The $p$-Norm Generalization of the LMS Algorithm for Adaptive Filtering

Kivinen, Jyrki

2006


http://hdl.handle.net/10138/27989
https://doi.org/10.1109/TSP.2006.872551

Downloaded from Helda, University of Helsinki institutional repository.

This is an electronic reprint of the original article.

This reprint may differ from the original in pagination and typographic detail.

Please cite the original version.
The $p$-Norm Generalization of the LMS Algorithm for Adaptive Filtering

Jyrki Kivinen, Manfred K. Warmuth, and Babak Hassibi

Abstract—Recently much work has been done analyzing online machine learning algorithms in a worst case setting, where no probabilistic assumptions are made about the data. This is analogous to the $H^\infty$ setting used in adaptive linear filtering. Bregman divergences have become a standard tool for analyzing online machine learning algorithms. Using these divergences, we motivate a generalization of the least mean squared (LMS) algorithm. The loss bounds for these so-called $p$-norm algorithms involve other norms than the standard 2-norm. The bounds can be significantly better if a large proportion of the input variables are irrelevant, i.e., if the weight vector we are trying to learn is sparse. We also prove results for nonstationary targets. We only know how to apply kernel methods to the standard LMS algorithm (i.e., $p = 2$). However, even in the general $p$-norm case, we can handle generalized linear models where the output of the system is a linear function combined with a nonlinear transfer function (e.g., the logistic sigmoid).

Index Terms—Adaptive filtering, Bregman divergences, $H^\infty$ optimality, least mean squares, online learning.

I. INTRODUCTION

We focus on the following linear model of adaptive filtering:

$$y_t = \mathbf{u} \cdot \mathbf{x}_t + v_t.$$  \hfill (1)

Here $\mathbf{u}$ is the unknown target, $\mathbf{x}_t$ is a known input, $v_t$ is unknown noise, and $\mathbf{y}_t$ is the known output signal. We are interested in algorithms that maintain a weight vector $\mathbf{w}_t$ based on the past examples $(\mathbf{x}_t, y_t), \tau = 1, \ldots, t$, and, over a sequence of $T$ trials, get as close as possible to the target $\mathbf{u}$. As we shall see, closely related online problems have also been studied in machine learning.

More specifically, at trial $t$ the algorithm receives $\mathbf{x}_t$ and $y_t$ (in order) and has to commit to a weight vector at some point after seeing $\mathbf{x}_t$. We consider three problems depending on whether the algorithm needs to commit to its weight vector before or after seeing $y_t$ and depending on how the loss of the algorithm is measured.

- **A priori filtering:** Here we are interested in predicting the noncorrupted output $\mathbf{u} \cdot \mathbf{x}_t$ before the signal $y_t$ is received. Therefore the algorithm needs to commit to its weight vector $\mathbf{w}_{t-1}$ right before seeing $y_t$ and our loss is the energy of the a priori filtering error $\mathbf{u} \cdot \mathbf{x}_t - \mathbf{w}_{t-1} \cdot \mathbf{x}_t$, i.e.,

$$\sum_{t=1}^{T} (\mathbf{u} \cdot \mathbf{x}_t - \mathbf{w}_{t-1} \cdot \mathbf{x}_t)^2.$$  \hfill (2)

- **A posteriori filtering:** Here we assume that for estimating the noncorrupted output $\mathbf{u} \cdot \mathbf{x}_t$, we also have access to the measurement $y_t$. Thus, the algorithm needs to commit to its weight vector $\mathbf{w}_t$ only after seeing $y_t$ and the loss is the square of the a posteriori error

$$\sum_{t=1}^{T} (\mathbf{u} \cdot \mathbf{x}_t - \mathbf{w}_t \cdot \mathbf{x}_t)^2.$$  \hfill (3)

Note that as in a priori filtering, the algorithm does not know $\mathbf{u}$ when it produces weight vector at trial $t$. It only knows the past instances and outputs.

- **Prediction:** Here we are interested in predicting the next observation $y_{t+1}$ before receiving it. Thus the algorithm needs to commit to its weight vector $\mathbf{w}_{t-1}$ before seeing $y_t$. The prediction error is $y_{t+1} - \mathbf{w}_{t-1} \cdot \mathbf{x}_t$ and the loss is

$$\sum_{t=1}^{T} (y_{t+1} - \mathbf{w}_{t-1} \cdot \mathbf{x}_t)^2.$$  \hfill (4)

The prediction problem of minimizing (4) is also studied in machine learning. Note that in the filtering problems, the term $v_t = y_t - \mathbf{u} \cdot \mathbf{x}_t$ is regarded as a disturbance, so we are interested in estimating the “true output” $\mathbf{u} \cdot \mathbf{x}_t$ of the linear system for the input $\mathbf{x}_t$. In the prediction problem we consider the $y_t$ as the “true outcome” of some event we are interested in predicting. In that case there is no particular value in matching the prediction $\mathbf{u} \cdot \mathbf{x}_t$ at those times when it is inaccurate.

We could also define the notion of a posteriori prediction, i.e.,

$$\sum_{t=1}^{T} (y_{t+1} - \mathbf{w}_{t} \cdot \mathbf{x}_t)^2.$$  \hfill (5)

However, since $y_t$ is known when $\mathbf{w}_t$ is chosen, the loss (5) is trivially minimized by just choosing $\mathbf{w}_t$ such that $\mathbf{w}_t \cdot \mathbf{x}_t = y_t$. 

1053-587X/10.00 © 2006 IEEE
Although there are algorithms that do satisfy $w_t \cdot x_t = y_t$ in some limiting cases, taking this condition as the primary design principle does not seem to add anything. Hence, we do not further consider the loss (5).

In contrast to the loss function used by the prediction problem, the loss functions for the two filtering problems include the target $u$ that is unknown. Because the algorithm cannot even evaluate its own loss, we need to be careful about setting a reasonable performance criterion. We next set the performance criteria we use in this paper, starting with a priori filtering and its connection to recent work in machine learning.

Clearly the quality of output depends on the amount of noise, which can be defined, for example, as $\sum_{t=1}^{T} (y_t - u \cdot x_t)^2$. Additionally, even with no noise, the loss (2) for any given algorithm can be made arbitrarily large by scaling $u$. To have a well-defined choice of $u$, we consider the regularized loss $\sum_{t=1}^{T} (u \cdot x_t - y_t)^2 + (1/\eta) ||u||_2^2$ where $\eta > 0$ is a tradeoff parameter. We now normalize the algorithm's loss (2) with respect to the regularized loss. Since we wish to avoid assumptions about $u$, we consider the worst case choice, leading us to the quantity

$$\max_u \frac{\sum_{t=1}^{T} (u \cdot x_t - y_t)^2}{\sum_{t=1}^{T} (u \cdot x_t - y_t)^2 + \frac{1}{\eta} ||u||_2^2}. \tag{6}$$

Given the data $(x_t, y_t)$ and an algorithm for producing $u_t$, the quantity (6) is always well defined. In control theory, (6) is seen as a maximum energy gain and called the $H^\infty$ norm. (For the above, and as done throughout this paper, we assumed $y_t = 0$; if $u_t \neq 0$, then $||u||_2^2$ must be replaced by $||u - u_t||_2^2$.)

