \|\mathbf{u}\|_{q_{t+1}}^2 - \frac{1}{2} \|\mathbf{u}\|_{q_t}^2 \\ &\quad + \|\mathbf{u}\|_1 \|\mathbf{f}_t(\mathbf{w}_{t+1}) - \mathbf{f}_{t+1}(\mathbf{w}_{t+1})\|_\infty \\ &\leq \frac{1}{2} \|\mathbf{w}_{t+1}\|_{q_{t+1}}^2 - \frac{1}{2} \|\mathbf{w}_{t+1}\|_{q_t}^2 + \frac{1}{2} \|\mathbf{u}\|_{q_{t+1}}^2 - \frac{1}{2} \|\mathbf{u}\|_{q_t}^2 \\ &\quad + \|\mathbf{f}_t(\mathbf{w}_{t+1}) - \mathbf{f}_{t+1}(\mathbf{w}_{t+1})\|_\infty. \end{aligned} \quad (13)$$

The second difference in the right-most side of (13) is telescoping, thus we only need to upper bound

$$\frac{1}{2} \|\mathbf{w}_{t+1}\|_{q_{t+1}}^2 - \frac{1}{2} \|\mathbf{w}_{t+1}\|_{q_t}^2 \quad (14)$$

and

$$\|\mathbf{f}_t(\mathbf{w}_{t+1}) - \mathbf{f}_{t+1}(\mathbf{w}_{t+1})\|_\infty. \quad (15)$$

First of all, observe that $\|\mathbf{w}_{t+1}\|_{q_t} \leq 1$. To bound (14) we use Lemma 14, part 1, with $b = q_t$, $a = q_{t+1}$, $\mathbf{w} = \mathbf{w}_{t+1}$ and $n = N_t$. Since $q_i = p_i/(p_i - 1)$ and $p_i = 2 \ln N_i$, $i = t, t + 1$, the condition $4(\ln n) \frac{b-a}{ab} \leq 1$ there can be seen to be equivalent to $N_{t+1} \leq N_t^2$. Under this condition we have

$$(14) \leq 2 \frac{p_{t+1} - p_t}{p_{t+1}}.$$

To bound (15) we proceed as follows. Let $\mathbf{f}_t = (f_{t,1}, f_{t,2}, \dots)$. We have

$$\begin{aligned} f_{t,i}(\mathbf{w}_{t+1}) - f_{t+1,i}(\mathbf{w}_{t+1}) &= \text{sign}(w_{t+1,i}) |w_{t+1,i}|^{q_t-1} \|\mathbf{w}_{t+1}\|_{q_t}^{2-q_t} \\ &\quad - \text{sign}(w_{t+1,i}) |w_{t+1,i}|^{q_{t+1}-1} \|\mathbf{w}_{t+1}\|_{q_{t+1}}^{2-q_{t+1}}. \end{aligned} \quad (16)$$

We fix component i and distinguish the two cases $\text{sign}(w_{t+1,i}) = +1$ and $\text{sign}(w_{t+1,i}) = -1$. Notice that if we have an upper bound on (16) for both values of $\text{sign}(w_{t+1,i})$ then we also have an upper bound on the absolute value of (16). Specifically, we will be applying the following simple argument: If a real number x is such that $x \leq X^+$ and $-x \leq X^-$, where $X^+, X^- > 0$ then $|x| \leq \max\{X^+, X^-\}$.

If $\text{sign}(w_{t+1,i}) = +1$ then

$$(16) = |w_{t+1,i}|^{q_t-1} \|\mathbf{w}_{t+1}\|_{q_t}^{2-q_t} - |w_{t+1,i}|^{q_{t+1}-1} \|\mathbf{w}_{t+1}\|_{q_{t+1}}^{2-q_{t+1}}.$$

Since $|w_{t+1,i}| \leq 1$ we can apply Lemma 13 with $x = |w_{t+1,i}|$, $A = \|\mathbf{w}_{t+1}\|_{q_t}^{2-q_t}$, $B = \|\mathbf{w}_{t+1}\|_{q_{t+1}}^{2-q_{t+1}}$, $a = q_t$ and $b = q_{t+1}$. Notice that the third argument of the maximum in the RHS of Lemma 13 is negative (since $a > b$). Therefore we can come to consider the second term only, namely, $A - B$ (which is positive). In turn, this is bounded through Lemma 14, part 2, where now $a = q_{t+1}$ and $b = q_t$. Hence, if $\text{sign}(w_{t+1,i}) = +1$ we can write

$$(16) \leq e^{-1} \frac{q_t - q_{t+1}}{2 - q_t} \leq q_t - q_{t+1},$$

where the last inequality holds if $N_t \geq 4$.