To get a reference point, consider the least mean squares (LMS) algorithm (2) (also known as the Widrow–Hoff algorithm), defined by the update rule

$$u_t = u_{t-1} - \eta (u_{t-1} \cdot x_t - y_t) x_t \tag{7}$$

where $\eta > 0$ is now a parameter of the algorithm and called the learning rate. According to the basic result for a priori filtering [3], if $\eta \leq 1/\max ||x_t||_2^2$, then the LMS algorithm satisfies

$$\max_u \frac{\sum_{t=1}^{T} (u \cdot x_t - y_t)^2}{\sum_{t=1}^{T} (u \cdot x_t - y_t)^2 + \frac{1}{\eta} ||u||_2^2} \leq 1. \tag{8}$$

In other words, LMS has $H^\infty$ norm at most 1. (Notice that the learning rate parameter of the algorithm becomes the tradeoff parameter for the regularized loss.) Further, no algorithm can have $H^\infty$ norm less than 1. Therefore, we say that LMS is $H^\infty$ optimal.

To compare prediction with filtering, we write (6) as

$$\sum_{t=1}^{T} (u_t \cdot x_t - u_{t-1} \cdot x_t)^2 \leq \sum_{t=1}^{T} (u \cdot x_t - y_t)^2 + \frac{1}{\alpha} X_2^2 ||u||_2^2 \tag{10}$$

where $X_2$ and $\eta$ are as above and $0 < \alpha \leq 1$. We see that the bounds are similar in form, except for the factor $1/(1-\alpha)$ in (9).

The factor $1/(1-\alpha)$ in (9) is a source of many difficulties in machine learning, where the goal is to tune the learning rate so as to obtain the smallest possible bound. However, the filtering bound (10) is optimized at $\alpha = 1$. Thus we omit the $\alpha$ parameter from the filtering bounds when the norm of instances is bounded.

Motivated by the similarity between (9) and (10), we are going to take machine learning techniques that have recently been used to generalize the LMS algorithm and apply them in the filtering setting. This leads to generalizations of (10) and new interpretations of the filtering algorithm. Techniques we are interested in include:

1) motivating algorithms in terms of minimization problems based on Bregman divergences [5], [6];
2) replacing the 2-norms in the bounds by other norms [5], [7], [8];
3) allowing for nonstationary targets [9] and nonlinear predictors [10].

Before going on with the above program, let us have a brief look at the a posteriori model. The $H^\infty$ norm for a posteriori filtering is

$$\max_u \frac{\sum_{t=1}^{T} (u \cdot x_t - u_t \cdot x_t)^2}{\sum_{t=1}^{T} (u \cdot x_t - y_t)^2 + \frac{1}{\eta} ||u||_2^2}. \tag{11}$$

Notice that since $y_t$ is available when choosing $u_t$, we can trivially obtain $H^\infty$ norm at most 1 by any choice that satisfies $u_t \cdot x_t = y_t$. One particular way of doing this would be to let the learning rate go to infinity in the normalized LMS algorithm [3]. However, there are other criteria that are minimized by using a finite learning rate, while still retaining the $H^\infty$ norm at most 1. For example, this is the case if the data points are generated by the model (1) with the noise variables $v_t$ independent and Gaussian [3, Theorem 9]. Thus, while requiring the $H^\infty$ norm to be at most 1 is a good robustness guarantee, in the a posteriori case such a worst case measure is not by itself a sufficient criterion for choosing a good algorithm. In the following we will state all our bounds both for a priori and a posteriori filtering, but they must be read with this caveat in mind.

Our $H^\infty$-based performance criteria do not directly address convergence. If the data are generated by the model (1) with the noise variables $v_t$ independent and Gaussian, then one could hope that the weights $u_t$ would converge toward the target $u$. However, if we do not wish to make such assumptions about noise, the issue becomes less clear. An algorithm geared toward fast convergence under zero-mean independent noise may fail badly if, say, the early data points have large amounts of biased and correlated noise. We aim for results that are not sensitive to probabilistic assumptions and develop bounds like (6) and (10), which hold for every sequence of examples. Such worst
case bounds are rather stringent. If the examples are independent identically distributed (i.i.d.), an averaging technique can be used to convert worst case loss bounds to bounds on the expected loss (see, e.g., [5, Section 8]) or bounds on the probability of high loss [11]. Clearly the choice of algorithm should depend on the assumptions. In particular, even with independent noise, updates like (7) with fixed learning rate do not typically lead to convergence but remain oscillating around the optimal weight setting.

In Section III, we introduce Bregman divergences and show how a Bregman divergence can be used to derive two subtly different updates: the implicit and explicit update. When the squared Euclidean distance is used as the Bregman divergence, these updates give the standard LMS and normalized LMS algorithm [3], respectively. In Section IV, we give filtering loss bounds for the explicit and implicit updates in the case of Bregman divergences based on squared norms [7]. These bounds generalize the results of Hassibi et al. [3] about the $H^\infty$ optimality of LMS and normalized LMS for the a priori and a posteriori filtering problems. The generalization replaces the product $\|x\|_p \|u\|_q$ by the bound produced by another norm $\|x\|_p \|u\|_q$, where $p$ and $q$ are such that $1/p + 1/q = 1$ and $2 \leq p < \infty$. The new bounds are significantly stronger when the target $u$ is sparse, i.e., has few nonzero components. In Section V, we generalize the $q$-norm based algorithms to allow for nonstationary targets $u_t$. The loss bounds in the nonstationary case include an extra term that depends on the total distance $u_t$ travels during the whole sequence, as measured by the $q$-norm. Again there are no distribution assumptions about this movement. Section VI gives bounds for generalized linear regression where the linear predictor is fed through a nonlinear transfer function (such as the logistic sigmoid). Some simulations are reported in Section VII, and our conclusions presented in Section VIII.

Some preliminary results of this paper were presented at the 13th IFAC Symposium on System Identification [1]. This paper includes some additional algorithms and new simulation results, as well as full proofs of the theoretical results.

II. THE LMS BOUND

As an introduction to our methods, we rederive the basic result of [3]. Later we will see how the algorithm and proof generalize from the Euclidean to other $p$-norms.

**Theorem 1** [3]: Assume that $\|x_t\|_2 \leq X_2$ for all $t$, and choose $\eta = 1/X_2$. Then the LMS algorithm (7) satisfies

$$\sum_{t=1}^{T} (u_t \cdot x_t - w_{t-1} \cdot x_t)^2 \leq \sum_{t=1}^{T} (u_t \cdot x_t - y_t)^2 + X_2^2 \|u\|_2^2$$

for any $u \in \mathbb{R}^n$.

**Proof:** Following [4], we analyze the progress $d_t = (1/2) \|u - w_{t-1}\|_2^2 - (1/2) \|u - w_{t-1}\|_2^2$ made at update $t$ toward the comparison vector $u$. Direct calculation gives us

$$d_t = \eta (y_t - w_{t-1} \cdot x_t) (u_t \cdot x_t - w_{t-1} \cdot x_t) - \eta^2/2 (y_t - w_{t-1} \cdot x_t)^2 \|x_t\|_2^2$$

By estimating $\|x_t\|_2 \leq X_2$ and rearranging terms, we get

$$d_t \geq \frac{\eta}{2} s_{t}^2 - \frac{\eta}{2} r_{t}^2 + \frac{\eta}{2} (s_{t} - r_{t})^2 (1 - \eta X_2^2)$$

where $s_t = u \cdot x_t - w_{t-1} \cdot x_t$ and $r_t = u \cdot x_t - y_t$. Since $\eta X_2^2 = 1$ and $u_0 = 0$, we can apply $\|u - w_0\|_2 = \|u\|_2$ and $\|u - w_{T+1}\|_2 \geq 0$ to get

$$\frac{1}{2} \|u\|_2^2 \geq \frac{1}{2} \|u - w_0\|_2^2 - \frac{1}{2} \|u - w_{T+1}\|_2^2$$

$$= \sum_{t=1}^{T} d_t$$

$$\geq \frac{1}{2 X_2^2} \left( \sum_{t=1}^{T} s_{t}^2 - \sum_{t=1}^{T} r_{t}^2 \right)$$

from which the claim follows.

III. DERIVATION OF ALGORITHMS

In this section we give the basic definitions of Bregman divergences and explain their use in deriving generalizations of the LMS algorithm. (See [12] and references therein for more background on these divergences.) Later the same Bregman divergences will be used to prove bounds for these new algorithms. Note that the bound for the LMS algorithm involves the 2-norms of the inputs $x$ and target $u$. The bounds for the new algorithm will depend on norms $\|x\|_p$ and $\|u\|_q$ where in general $p, q \neq 2$.

Assume that $F$ is a strictly convex twice differentiable function from a subset of $\mathbb{R}^n$ to $\mathbb{R}$. Denote its gradient by $f = \nabla F$; notice that $f$ is one-to-one. The Bregman divergence $\Delta_F(u, w)$ [13] is defined for $u, w \in \mathbb{R}^n$ as the error in approximating $F(u)$ by its first order Taylor polynomial around $w$. More formally

$$\Delta_F(u, w) = F(u) - F(w) - (u - w) \cdot f(w).$$

The Bregman divergence $\Delta_F(u, w)$ is always nonnegative, and zero only for $u = w$. It is (strictly) convex in $u$ but might not be convex in $w$. Usually, $\Delta_F$ is not symmetric.

**Example 1:** For $q > 1$, define $F(w) = (1/2) \|w\|_q^2$, where $\|\cdot\|_q$ denotes the $q$-norm defined as $\|w\|_q = (\sum |w_i|^q)^{1/q}$. We denote the corresponding Bregman divergence by $\Delta_q$. Thus

$$\Delta_q(u, w) = \frac{1}{2} \|w\|_q^2 - \frac{1}{2} \|w\|_q^2 - (u - w) \cdot f(w)$$

where the gradient is given by

$$f_i(w) = \frac{\text{sign}(w_i) |w_i|^{q-1}}{|w_i|^{q-2}}.$$
is the unnormalized relative entropy. (When \( \sum a_i = \sum w_i = 1 \), this gives the standard relative entropy.) The gradient is given by

\[ f_t(w) = \ln w_i. \]

The following generalization of the Pythagorean theorem follows directly from the definition of a Bregman divergence:

\[ \Delta_F(u, u') = \Delta_F(u, w) + \Delta_F(w, u') + (w - u) : (f'(u') - f'(w)). \quad (11) \]

Since the dot product \((w - u) : (f'(u') - f'(w))\) can be positive, this shows in particular that \(\Delta_F\) does not satisfy the triangle inequality. We recover the standard Pythagorean theorem when the divergence is the squared Euclidean distance (i.e., \(f\) is identity) and the dot product is zero (i.e., \((u' - w)\) and \(w - u\) are orthogonal).

We now use a Bregman divergence \(\Delta_F\) as a regularizer for deriving an update rule. This framework for motivating updates was introduced in [5] in the prediction setting. In the following, we are mainly interested in Bregman divergences based on the squared \(q\)-norm. They were introduced in [7] to analyze algorithms for learning linear threshold functions.

Suppose an example \((x_i, y_i)\) has been observed and we wish to update our hypothesis \(w_{t-1}\) based on this example. We wish to decrease the squared loss \((y_i - w : x_i)^2\) (other convex loss functions can also be considered; see Section VI). However, we should not make big changes based on just a single example. Thus, we define

\[ C_t(w) = \Delta_F(w, w_{t-1}) + \frac{1}{2} \eta (y_i - w : x_i)^2 \]

where \(\eta > 0\) is a tradeoff parameter, and tentatively set \(w_t = \arg \min_w C_t(w)\). Since \(C_t\) is convex, we can minimize by setting \(\nabla C_t(w_t) = 0\). By substituting the definition of \(\Delta_F\), this becomes

\[ w_t = f^{-1}(f(w_{t-1}) - \eta (w_{t-1} : x_i - y_i)x_i). \quad (12) \]

Since \(w_t\) appears on both sides of (12), we call the update rule defined by this equality the implicit update for divergence \(\Delta_F\). Notice that (12) can be solved numerically by a line search since \(w = f^{-1}(f(w_{t-1} + \alpha x))\) for some scalar \(\alpha\), and the inverse \(f^{-1}\) is easy to compute in the cases we consider. Also in the special case of 2-norm (\(\Delta_F = \Delta_2\)), with \(f\) the identity function, we can solve (12) in closed form to get

\[ w_t = w_{t-1} - \frac{\eta}{1 + \eta |x_i|^2} (w_{t-1} : x_i - y_i)x_i. \quad (13) \]

This is the algorithm called normalized LMS in [3].

Instead of solving (12) numerically, we often find it sufficient to notice that for reasonable values of \(\eta\), the values \(w_{t-1} : x_i\) and \(w_{t-1} : x_t\) should be fairly close to each other. Thus, we may approximate the solution of (12) by

\[ w_t = f^{-1}(f(w_{t-1}) - \eta (w_{t-1} : x_i - y_i)x_i). \quad (14) \]

We call this the explicit update for divergence \(\Delta_F\). The special case \(\Delta_F = \Delta_2\) gives the usual LMS algorithm.