If $\text{sign}(w_{t+1,i}) = -1$ then

$$(16) = |w_{t+1,i}|^{q_{t+1}-1} \|\mathbf{w}_{t+1}\|_{q_{t+1}}^{2-q_{t+1}} - |w_{t+1,i}|^{q_t-1} \|\mathbf{w}_{t+1}\|_{q_t}^{2-q_t}.$$

We can apply Lemma 13 with $x = |w_{t+1,i}|$, $A = \|\mathbf{w}_{t+1}\|_{q_{t+1}}^{2-q_{t+1}}$, $B = \|\mathbf{w}_{t+1}\|_{q_t}^{2-q_t}$, $a = q_{t+1}$ and $b = q_t$. The resulting upper bound is again bounded by Lemma 14. In particular:

- from Lemma 14, part 3 we have

$$\begin{aligned} \|\mathbf{w}_{t+1}\|_{q_{t+1}}^{2-q_{t+1}} - \|\mathbf{w}_{t+1}\|_{q_t}^{2-q_t} &\leq N_t^{q_t-q_{t+1}} (\ln N_t) (q_t - q_{t+1}) \\ &\leq e \frac{p_{t+1} - p_t}{p_{t+1} - 1}, \end{aligned}$$

where one can verify that the last inequality holds if $N_t \geq 3$;

- from Lemma 14, part 4 we have

$$\left(\|\mathbf{w}_{t+1}\|_{q_t}^{2-q_t}\right)^{\frac{q_{t+1}-1}{q_{t+1}-q_t}} / \left(\|\mathbf{w}_{t+1}\|_{q_{t+1}}^{2-q_{t+1}}\right)^{\frac{q_t-1}{q_{t+1}-q_t}} \leq N_t^{q_t-1} \leq e,$$

where the last inequality holds if $N_t \geq 3$;

- from Lemma 14, part 5 we have

$$\left(\frac{q_{t+1}-1}{q_t-1}\right)^{\frac{q_{t+1}-1}{q_t-q_{t+1}}} - \left(\frac{q_{t+1}-1}{q_t-1}\right)^{\frac{q_t-1}{q_t-q_{t+1}}} \leq \frac{q_t - q_{t+1}}{q_{t+1} - 1} = \frac{p_{t+1} - p_t}{p_t - 1}.$$

Hence, if $\text{sign}(w_{t+1,i}) = -1$ then

$$(16) \leq \max \left\{ e \frac{p_{t+1} - p_t}{p_{t+1} - 1}, e \frac{p_{t+1} - p_t}{p_t - 1} \right\} = e \frac{p_{t+1} - p_t}{p_t - 1},$$

as long as $N_t \geq 3$.

Since the previous argument holds for any component i , we conclude that $N_t \geq 4$ implies

$$(15) \leq \max \left\{ q_t - q_{t+1}, e \frac{p_{t+1} - p_t}{p_t - 1} \right\} = e \frac{p_{t+1} - p_t}{p_t - 1}.$$

We plug the bounds on (14) and (15) back into (13) and rewrite (12). We obtain the following main inequality:

$$\begin{aligned} 2(1 - \alpha_t) l_t &\leq D_{\mathbf{u},t} + \frac{1}{\eta_t} d_t(\mathbf{u}, \mathbf{w}_t) - \frac{1}{\eta_{t+1}} d_{t+1}(\mathbf{u}, \mathbf{w}_{t+1}) \\ &\quad + \frac{1}{\eta_{t+1}} \left(2 \frac{p_{t+1} - p_t}{p_{t+1}} + \frac{1}{2} \|\mathbf{u}\|_{q_{t+1}}^2 - \frac{1}{2} \|\mathbf{u}\|_{q_t}^2 + e \frac{p_{t+1} - p_t}{p_t - 1} \right) \\ &\quad + 2 \left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t} \right). \end{aligned} \tag{17}$$

The RHS of this inequality is not yet telescoping, so further manipulation is needed. In particular, we need to find suitable upper bounds on the terms in (17). We can write:

$$\begin{aligned} \frac{2}{\eta_{t+1}} \left(\frac{p_{t+1} - p_t}{p_{t+1}} \right) &= \sqrt{e} \sqrt{(p_{t+1} - 1)e + L_t + 1} \sqrt{p_{t+1} - 1} \left(\frac{p_{t+1} - p_t}{p_{t+1}} \right) \\ &\leq \sqrt{e} \sqrt{(p_{t+1} - 1)e + L_t + 1} \frac{p_{t+1} - p_t}{\sqrt{p_{t+1}}} \\ &\leq 2\sqrt{e} \sqrt{(p_{t+1} - 1)e + L_t + 1} (\sqrt{p_{t+1}} - \sqrt{p_t}) \\ &\leq 2\sqrt{e} \sqrt{(p_{T+1} - 1)e + L_T + 1} (\sqrt{p_{t+1}} - \sqrt{p_t}), \end{aligned}$$

where the second inequality uses $\frac{b-a}{\sqrt{b}} \leq 2(\sqrt{b} - \sqrt{a})$, for $b \geq a > 0$;

$$\begin{aligned} \frac{1}{\eta_{t+1}} \left(\frac{1}{2} \|\mathbf{u}\|_{q_{t+1}}^2 - \frac{1}{2} \|\mathbf{u}\|_{q_t}^2 \right) &\leq \frac{\frac{1}{2} \|\mathbf{u}\|_{q_{t+1}}^2}{\eta_{t+1}} - \frac{\frac{1}{2} \|\mathbf{u}\|_{q_t}^2}{\eta_t}; \\ \frac{e}{\eta_{t+1}} \left(\frac{p_{t+1} - p_t}{p_t - 1} \right) &= \frac{e\sqrt{e}}{2} \sqrt{(p_{t+1} - 1)e + L_t + 1} \frac{\sqrt{p_{t+1} - 1}}{p_t - 1} (p_{t+1} - p_t) \\ &\leq e\sqrt{e} \sqrt{(p_{t+1} - 1)e + L_t + 1} \frac{(p_{t+1} - p_t)}{\sqrt{p_{t+1}}} \\ &\leq 2e\sqrt{e} \sqrt{(p_{t+1} - 1)e + L_t + 1} (\sqrt{p_{t+1}} - \sqrt{p_t}) \\ &\leq 2e\sqrt{e} \sqrt{(p_{T+1} - 1)e + L_T + 1} (\sqrt{p_{t+1}} - \sqrt{p_t}), \end{aligned}$$

where the first inequality holds when $N_{t+1} \leq e^{-1} N_t^2$ and the second one uses again $\frac{b-a}{\sqrt{b}} \leq 2(\sqrt{b} - \sqrt{a})$.