Note that the explicit update uses the gradient of the square loss evaluated at the old weight vector \(w_{t-1}\), whereas the implicit update is based on the gradient at the updated parameter vector \(w_t\). For a discussion of taking the old gradient versus the future gradient in for the prediction problem, and a derivation of the implicit LMS algorithm, see [5]. In [14], an implicit update was derived as an alternate to the TD(\(\lambda\)) algorithm. In this case the implicit definition was crucial for producing an improved algorithm.

IV. Bounds in Terms of Different Norms

Our interest in considering the generalization of LMS to the \(p\)-norm based algorithms comes from the fact that for these algorithms, the term \(\|x\|_2\|u\|_2\) in the LMS bound is replaced by another product of dual norms \(\|x\|_p\|u\|_q\) (i.e., \(1/p + 1/q = 1\)). We discuss the implications of this after giving the main result, which is a direct generalization of Theorem 1.

We consider the explicit (14) and implicit (12) updates for the divergence \(\Delta_q(u, w)\) given in Example 1. The special case \(q = 2\) gives the classic LMS and Theorem 1. For the updates, we need the gradient \(f\), which was given in Example 1, and also its inverse \(f^{-1}\), which is easily seen to be

\[ f_i^{-1} = \frac{\text{sign}(\theta_i)}{|\theta_i|^{p-1}} \]

where \(1/p + 1/q = 1\).

We assume the relationship \(1/p + 1/q = 1\) throughout this paper. It means that we can apply Hölder’s inequality \(\|u : x\| \leq \|u\|_p\|x\|_q\). As a further convention, we assume \(q \leq p\), so \(1 < q \leq 2 \leq p < \infty\). The important special case \(p = q = 2\) gives \(\Delta_2(u, w) = (1/2)\|u - w\|_2^2\), with \(f\) the identity function.

We use the following inequality for proving bounds for the updates:

\[ \Delta_q(u, f^{-1}(f(w) + x)) \leq \frac{p-1}{2} \|x\|_p^2. \quad (15) \]

This inequality is implied by derivations given in [7] and was stated explicitly in [8, Lemma 2]. For completeness, we give the proof in Appendix 1.

Theorem 2: Fix \(p\) and \(q\) such that \(1/p + 1/q = 1\) and \(2 \leq p < \infty\). Assume that \(\|x\|_p \leq \bar{X}_p\) for all \(x\). Then the explicit update (14) for \(\Delta_q\) with learning rate \(\eta = 1/((p - 1)\bar{X}_p^2)\) satisfies

\[ \sum_{i=1}^T (u_i : x_i - y_i x_i)^2 \leq \sum_{i=1}^T (u_i : x_i - y_i)^2 + (p - 1)\bar{X}_p^2 \|u\|_p^2 \]

for any \(u \in \mathbb{R}^n\).

Proof: Following [5], we analyze the progress

\[ d_t = \Delta_q(u, w_{t-1}) - \Delta_q(u, w_t) \]

made at update \(t\) toward the comparison vector \(u\). By substituting (14) into (11) and then using (15), we get

\[ d_t \geq \eta (y_i - w_{t-1} : x_t) w_{t-1} : x_t + (u_t : x_t) - \eta (y_i - w_{t-1} : x_t) u_t : x_{t-1} - \|x\|_p^2 \]

\[ \geq \frac{\eta}{2} (y_i - w_{t-1} : x_t)^2 X_p^2. \]

By rearranging terms, we can write this as

\[ d_t \geq \frac{\eta}{2} s_i^2 - \frac{\eta}{2} r_i^2 + \frac{\eta}{2} (s_t - r_t)^2 (1 - \eta (p - 1)X_p^2) \]
where \( s_t = \mathbf{u} \cdot \mathbf{x}_t - w_{t-1} \cdot \mathbf{x}_t \) and \( r_t = \mathbf{u} \cdot \mathbf{x}_t - y_t \). Since \( \eta (p-1) X_p^2 = 1 \) and \( \mathbf{u}_0 = 0 \), we can apply \( \Delta_q(\mathbf{u}, \mathbf{u}_0) = (1/2)\|\mathbf{u}\|_q^2 \) and \( \Delta_q(\mathbf{u}, \mathbf{w}_T) \geq 0 \) to get

\[
\frac{1}{2} \|\mathbf{u}\|_q^2 \geq \Delta_q(\mathbf{u}, \mathbf{w}_0) - \Delta_q(\mathbf{u}, \mathbf{w}_{T+1})
\]

\[
= \sum_{t=1}^{T} d_t
\]

\[
\geq \frac{1}{2(p-1)} X_p^2 \left( \sum_{t=1}^{T} s_t^2 - \sum_{t=1}^{T} r_t^2 \right)
\]

from which the claim follows.

The main intuitive implication of Theorem 2 (and later Theorem 3, which will deal with the implicit update) is that the bound favors large \( p \) when the target \( \mathbf{u} \) is sparse. To make this more precise, we compare the bound for \( p = 2 \) (i.e., classic LMS) against \( p = 2 \ln n \) (i.e., fairly large \( p \)). Gentile and Littlestone [8, Corollary 7] have shown that for the particular choice \( p = 2 \ln n \), we have

\[
(p-1)\|\mathbf{x}\|_p^2 \|\mathbf{u}\|_q^2 \leq (2e \ln n)\|\mathbf{x}\|_\infty^2 \|\mathbf{u}\|_q^2
\]  

(16)

(where \( \|\mathbf{x}\|_\infty = \max_i |x_i| \)). Thus, we compare the bound

\[
\|\mathbf{x}\|_\infty^2 \|\mathbf{u}\|_q^2
\]

for LMS with the bound \((2e \ln n)\|\mathbf{x}\|_\infty^2 \|\mathbf{u}\|_q^2\) (for large \( p \)).

Since the \( p \)-norm is decreasing in \( p \), we have \( \|\mathbf{u}\|_2 \leq \|\mathbf{u}\|_1 \) and \( \|\mathbf{x}\|_2 \geq \|\mathbf{x}\|_\infty \), with equality if the vector has only one nonzero component. Hence, the dependence on \( \mathbf{u} \) favors \( p = 2 \), but the advantage gets smaller if \( \mathbf{u} \) is very sparse. Similarly, the dependence on \( \mathbf{x} \) favors large \( p \), but the advantage gets smaller if \( \mathbf{x} \) is very sparse.

To get a concrete picture of the tradeoff, let us consider two extreme cases. In the first case, we choose \( \mathbf{u} \in \{-1,1\}^n \) and \( \mathbf{x} \in \{-1,0,1\}^n \) such that exactly one component \( x_i \) is nonzero. Then \( \|\mathbf{u}\|_2^2 = n, \|\mathbf{u}\|_1^2 = n^2 \), and \( \|\mathbf{x}\|_2 = \|\mathbf{x}\|_\infty = 1 \). The LMS bound becomes simply \( n \), while the large \( p \) bound becomes \( 2e n^2 \ln n \). Hence, the LMS bound is clearly better for large \( n \).