We plug back into (17), sum the main inequality over all $t \in \mathcal{M}$ such that $t \geq t_0 + 1$, drop nonpositive terms from the RHS, upper bound $d_{(t_0+1)}(\mathbf{u}, \mathbf{w}_{(t_0+1)})$ by 2 (recall Lemma 3), upper bound $\|\mathbf{u}\|_{q_{T+1}}^2$ by $\|\mathbf{u}\|_1^2 \leq 1$, and rearrange. We yield:

$$\begin{aligned} &\sum_{t \in \mathcal{M}, t \geq t_0+1} (1 - \alpha_t) l_t \\ &\leq \frac{1}{2} D_{\mathbf{u}, (t_0+1) \dots T} + \sqrt{e}(e+1) \sqrt{(p_{T+1} - 1)e + L_T + 1} \sqrt{p_{T+1}} + \frac{5}{4} \frac{1}{\eta_{T+1}}. \end{aligned} \tag{18}$$

Notice that the terms $\sqrt{e}(e+1) \sqrt{(p_{T+1} - 1)e + L_T + 1} \sqrt{p_{T+1}}$ and $\frac{5}{4} \frac{1}{\eta_{T+1}}$ in the RHS of (18) do not depend on t_0 . This turns out to be a key feature in the analysis.

We bound the LHS of (18) from below using Lemma 12. We can write:

$$\begin{aligned} \sum_{t \in \mathcal{M}, t \geq t_0+1} \alpha_t l_t &= \sum_{t \in \mathcal{M}, t \geq t_0+1} \frac{\sqrt{(p_t - 1)e} l_t}{\sqrt{(p_t - 1)e + L_{t-1} + 1}} \\ &\leq \sqrt{(p_T - 1)e} \sum_{t \in \mathcal{M}, t \geq t_0+1} \frac{l_t}{\sqrt{(p_T - 1)e + L_t}} \end{aligned}$$

$$\begin{aligned}
&\leq 2\sqrt{(p_T - 1)e} \left(\sqrt{(p_T - 1)e + L_T} - \sqrt{(p_T - 1)e + L_{t_0}} \right) \\
&\leq 2\sqrt{(p_T - 1)e} \left(\sqrt{(p_T - 1)e + L_T} - \sqrt{(p_T - 1)e} \right) \\
&= 2\sqrt{(p_T - 1)e} \sqrt{(p_T - 1)e + L_T} - 2(p_T - 1)e.
\end{aligned}$$

We also set the dummy value p_{T+1} to p_T (so that $\frac{1}{\eta_{T+1}} = \frac{\sqrt{(p_T - 1)e}}{2} \sqrt{(p_T - 1)e + L_T + 1}$). Substituting back into (18) and rearranging gives

$$\begin{aligned}
L_{(t_0+1)..T} &\leq \frac{1}{2} D_{\mathbf{u}, (t_0+1)..T} \\
&\quad + 2\sqrt{(p_T - 1)e} \sqrt{(p_T - 1)e + L_T} - 2(p_T - 1)e \\
&\quad + \left(\sqrt{e}(e + 1)\sqrt{p_T} + \frac{5}{8}\sqrt{(p_T - 1)e} \right) \sqrt{(p_T - 1)e + L_T + 1},
\end{aligned}$$

where we have set $L_{(t_0+1)..T} = \sum_{t \in \mathcal{M}, t \geq t_0+1} l_t$.

We overestimate the last square root by $\sqrt{(p_T - 1)e + L_T} + 1$, plug back and bound from above some of the resulting constants (using $p_T \geq 2$):

$$L_{(t_0+1)..T} \leq \frac{1}{2} D_{\mathbf{u}, (t_0+1)..T} + \frac{21}{2} \sqrt{p_T} \sqrt{(p_T - 1)e + L_T} + 2e.$$

Solving for $L_T = L_{t_0} + L_{(t_0+1)..T}$ and overestimating once more yields the desired bound. $\square$

Now, consider the case when the comparison vectors $\mathbf{u}$ are unit vectors. In order to obtain bounds of the form (10), we need to let N_t be an increasing function of L_t . From (11) the reader can see that a non-vacuous bound can be obtained only if $\ln N_t$ is *sub-linear* in L_t . Therefore we set $N_t = h(L_t)$, where $h(\cdot)$ is an invertible function such that $\ln h(x) = o(x)$ as $x \rightarrow \infty$. Also, comparing (10) and (11), we see that L_{t_0} is forced to be $O(\ln \frac{1}{W_i})$. We let $\mathbf{u}$ be the i -th unit vector and t_0 be the largest t such that $N_t < i$. This has two consequences:

1. The cumulative loss $D_{\mathbf{u}, (t_0+1)..T}$ can be upper bounded by the *true* (unobserved) cumulative loss $\sum_{t=1}^T D_1(\mathbf{u}, (\mathbf{x}_t, y_t)) = \sum_{t=1}^T |y_t - x_{t,i}|$ of the i -th expert over the whole trial sequence.
2. Since $N_{t_0} = h(L_{t_0}) = \Omega(i)$ we have $h^{-1}(i) = O(\ln \frac{1}{W_i})$. Hence the size of the active pool N_t is also determined by the prior distributions $\{W_i, i = 1, 2, \dots\}$. The faster W_i decays the slower N_t is allowed to increase. Notice that this setup cannot handle a polynomially decaying prior, such as $W_i = \frac{1}{i(i+1)}$, since this prior would force $\ln h$ to be a linear function.

The following corollary gives a specific example of the bounds which can be derived from Theorem 15. Here N_t grows linearly with L_t . This is useful when the prior has negative exponential form, i.e., when $W_i = 2^{-O(i)}$. Other trade-offs between the cumulative loss of the algorithm and the growth rate of the active pool are clearly possible. We refer the reader to Littlestone and Warmuth (1994) for further discussion.

Corollary 16. *Let $\mathcal{X} = [-1, +1]^\infty$ and $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_T, y_T)) \in (\mathcal{X} \times \{-1, +1\})^T$. Then for any expert i the algorithm in Table 2 with $N_t = \lceil L_{t-1} \rceil + 4, t = 1, 2, \dots$, achieves the following bound on the cumulative absolute loss:*

$$\begin{aligned} L_T &\leq L_i(S) + i + 83 \sqrt{224 (L_i(S) + i) \ln(L_i(S) + i + 2e + 5)} + 4550 + 2e \\ &= L_i(S) + i + O(\sqrt{(L_i(S) + i) \ln(L_i(S) + i)}), \end{aligned} \quad (19)$$

where

$$L_i(S) = \sum_{t=1}^T \frac{1}{2} |y_t - x_{t,i}|.$$

Proof: Fix an expert $i \in \{1, 2, \dots\}$. We apply Theorem 15 with t_0 as specified before. The reader can easily verify that the sequence N_t meets the requirements given in that theorem.

From the definitions of N_t and t_0 we have $L_{t_0} + 3 \leq N_{t_0} < i$. Hence $L_{t_0} < i - 3$ if $i \geq 4$ and $L_{t_0} = 0$ otherwise. Thus $L_{t_0} \leq i$ anyway. Also, since $N_T \leq L_T + 5$, the terms p_T in the bound of Theorem 15 are upper bounded by $2 \ln(L_T + 5)$. Moreover, if $\mathbf{u}$ in Theorem 15 denotes the i -th unit vector, we have $\frac{1}{2} D_{\mathbf{u}, (t_0+1) \dots T} \leq L_i(S)$. Therefore the bound of Theorem 15 implies the following:

$$L_T + 5 \leq L_i(S) + i + \sqrt{224 \ln(L_T + 5) (L_i(S) + i)} + 270 \ln(L_T + 5) + 2e + 5.$$

Setting for brevity

$$\begin{aligned} L &= L_T + 5 \\ A &= L_i(S) + i + 2e + 5 \\ B &= \sqrt{224 (L_i(S) + i)} \\ C &= 270, \end{aligned}$$

the last inequality becomes

$$L \leq A + B \sqrt{\ln L} + C \ln L. \quad (20)$$

We want to find an upper bound on the largest L such that the above is satisfied as an equality. We argue that

$$L^* = A + 83 B \sqrt{\ln A} + 4550$$

is such an upper bound (notice that this would prove (19)). We set $L = A + A'$ in (20) and upper bound the resulting term $B \sqrt{\ln(A + A')}$ as

$$\begin{aligned} B \sqrt{\ln(A + A')} &= B \sqrt{\ln A + \ln(1 + A'/A)} \\ &\leq B \sqrt{\ln A} + B \sqrt{\ln(1 + A'/A)} \end{aligned}$$

$$\begin{aligned}
&\leq B \sqrt{\ln A} + B \sqrt{A'}/\sqrt{A} \\
&\leq B \sqrt{\ln A} + \sqrt{224 A'},
\end{aligned}$$

where the first inequality uses $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$, for $a, b \geq 0$, and the second one derives from $\ln(1+a) \leq a$, for $a \geq 0$. We obtain

$$A' \leq B \sqrt{\ln A} + \sqrt{224 A'} + C \ln(A + A').$$

In order to prove our claim, it suffices to show that the two inequalities

$$\begin{aligned}
41 B \sqrt{\ln A} + 4550/2 &\geq \sqrt{224(83 B \sqrt{\ln A} + 4550)} \\
41 B \sqrt{\ln A} + 4550/2 &\geq C \ln(A + 83 B \sqrt{\ln A} + 4550)
\end{aligned}$$

hold true for any nonnegative value of the sum $L_i(S) + i$ (recall that A and B are functions of $L_i(S) + i$). Both inequalities can be proven in the following way. We first verify by direct inspection that the inequalities hold when $L_i(S) + i = 0$. Then, computing derivatives of both sides (w.r.t. $L_i(S) + i$), we check that the LHS grows faster than the corresponding RHS. The details are rather simple and are omitted. $\square$

We now compare the results of this subsection to the state of the art. To the best of our knowledge, bounds of the form (10) are new in the on-line setting we consider here. The problem of learning against an infinite set of experts has been investigated by Barzdin and Frievald (1972) and by Littlestone and Warmuth (1994). Barzdin and Frievald consider only the case when an expert is consistent with the trial sequence, whereas the bounds which can be proven through the techniques introduced by Littlestone and Warmuth are weaker than ours (they do not have the constant one in front of $L_i(S)$). On the other hand, such bounds hold for all prior distributions over the experts. As far as we understand, bounds of the form (10) can be obtained only when the problem of tuning the learning rate is somehow addressed. Learning rate tuning is not explored in Littlestone and Warmuth (1994). Extending the doubling strategy used in Cesa-Bianchi et al. (1997) to the infinite expert framework seems to be quite cumbersome (this is actually left as an open problem in that paper). In contrast to that, our self-confident approach is flexible enough to be applied to more general comparison classes and loss functions (Auer & Gentile, 2000).

6. Conclusions and open problems

In this paper we have studied the p -norm algorithms for (both deterministic and randomized) binary classification and linear regression. In the case of deterministic binary classification we pointed out that the p -norm algorithms are relatively insensitive to the tuning of their parameters. This makes them compare favourably with the algorithms in the “Winnow family” when applied to learning noisy data. In the case of randomized binary classification we have shown how to extend the p -norm algorithms to perform on-line model selection with a countably infinite set of models. The resulting algorithm is again less sensitive to

noise than algorithms in the “Winnow family”, such as the Weighted Majority algorithm. In linear regression we have provided the first analysis of the p -norm algorithms, which we feel provides an interesting and versatile basic on-line regression tool.

There are various directions in which this work could be extended. The bounds exhibited in Corollary 11, parts 1 and 2 have a quadratic dependence on k . We would like to see if there are efficient and simple algorithms yielding bounds of the form

$$|\mathcal{M}| \leq 2a_{g,\mathcal{M}} + c_1 k \ln n + c_2 \sqrt{a_{g,\mathcal{M}} k \ln n}.$$

where c_1 and c_2 can depend on r .

The self-confident algorithm we used in Section 5.3 keeps a weight vector with bounded norm. This makes the algorithm useful when the best expert shifts with time (Herbster & Warmuth, 1998a). We would like to see whether (and to what extent) in this framework a p -norm update rule outperforms a multiplicative rule.

Theorem 15 yields vacuous bounds when applied to polynomially decaying priors. It is not clear to us whether it is possible to obtain bounds of the form (10) holding for all prior distributions.

A competing algorithm for binary classification tasks that also does not need parameters to be set is a so-called apobayesian algorithm described by Littlestone and Mesterharm (1997)—it is closely related to the algorithm studied by Cesa-Bianchi, Helmbold, and Panizza (1998). That algorithm functions roughly as a variant of Winnow that learns its parameters, though the parameters do not appear as such in the algorithm. It is quite different from the p -norm algorithms, and it remains to be clarified how the capabilities of the two algorithms compare.

Appendix A: Proof of Lemma 13

An elementary analysis shows that the maximum on the LHS is achieved at either $x = 0$ or $x = 1$ or $x = (\frac{b-1}{a-1} \frac{B}{A})^{\frac{1}{a-b}}$. Substituting into $x^{a-1} A - x^{b-1} B$ yields the result.

Appendix B: Proof of Lemma 14

In parts 1–4 we proceed by induction on the dimension n . For $n = 1$ the assertions in parts 1, 3 and 4 are easily verified. To prove part 2 for $n = 1$, we upper bound $x^{2-b} - x^{2-a}$ by using the inequality $1 - e^{-x} \leq x$:

$$x^{2-b} - x^{2-a} = x^{2-b} (1 - e^{-\ln(1/x)(b-a)}) \leq x^{2-b} (\ln(1/x)) (b-a).$$

Now, $x^{2-b} \ln(1/x)$, viewed as a function of $x \in [0, 1]$, is maximized by $x = e^{-1/(2-b)}$. The maximum equals $\frac{e^{-1}}{2-b}$ giving the desired upper bound.

We assume that the assertions in parts 1–4 are true for $1, 2, \dots, n-1$ and continue by proving them for dimension n . We apply a simple Kuhn-Tucker analysis. This shows that in all four parts the maximum is achieved when all the nonzero components of $\mathbf{w}$ have a constant absolute value, i.e., when $\mathbf{w} = (w_1, \dots, w_n)$ is such that either $w_i = 0$ or $|w_i| = \beta > 0$, where β does not depend on i . However, any $w_i = 0$ at the maximum would

reduce the dimension n of the problem by one, and the assertions would then hold by the inductive hypothesis. Hence we can assume that in all four parts $\mathbf{w} = (\beta, \dots, \beta)$, $\beta > 0$. Since $\|\mathbf{w}\|_b = \beta n^{1/b}$ we can continue by maximizing over $\{\beta : 0 \leq \beta \leq n^{-1/b}\}$.