In the second case, choose \( \mathbf{u} \in \{-1,0,1\}^n \) such that exactly one component \( u_j \) is nonzero, and choose \( \mathbf{x} = \{-1,1\}^n \). Then \( \|\mathbf{u}\|_2^2 = \|\mathbf{u}\|_1 = 1, \|\mathbf{u}\|_\infty^2 = n, \) and \( \|\mathbf{x}\|_\infty = 1 \). The LMS bound is \( n \) as in the first case, but the large \( p \) bound drops to \( 2e n \). Notice that the dependence on \( n \) in this last bound is only logarithmic, so for large \( p \) the difference to LMS can be quite large.

In the context of prediction, much attention has been paid to multiplicative algorithms such as Winnow [15] and EG [5], which have bounds similar to the \( p \)-norm algorithms for \( p = \Theta(\log n) \). In addition to upper bounds and simulations [5], there are also some lower bounds [16] showing that in certain situations LMS-style algorithms cannot perform as well as multiplicative ones. The multiplicative EG algorithm can be seen as applying the update (14) with \( f_t(\mathbf{w}) = \ln u_t \) (with a further normalization step). The analysis of EG can also be lifted to the filtering setting, resulting in the bound

\[
\sum_{t=1}^{T} (\mathbf{u} \cdot \mathbf{x}_t - w_{t-1} \cdot \mathbf{x}_t)^2 \leq \sum_{t=1}^{T} (\mathbf{u} \cdot \mathbf{x}_t - y_t)^2 + \ln(2n)\|\mathbf{x}\|_\infty^2 \|\mathbf{u}\|_q^2
\]

for a scaled explicit version. See Appendix II for details and notice the improved constant of \( \ln 2n \) over \( 2e \ln n \) appearing in (16). Multiplicative algorithms are closely related to \( L_2 \) regularization, which can be seen as a form of feature selection [17].

We now consider the a posteriori case. The following theorem generalizes the result about normalized LMS in [3]. However, our result has an additional restriction on the learning rate, which we believe to be an artefact of the proof technique. We shall discuss this after giving the theorem and its proof.

**Theorem 3:** Fix \( p \) and \( q \) such that \( 1/p + 1/q = 1 \) and \( 2 \leq p < \infty \). Assume that \( \|\mathbf{x}\|_p \leq X_p \) for all \( t \). Then the implicit update for \( \Delta_q \) with learning rate \( \eta = 1/(p-1) X_p^2 \) satisfies

\[
\sum_{t=1}^{T} (\mathbf{u} \cdot \mathbf{x}_t - w_{t-1} \cdot \mathbf{x}_t)^2 \leq \sum_{t=1}^{T} (\mathbf{u} \cdot \mathbf{x}_t - y_t)^2 + (p-1) X_p^2 \|\mathbf{u}\|_q^2
\]

**Proof:** Again let \( d_t = \Delta_q(\mathbf{u}, \mathbf{w}_{t-1}) - \Delta_q(\mathbf{u}, \mathbf{w}_t) \). By substituting (12) into (11) and applying (15), we get

\[
d_t = \eta (y_t - \mathbf{u}_t \cdot \mathbf{x}_t) ((\mathbf{u}_t - \mathbf{w}_{t-1} \cdot \mathbf{x}_t) - \eta (y_t - \mathbf{u}_t \cdot \mathbf{x}_t))
\]

\[
\times ((y_t - \mathbf{u}_t \cdot \mathbf{x}_t) - \frac{p-1}{2} \eta^2 (y_t - \mathbf{w}_t \cdot \mathbf{x}_t)^2 X_p^2.
\]

Since \( \mathbf{u}_t \) minimizes \( C_t \), it is easy to show that \( \mathbf{w}_{t-1} \cdot \mathbf{x}_t \leq \mathbf{w}_t \cdot \mathbf{x}_t \leq y_t \) or \( y_t \leq \mathbf{w}_t \cdot \mathbf{x}_t \leq \mathbf{w}_{t-1} \cdot \mathbf{x}_t \); that is, the update moves \( \mathbf{w}_t \cdot \mathbf{x}_t \) to the right direction but not too far. This implies

\[
(y_t - \mathbf{w}_t \cdot \mathbf{x}_t)(y_t - \mathbf{w}_{t-1} \cdot \mathbf{x}_t) \geq (y_t - \mathbf{w}_t \cdot \mathbf{x}_t)^2
\]

so we get

\[
d_t \geq \eta (y_t - \mathbf{u}_t \cdot \mathbf{x}_t)^2 - \eta (y_t - \mathbf{w}_t \cdot \mathbf{x}_t)(y_t - \mathbf{u}_t \cdot \mathbf{x}_t)
\]

\[
- \frac{p-1}{2} \eta^2 (y_t - \mathbf{w}_t \cdot \mathbf{x}_t)^2 X_p^2.
\]

We can rewrite this as

\[
d_t \geq \eta (s_t - r_t)^2 + \eta (s_t - r_t) r_t - \frac{p-1}{2} \eta^2 X_p^2 (s_t - r_t)^2
\]

where \( s_t = \mathbf{u} \cdot \mathbf{x}_t - \mathbf{w}_t \cdot \mathbf{x}_t \), and \( r_t = \mathbf{u}_t \cdot \mathbf{x}_t - y_t \). By rearranging terms, this becomes

\[
d_t \geq \eta s_t^2 - \frac{p-1}{2} \eta^2 X_p^2 s_t^2 + \eta (s_t - r_t)^2 (1 - \eta (p-1) X_p^2).
\]

The rest follows as in the proof of Theorem 2.

**Our proof actually implies**

\[
\sum_{t=1}^{T} (\mathbf{u} \cdot \mathbf{x}_t - \mathbf{w}_t \cdot \mathbf{x}_t)^2 \leq \sum_{t=1}^{T} (\mathbf{u} \cdot \mathbf{x}_t - y_t)^2 + \frac{1}{\eta} \|\mathbf{u}\|_q^2
\]

(18)
for any learning rate $0 < \eta \leq 1/(p - 1)X_p^2$. For the case $p = 2$, Hassibi et al. [3] actually show (18) for any $\eta > 0$. Notice that the estimate (17) in our proof can equivalently be written as $(\mathbf{u}_{t-1} \cdot \mathbf{x}_t - \mathbf{u}_t \cdot \mathbf{x}_t)/(\mathbf{u}_t \cdot \mathbf{x}_t - y_t) \geq 0$. This holds as equality for $\eta = 0$, but becomes very loose as $\eta$ approaches infinity (so $\mathbf{u}_t \cdot \mathbf{x}_t - y_t$ approaches zero). In the case $p = 2$, we can use the closed form (13) of the normalized LMS algorithm to obtain $(\mathbf{u}_{t-1} \cdot \mathbf{x}_t - \mathbf{u}_t \cdot \mathbf{x}_t)/(\mathbf{u}_t \cdot \mathbf{x}_t - y_t) = \eta \|\mathbf{x}_t\|^2_2$. Using this tighter estimate allows the proof to go through for arbitrary $\eta > 0$. Unfortunately, we have not been able to obtain a similar bound for the case $p > 2$, with nonlinear $f$ in the update (12).