1. Since $a < b$ we have

$$\begin{aligned} \max_{\beta : 0 \leq \beta \leq n^{-1/b}} \beta^2 (n^{2/a} - n^{2/b}) &= n^{-2/b} (n^{2/a} - n^{2/b}) \\ &= e^{(\ln n)(2/a - 2/b)} - 1 \\ &\leq 4(\ln n) \frac{b-a}{ab}, \end{aligned}$$

where the last inequality uses $e^x - 1 \leq 2x$ for $x \in [0, 1/2]$.

2. We can write

$$\begin{aligned} &\max_{\beta : 0 \leq \beta \leq n^{-1/b}} (\beta^{2-b} n^{2/b-1} - \beta^{2-a} n^{2/a-1}) \\ &= \max_{\beta : 0 \leq \beta \leq n^{-1/b}} \beta^{2-b} n^{2/b-1} (1 - e^{(b-a) \ln \beta + (2/a - 2/b) \ln n}) \\ &\leq \max_{\beta : 0 \leq \beta \leq n^{-1/b}} \beta^{2-b} n^{2/b-1} ((b-a) \ln(1/\beta) - (2/a - 2/b) \ln n) \\ &\leq e^{-1} n^{-\frac{(2-a)(2-b)}{ab}} \frac{b-a}{2-b} \\ &\leq e^{-1} \frac{b-a}{2-b}, \end{aligned}$$

where in the first inequality we used $1 - e^x \leq -x$, and in the second inequality we exploited the fact that, viewed as a function of $\beta \in [0, 1]$, the expression $\beta^{2-b} ((b-a) \ln(1/\beta) - (2/a - 2/b) \ln n)$ is maximized when $\beta = e^{-\frac{1}{2-b}} n^{-\frac{2}{ab}}$.

3. We have:

$$\begin{aligned} &\max_{\beta : 0 \leq \beta \leq n^{-1/b}} (\beta^{2-a} n^{2/a-1} - \beta^{2-b} n^{2/b-1}) \\ &= \max_{\beta : 0 \leq \beta \leq n^{-1/b}} \beta^{2-a} n^{2/a-1} (1 - e^{-(b-a) \ln \beta - (2/a - 2/b) \ln n}) \\ &\leq \max_{\beta : 0 \leq \beta \leq n^{-1/b}} \beta^{2-a} n^{2/a-1} ((b-a) \ln \beta + (2/a - 2/b) \ln n), \end{aligned} \quad (21)$$

where the inequality uses again $1 - e^x \leq -x$. Now, consider the expression $\beta^{2-a} ((b-a) \ln \beta + (2/a - 2/b) \ln n)$, as a function of β . This function is negative for $0 \leq \beta \leq n^{-\frac{2}{ab}}$ and is increasing for $\beta \geq e^{-\frac{1}{2-a}} n^{-\frac{2}{ab}}$. Since $e^{-\frac{1}{2-a}} n^{-\frac{2}{ab}} < n^{-\frac{2}{ab}} < n^{-1/b}$ the maximum in (21) is achieved at $\beta = n^{-1/b}$. This gives

$$(21) = n^{\frac{(2-a)(b-a)}{ab}} (\ln n) \frac{(2-a)(b-a)}{ab} \leq n^{b-a} (\ln n) (b-a).$$

4. We can write

$$\begin{aligned}
 \ln(B^{\frac{a-1}{a-b}} / A^{\frac{b-1}{a-b}}) &= \frac{b-1}{b-a} \ln A - \frac{a-1}{b-a} \ln B \\
 &= \frac{b-1}{b-a} \left(\frac{2-a}{a} \ln n + (2-a) \ln \beta \right) \\
 &\quad - \frac{a-1}{b-a} \left(\frac{2-b}{b} \ln n + (2-b) \ln \beta \right) \\
 &= \frac{b(b-1)(2-a) - a(a-1)(2-b)}{(b-a)ab} (\ln n) + \ln \beta.
 \end{aligned}$$

The maximum over $\{\beta : 0 \leq \beta \leq n^{-1/b}\}$ is clearly achieved by $\beta = n^{-1/b}$. Substituting into the previous chain of equalities and simplifying gives

$$\max_{\beta : 0 \leq \beta \leq n^{-1/b}} \ln(B^{\frac{a-1}{a-b}} / A^{\frac{b-1}{a-b}}) = \frac{(2-a)(b-1)}{ab} \ln n \leq (b-1) \ln n.$$

This concludes the proofs of parts 1–4.

As far as part 5 is concerned, we can write

$$\begin{aligned}
 \left(\frac{a-1}{b-1} \right)^{\frac{a-1}{b-a}} - \left(\frac{a-1}{b-1} \right)^{\frac{b-1}{b-a}} &= \left(\frac{a-1}{b-1} \right)^{\frac{a-1}{b-a}} \left(\frac{b-a}{b-1} \right) \\
 &= \frac{(a-1)^{\frac{a-1}{b-a}}}{(b-1)^{\frac{b-1}{b-a}}} (b-a).
 \end{aligned}$$

We take the logarithm of the main fraction. We get

$$\begin{aligned}
 \ln \frac{(a-1)^{\frac{a-1}{b-a}}}{(b-1)^{\frac{b-1}{b-a}}} &= \frac{a-1}{b-a} \ln(a-1) - \frac{b-1}{b-a} \ln(b-1) \\
 &= \frac{1}{b-a} ((a-1) \ln(a-1) - (b-1) \ln(b-1)) \\
 &\leq -1 - \ln(a-1) \\
 &\leq -\ln(a-1),
 \end{aligned}$$

where the first inequality derives from a first-order Taylor approximation to the convex function $x \ln x$.