As discussed in [5], whenever a learning rate $\eta$ needs to be tuned, then the tuned choice should be of the correct “type.” As we shall see, this is indeed the case in the above two theorems. We denote the type of the weight vectors as $[\mathbf{w}]$ and the type of the instances as $[\mathbf{x}]$. The type of the outputs must then be $[\mathbf{w} \cdot \mathbf{x}] = [\mathbf{w}] [\mathbf{x}]$. It is easy to check that the transformations $f$ and $f^{-1}$ for $\Delta_q$ do not change the type of a weight vector. So now the type of $\eta$ in the implicit and explicit update for $\Delta_q$ must be $[\eta]^{-2}$ and the tunings prescribed in the theorems indeed choose an $\eta$ of this type. Throughout this paper, our tunings of $\eta$ always fix the type of $\eta$ for all the updates discussed.

V. Nonstationary Targets

Following [9], we now consider a variant of the algorithm that keeps the $q$-norm of the weight vector bounded by $U_q$, where $U_q > 0$ is a parameter to the algorithm. We call this two-step update the bounded explicit update for $\Delta_F$.

- **Explicit update step:** Let
  $$ \mathbf{w}_t = f^{-1}(f(\mathbf{w}_{t-1}) - \eta f(\mathbf{w}_{t-1} \cdot \mathbf{x}_t - y_t) \mathbf{x}_t). $$

- **Out-of-bound update step:** If $\|\mathbf{w}_t\|_q > U_q$, then $\mathbf{w}_t = U_q \mathbf{w}_t/\|\mathbf{w}_t\|_q$; otherwise $\mathbf{w}_t = \mathbf{w}_t'$.

Thus the update tries to increase the $q$-norm of its weight vector above $U_q$, then we scale it back.

We now let the target $\mathbf{w}_t$ vary with time (nonstationary model):

$$ y_t = \mathbf{w}_t \cdot \mathbf{x}_t + \eta_t. \quad (19) $$

As previously, our bound will include a penalty for the (maximum) norm of $\mathbf{w}_t$. Additionally, there is now also a penalty for the total distance the target moves during the process.

**Theorem 4:** Fix $p$ and $q$ such that $1/p + 1/q = 1$ and $2 \leq p < \infty$. Assume $\|\mathbf{x}_t\|_p \leq X_p$ and $\|\mathbf{w}_t\|_q \leq U_q$ for all $t$. Then the bounded explicit update for $\Delta_q$ with learning rate $\eta = 1/((p - 1)X_p^2)$ and parameter $U_q$ satisfies

$$ \sum_{t=1}^T \sum_{t=1}^T (\mathbf{u}_t \cdot \mathbf{x}_t - y_t)^2 \leq \frac{T}{p} \sum_{t=1}^T (\mathbf{u}_t \cdot \mathbf{x}_t - y_t)^2 + (p - 1)X_p^2 U_q^2 $$

$$ + 2(p - 1)X_p^2 U_q \sum_{t=1}^T \|\mathbf{w}_{t+1} - \mathbf{w}_t\|_q^2. $$

**Proof:** We apply the proof technique introduced in the prediction setting in [9]. We define the progress at trial $t$ as the sum of three parts $d_t^1 = d_t^3 + d_t^2 + d_t^3$, where

$$ d_t^1 = \Delta_q(\mathbf{u}_t, \mathbf{w}_{t-1}) - \Delta_q(\mathbf{u}_t, \mathbf{w}_t) $$

$$ d_t^2 = \Delta_q(\mathbf{u}_t, \mathbf{w}_t') - \Delta_q(\mathbf{u}_t, \mathbf{w}_t) $$

$$ d_t^3 = \Delta_q(\mathbf{u}_t, \mathbf{w}_t') - \Delta_q(\mathbf{u}_{t+1}, \mathbf{w}_t). $$

Then $d_t = \Delta_q(\mathbf{u}_t, \mathbf{w}_{t-1}) - \Delta_q(\mathbf{u}_{t+1}, \mathbf{w}_t)$. (For notational convenience we define $\mathbf{w}_T = \mathbf{w}_T'$ for the last time step.)

For $\eta \leq 1/((p - 1)X_p^2)$, the proof of Theorem 2 gives directly

$$ d_t^1 \geq \frac{\eta}{2} s_t^2 - \frac{\eta}{2} r_t^2 $$

(20)

where $s_t = \mathbf{u}_t \cdot \mathbf{x}_t - \mathbf{w}_{t-1} \cdot \mathbf{x}_t$ and $r_t = \mathbf{u}_t \cdot \mathbf{x}_t - y_t$.

For estimating $d_t^2$, first note that the out-of-bound step can be expressed as

$$ \mathbf{w}_t = \arg \min_{\|\mathbf{w}\| \leq U_q} \Delta_q(\mathbf{w}, \mathbf{w}_t'). $$

In other words, $\mathbf{w}_t$ is the projection of $\mathbf{w}_t'$ into the closed convex set $B = \{\mathbf{w} : \|\mathbf{w}\|_q \leq U_q\}$ with respect to $\Delta_q$. Well-known properties of such projections [9], [13] imply that for any $\mathbf{u} \in B$, we have $\Delta_q(\mathbf{u}, \mathbf{w}_t') \leq \Delta_q(\mathbf{u}, \mathbf{w}_t')$ and thus $d_t^2 \geq 0$.

From the definition of $\Delta_q$, we get

$$ d_t^3 = \frac{1}{2} \|\mathbf{w}_t\|_q^2 - \frac{1}{2} \|\mathbf{w}_{t+1}\|_q^2 \geq \|\mathbf{w}_{t+1} - \mathbf{w}_t\|_q \|\mathbf{f}(\mathbf{w}_t)\|_p. $$

By Hölder’s inequality, $\|\mathbf{w}_{t+1} - \mathbf{w}_t\|_q \|\mathbf{f}(\mathbf{w}_t)\|_p \leq \|\mathbf{w}_{t+1} - \mathbf{w}_t\|_q \|\mathbf{f}(\mathbf{w}_t)\|_q$. Since $\|\mathbf{f}(\mathbf{w}_t)\|_p = \|\mathbf{w}_t\|_q \leq U_q$, we get

$$ d_t^3 \geq \frac{1}{2} \|\mathbf{w}_t\|_q^2 - \frac{1}{2} \|\mathbf{w}_{t+1}\|_q^2 \geq 0. $$

By summing over $t = 1, \ldots, T$ and substituting the value of $\eta$, we obtain

$$ \Delta_q(\mathbf{u}_0, \mathbf{w}_0) - \Delta_q(\mathbf{u}_{T+1}, \mathbf{w}_t) $$

$$ = \sum_{t=1}^T d_t $$

$$ \geq \sum_{t=1}^T \frac{1}{2} \|\mathbf{w}_t\|_q^2 - \frac{1}{2} \|\mathbf{w}_{t+1}\|_q^2 \geq 0. $$

For $\mathbf{w}_0 = \mathbf{0}$, we have $\Delta_q(\mathbf{u}_0, \mathbf{w}_0) = \|\mathbf{w}_0\|_q^2/2$. Estimating $\Delta_q(\mathbf{u}_{T+1}, \mathbf{w}_t) \geq 0$ and $\|\mathbf{w}_{T+1}\|_q \leq U_q$ gives the claim.

In the special case $\mathbf{w}_t = \mathbf{u}_{t+1}$ for all $t$, the result becomes Theorem 2 with the exception that the norm bound $U_q$ must be fixed in advance.

The same technique can be applied to the a posteriori problem. Given $U_q > 0$, we define the bounded implicit update for $\Delta_F$ with the following two-step update:

- **Implicit update step:** Let $\mathbf{w}_t$ be such that

$$ \mathbf{w}_t = f^{-1}(f(\mathbf{w}_{t-1}) - \eta f(\mathbf{w}_{t-1} \cdot \mathbf{x}_t - y_t) \mathbf{x}_t). $$

- **Out-of-bound update step:** If $\|\mathbf{w}_t\|_q > U_q$, then $\mathbf{w}_t = U_q \mathbf{w}_t/\|\mathbf{w}_t\|_q$; otherwise $\mathbf{w}_t = \mathbf{w}_t'$.

Thus, we swapped the notation from the explicit update and use $\mathbf{w}_t'$ for the bounded and $\mathbf{w}_t$ for the unbounded weight. Basically we now want to predict with the unbounded weights. The bound is as expected.

**Theorem 5:** Fix $p$ and $q$ such that $1/p + 1/q = 1$ and $2 \leq p < \infty$. Assume $\|\mathbf{x}_t\|_p \leq X_p$ and $\|\mathbf{w}_t\|_q \leq U_q$ for all $t$. Then the
bounded implicit update for $\Delta_q$ with learning rate $\eta = 1/((p-1)X^2_{p,t})$ and parameter $U_q$ satisfies

$$
T \sum_{t=1}^{T} (u_t \cdot x_t - w_{t-1} \cdot x_t)^2 \leq T \sum_{t=1}^{T} (u_t \cdot x_t - y_t)^2
+ (p-1)X^2_{p,t}U^2_q + 2(p-1)X^2_{p,t}U_q \sum_{t=1}^{T-1} ||u_{t+1} - u_t||_q.
$$

**Proof:** We mimic the proof of Theorem 4. This time we set

$$
d^2_t = \Delta_q(u_t, w_{t-1}) - \Delta_q(u_t, w_t),
$$

$$
d^2_t = \Delta_q(u_t, w_t) - \Delta_q(u_t, w_t')
$$

and

$$
d^3_t = \Delta_q(u_t, w_t') - \Delta_q(u_{t+1}, w_t').
$$

For $\eta \leq 1/((p-1)X^2_{p,t})$, the proof of Theorem 3 gives

$$
d^1_t \geq \eta \frac{\eta}{2} \frac{\eta}{2} R^2
$$

where $s_t = u_t \cdot x_t - w_{t-1} \cdot x_t$ and $r_t = u_t \cdot x_t - y_t$. We estimate $d^2_t$ and $d^3_t$ and sum over $t$ exactly as in the proof of Theorem 4.

All the previous bounds are for algorithms that use a constant learning rate that needs to be set at the beginning, and the optimal choice depends on the norms of the instances, which may not be known in advance. We close this section by considering a variant where we use a variable learning rate based on the norms of instances seen thus far. For simplicity, we deal only with the explicit update case.

Thus, define the explicit update with variable learning rate as

$$
w_t' = f^{-1}(f(w_{t-1}) - \eta_t||w_{t-1} \cdot x_t - y_t||_q)x_t
$$

where now $\eta_t$ is a time-dependent learning rate. The out-of-bound update is as before.

The bound proven below is identical to the fixed $\eta$ version given in Theorem 4 except for an additional factor of five in the second term on the right-hand side.

**Theorem 6:** Fix $p$ and $q$ such that $1/p + 1/q = 1$ and $2 \leq p < \infty$. Let $\eta_t = 1/((p-1)X^2_{p,t})$ where $X_{p,t} = \max_{\tau \leq t} ||x_\tau||_p$. Assume $||u_t||_q \leq U_q$ for all $t$. Then the bounded explicit update for $\Delta_q$ with the variable learning rate $\eta_t = 1/((p-1)X^2_{p,t})$ and parameter $U_q$ satisfies

$$
T \sum_{t=1}^{T} (u_t \cdot x_t - w_{t-1} \cdot x_t)^2 \leq T \sum_{t=1}^{T} (u_t \cdot x_t - y_t)^2 + 5(p-1)X^2_{p,t}U^2_q
+ 2(p-1)X^2_{p,t}U_q \sum_{t=1}^{T-1} ||u_{t+1} - u_t||_q.
$$

**Proof:** We modify the proof of Theorem 4 using the method of [18] for handling the variable learning rate. Fortunately, in filtering, the techniques are much easier than in the prediction setting.

Thus, we consider the quantity $\Delta_q(u_t, w_{t-1})/\eta_t$. By replacing $\eta$ with $\eta_t$ in (20), we see that the proof of Theorem 4 implies

$$
\frac{\Delta_q(u_t, w_{t-1})}{\eta_t} - \frac{\Delta_q(u_{t+1}, w_t)}{\eta_t} \geq \frac{s^2}{2} + \frac{r^2}{2} + \frac{1}{2\eta_t}||u_t||^2_q
- \frac{1}{2\eta_t}||u_{t+1}||^2_q - \frac{U_q}{\eta_t}||u_{t+1} - u_t||_q,
$$

where $s_t = u_t \cdot x_t - w_{t-1} \cdot x_t$ and $r_t = u_t \cdot x_t - y_t$. (Again we set $w_{t+1} = u_t$; also let $X_{p,t+1} = X_{p,T}$.) By substituting

$$
\eta_t = 1/((p-1)X^2_{p,t})
$$

and then noticing that $X_{p,t} \leq X_{p,t+1} \leq X_{p,T}$, we get

$$
(p-1)X^2_{p,t} (\Delta_q(u_t, w_{t-1}) - \Delta_q(u_{t+1}, w_t)) \geq \frac{s^2}{2} + \frac{r^2}{2} + \frac{1}{2\eta_t}||u_t||^2_q - \frac{1}{2\eta_t}||u_{t+1}||^2_q
+ (p-1)X^2_{p,t+1}U_q ||u_{t+1} - u_t||_q.
$$