Acknowledgments

We are indebted to Nick Littlestone with whom we had the pleasure and the honor to work with in the preliminary version of this paper (Gentile & Littlestone, 1999). We have benefited from many discussions with Manfred Warmuth about these and related topics, and

we would like to thank him for valuable suggestions regarding this paper. We would also like to thank David Helmbold and Nicolò Cesa-Bianchi for useful discussions about the topics of Section 5.3. The author has been partially supported by ESPRIT Working Group EP 27150, Neural and Computational Learning II (NeuroColt II) and by MURST Cofin 99 (project: “Stochastic Processes with Spatial Structure”).

Notes

1. The analysis in Grove, Littlestone, and Schuurmans (2001) and here does not cover the original Winnow (Littlestone, 1988).
2. In the deterministic binary classification case, this failure is sometimes associated with the inability of the algorithm to even represent a perfect classifier with the weights available to it. Consider common versions of the multiplicative classification algorithm for learning boolean functions, where the changes to the weights all consist of multiplying or dividing by the same factor (when the weights are changed at all). In that case, if the algorithm starts with all weights equal to 1, the weights will always be integral powers of the factor. It is not hard to construct targets where this leads to a limit on the size of the update factor that can yield a perfect classifier. This issue is discussed further (in the classification context) by Littlestone and Mesterharm (1999), who report on simulation experiments.
3. The existence of bounds depending on a generic pair of dual norms was conjectured by Kivinen and Warmuth (1997), and proved for the classification setting by Grove, Littlestone, and Schuurmans (2001).
4. Here and throughout “ $\cdot$ ” denotes the dot product.
5. sign denotes the usual signum function: $\text{sign}(z) = +1$ if $z \geq 0$ and -1 otherwise.
6. The reader familiar with convexity theory will recognize these properties as simple consequences of the fact that the two functions $P_T(\mathbf{w})$ and $P_{T-1}(\boldsymbol{\theta})$ are Legendre conjugate (e.g., Rockafellar, 1970).
7. This turns out to be a special case of a more general definition of Bregman divergence given by Gordon (1999). See also Azoury and Warmuth (2001).
8. Here and throughout $\ln$ denotes the natural logarithm and e denotes its base.
9. Here we use -1 instead of 0 to match the labels $y_t \in \{-1, +1\}$.
10. A nonuniform prior distribution is actually useful even when the number of models is finite. (See, e.g., Helmbold & Schapire, 1997; Maruoka, Takimoto, & Vovk, 1999) for applications of the Weighted Majority algorithm to learning the best pruning of a decision tree.
11. Since $\mathbf{w}_t^m$ has a finite number of non-zero components, stating Lemma 2 for finite-dimensional vectors is enough.

References

- Angluin, D. (1988). Queries and concept learning. *Machine Learning*, 2:4, 319–342.
- Auer, P., & Warmuth, M. K. (1998). Tracking the best disjunction. *Machine Learning*, 32:2, 127–150.
- Auer, P., & Gentile, C. (2000). Adaptive and self-confident on-line learning algorithms. In *Proc. 13th Annu. Conf. on Comput. Learning Theory* (pp. 107–117). San Mateo, CA: Morgan Kaufmann.
- Azoury, K., & Warmuth, M. K. (2001). Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43, 211–246.
- Barzdin, J. M., & Friedvald, R. V. (1972). On the prediction of general recursive functions. *Soviet Math. Doklady*, 13, 1224–1228.
- Block, H. D. (1962). The perceptron: A model for brain functioning. *Reviews of Modern Physics*, 34, 123–135. Reprinted in Neurocomputing by Anderson and Rosenfeld.
- 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 Physics*, 7, 200–217.
- Bylander, T. (1997). The binary exponentiated gradient algorithm for learning linear functions. In *Proc. 8th Annu. Conf. on Comput. Learning Theory* (pp. 184–192). ACM.

- Cesa-Bianchi, N., Freund, Y., Helmbold, D. P., & Warmuth, M. K. (1996). On-line prediction and conversion strategies. *Machine Learning*, 25, 71–110.
- Cesa-Bianchi, N., Freund, Y., Haussler, D., Helmbold, D. P., Schapire, R. E., & Warmuth, M. K. (1997). How to use expert advice. *Journal of the ACM*, 44:3, 427–485.
- Cesa-Bianchi, N., Helmbold, D. P., & Panizza, S. (1998). On bayes methods for on-line boolean prediction. *Algorithmica*, 22:1, 112–137.
- Cesa-Bianchi, N., Long, P., & Warmuth, M. K. (1996). Worst-case quadratic loss bounds for on-line prediction of linear functions by gradient descent. *IEEE Transactions on Neural Networks*, 7, 604–619.
- Censor, Y., & Lent, A. (1981). An iterative row-action method for interval convex programming. *Journal of Optimization Theory and Applications*, 34:3, 321–353.
- Forster, J., & Warmuth, M. K. (2000). Relative expected instantaneous loss bounds. In *Proc. 13th Annu. Conf. on Comput. Learning Theory* (pp. 90–99). San Mateo, CA: Morgan Kaufmann.
- Freund, Y., & Schapire, R. E. (1999). Large margin classification using the perceptron algorithm. *Machine Learning*, 37:3, 277–296.
- Gentile, C., & Littlestone, N. (1999). The robustness of the p -norm algorithms. In *Proc. 12th Annu. Conf. on Comput. Learning Theory* (pp. 1–11). ACM.
- Gentile, C., & Warmuth, M. K. (1999). Linear Hinge Loss and Average margin. In *Proc. Advances in Neural Information Processing Systems 11* (pp. 225–231). Cambridge, MA: MIT Press.
- Gordon, G. J. (1999). Regret bounds for prediction problems. In *Proc. 12th Annu. Conf. on Comput. Learning Theory* (pp. 29–40). ACM.
- Grove, A. J., Littlestone, N., & Schuurmans, D. (2001). General convergence results for linear discriminant updates. *Machine Learning*, 43:3, 173–210.
- Helmbold, D. P., Kivinen, J., & Warmuth, M. K. (1999). Worst-case loss bounds for sigmoided linear neurons. *IEEE Transactions on Neural Networks*, 10:6, 1291–1304.
- Helmbold, D. P., & Schapire, R. E. (1997). Predicting nearly as well as the best pruning of a decision tree. *Machine Learning*, 27, 51–68.
- Helmbold, D. P., & Warmuth, M. K. (1995). On weak learning. *Journal of Computer and System Sciences*, 50:3, 551–573.
- Herbster, M., & Warmuth, M. K. (1998a). Tracking the best expert. *Machine Learning*, 32:2, 151–178.
- Herbster, M., & Warmuth, M. K. (1998b). Tracking the best regressor. In *Proc. 11th Annu. Conf. on Comput. Learning Theory* (pp. 24–31). ACM.
- Jagota, A., & Warmuth, M. K. (1998). Continuous and discrete time nonlinear gradient descent: Relative loss bounds and convergence. In *Electronic Proceedings of Fifth International Symposium on Artificial Intelligence and Mathematics*. Electronic, <http://rutcor.rutgers.edu/~amai>.
- Kivinen, J., & Warmuth, M. K. (1997). Additive versus exponentiated gradient updates for linear prediction. *Information and Computation*, 132:1, 1–64.
- Kivinen, J., & Warmuth, M. K. (2001). Relative loss bounds for multidimensional regression problems. *Machine Learning*, 45:3, 301–329.
- Kivinen, J., & Warmuth, M. K. (1999). Averaging expert predictions. In *Proc. 4th European Conference on Comput. learning Theory* (pp. 153–167). Lecture Notes in Computer Science, Vol. 1572. Springer.
- Littlestone, N. (1988). Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2, 285–318.
- Littlestone, N. (1989a). From on-line to batch learning. In *Proc. 2nd Annu. Workshop on Comput. Learning Theory* (pp. 269–284). San Mateo, CA: Morgan Kaufmann.
- Littlestone, N. (1989b). Mistake Bounds and Logarithmic Linear-Threshold Learning Algorithms. PhD thesis, Technical Report UCSC-CRL-89-11, University of California Santa Cruz.
- Littlestone, N. (1991). Redundant noisy attributes, attribute errors, and linear threshold learning using Winnow. In *Proc. 4th Annu. Workshop on Comput. Learning Theory* (pp. 147–156). San Mateo, CA: Morgan Kaufmann.
- Littlestone, N., & Mesterharm, C. (1997). An apobayesian relative of Winnow. In *Proc. Advances in Neural Information Processing Systems 9*. Cambridge, MA: MIT Press.
- Littlestone, N., & Mesterharm, C. (1999). A simulation study of Winnow and related learning algorithms. Unpublished manuscript.

- Littlestone, N., & Warmuth, M. K. (1994). The weighted majority algorithm. *Information and Computation*, 108:2, 212–261.
- Maruoka, A., Takimoto, E., & Vovk, V. (1999). Predicting nearly as well as the best pruning of a decision tree through dynamic programming scheme. *Theoretical Computer Science*, to appear.
- Novikov, A. B. J. (1962). On convergence proofs on perceptrons. In *Proc. of the Symposium on the Mathematical Theory of Automata*, vol. XII (pp. 615–622).
- Rockafellar, R. *Convex Analysis*, Princeton University press, 1970.
- Rosenblatt, F. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan Books, Washington, D.C., 1962.
- Vovk, V. (1997). Competitive on-line linear regression. Technical Report CSD-TR-97-13, Department of Computer Science, Royal Holloway, University of London. Preliminary version in *Proc. Advances in Neural Information Processing Systems 10* (pp. 364–370). Cambridge, MA: MIT Press.
- Vovk, V. (1990). Aggregating strategies. In *Proc. 3rd Annu. Workshop on Comput. Learning Theory* (pp. 371–383). San Mateo, CA: Morgan Kaufmann.
- Widrow, B., Hoff, M. E. (1960). Adaptive switching circuits. *1960 IRE WESCON Conv. Record, Part 4* (pp. 96–104).
- Yamanishi, K. (1998). A decision-theoretic extension of stochastic complexity and its applications to learning. *IEEE Information Theory*, 44:4, 1424–1439.

Received January 31, 2001

Revised June 12, 2002

Accepted June 12, 2002

Final manuscript June 12, 2002