By [18, Lemma 3.2], we have $\Delta_q(u, w') < 2V$ whenever $||u||_q \leq V$ and $||w'||_q \leq V$, so in particular $\Delta_q(u_{t+1}, w_t) \leq 2V^2$. Remembering that $X^2_{p,t} - X^2_{p,t+1} \leq 0$, we get

$$
(p-1)X^2_{p,t} (\Delta_q(u_t, w_{t-1}) - \Delta_q(u_{t+1}, w_t)) \geq \frac{s^2}{2} + \frac{r^2}{2} + \frac{1}{2\eta_t}||u_t||^2_q - \frac{1}{2\eta_t}||u_{t+1}||^2_q
+ (p-1)X^2_{p,t+1}U_q ||u_{t+1} - u_t||_q.
$$

By summing over $t = 1, \ldots, T$, we get

$$
(p-1)X^2_{p,1} \Delta_q(u_1, w_0) - (p-1)X^2_{p,T+1} \Delta_q(u_{T+1}, w_T) \geq \frac{1}{2} \sum_{t=1}^{T} s^2_t - \frac{1}{2} \sum_{t=1}^{T} r^2_t + \frac{p-1}{2} X^2_{p,1}||u_1||^2_q
- \frac{p-1}{2} X^2_{p,T+1}||u_{T+1}||^2_q
+ 2(p-1) X^2_{p,T} U_q \sum_{t=1}^{T} ||u_{t+1} - u_t||_q.
$$

The result follows by solving for $\sum_{t=1}^{T} s^2_t$, noticing $\Delta_q(u_1, w_0) = ||u_1||^2_q/2$ and then ignoring the negative terms $-(p-1)X^2_{p,T+1} \Delta_q(u_{T+1}, w_T)$ and $4(p-1)X^2_{p,1} U_q^2$.

**VI. GENERALIZED LINEAR MODELS**

We extended framework slightly to cover generalized linear regression. Here we replace the model (1) by

$$
y_k = h(u \cdot x_k + v_k)
$$

where $h$ is a continuous, strictly increasing transfer function. The logistic sigmoid $h(r) = 1/(1 + \exp(-r))$ is a typical example. In the prediction setting (where the learner tries to match $y_k$), the prediction becomes $\hat{y}_t = h(u_{t-1} \cdot x_t)$. In the filtering setting, we would naturally also include the transfer function in the prediction, giving $\hat{y}_t = h(u_{t-1} \cdot x_t)$ for the a priori and $\hat{y}_t = h(u_t \cdot x_t)$ for the a posteriori case. The algorithm then tries to match $\hat{y}_t$ to $h(u \cdot x_t)$. One could in principle still use the squared error $h(u \cdot x_t) - h(u \cdot x_t)^2$ as the performance measure, but this is nonconvex in $u$ and $w$ and actually leads to a very badly behaved optimization problem [19]. We obtain a better behaved problem by using the matching loss for $h$ [19], defined for $y$ and $y'$ in the range of $h$ as

$$
L(y, y') = \int_{h^{-1}(y')}^{h^{-1}(y)} (h(r) - y) dr.
$$

(Notice that by our assumptions $h$ is one-to-one.) It is easy to see that for the identity transfer function $h(r) = r$, we get
\[ L(y_t, y'_t) = \frac{(y - y'_t)^2}{2}; \quad \text{and for the logistic sigmoid } h(r) = \frac{1}{1 + \exp(-r)}, \text{ we get the logarithmic loss} \]
\[ L(y_t, y'_t) = y \ln \frac{y}{y'_t} + (1 - y) \ln \frac{1 - y}{1 - y'_t}. \]

The definition (22) may seem arbitrary, but it is actually one-dimensional Bregman divergence: if we let \( H(r) = \int h(r) \, dr \), then
\[ L(h(a), h(d)) = \Delta_H(d', a). \quad (23) \]

Using a Bregman divergence as a loss naturally generalizes to multidimensional outputs [6], but we shall not pursue that here.

Directly from (22), we obtain a simple expression for its gradient
\[ \nabla_w L(y_t, h(w \cdot x_t)) = (h(w \cdot x_t) - y) x_t. \quad (24) \]

Therefore, the explicit update (14) naturally generalizes to
\[ w_t = f^{-1}(f(w_{t-1}) - \eta(y_t - y_t)x_t) \]
where \( \hat{y}_t = h(w_{t-1} \cdot x_t) \). The implicit update can be generalized similarly; for it we use \( \hat{y}_t = h(\mathbf{w}_t \cdot x_t) \). For these updates we can now prove bounds that have as an additional factor an upper bound on the slope of the transfer function. The techniques are essentially those introduced by [10].

**Theorem 7:** Fix \( p \) and \( q \) such that \( 1/p + 1/q = 1 \) and \( 2 \leq p < \infty \). Let \( h \) be strictly increasing and continuously differentiable with \( c \) such that \( 0 < h(r) < c \) holds for all \( r \) and let \( L \) be the matching loss for \( h \). Assume that \( \|x_t\|_p \leq X_p \) for all \( t \). Then both the explicit update and implicit update for \( \Delta_q \) with learning rate \( \eta = 1/(p-1)C_p^2 \) satisfy
\[ \sum_{t=1}^{T} L(y_t, h(u \cdot x_t)) \leq \sum_{t=1}^{T} L(y_t, h(u \cdot x_t)) + (p-1)C_p^2 \|x_t\|_q^2 \]
for any \( u \in \mathbb{R}^n \).

**Proof:** Consider first the explicit update. As in the proof of Theorem 2, let
\[ d_t = \Delta_q(u, w_{t-1}) - \Delta_q(u, w_t) = \eta(y_t - \hat{y}_t)(u \cdot x_t - w_{t-1} \cdot x_t) - \Delta_q(w_{t-1}, w_t). \]

Using (23), we get
\[ (y_t - \hat{y}_t)(u \cdot x_t - w_{t-1} \cdot x_t) = L(y_t, h(u \cdot x_t)) \]
and
\[ L(y_t, h(u \cdot x_t)) = L(y_t, h(u \cdot x_t)) + L(y_t, \hat{y}_t). \]

Simple calculus shows that \( L(y, \hat{y}) \geq (1/2c^2)(y - \hat{y})^2 \) for all \( y \) and \( \hat{y} \). By combining this with (15), we get
\[ d_t \geq \eta(y_t - \hat{y}_t)(u \cdot x_t - w_{t-1} \cdot x_t) \geq L(y_t, h(u \cdot x_t)) + \eta(y_t - \hat{y}_t)^2 \left( \frac{1}{c} - \eta(p-1)X_p^2 \right). \]

The claim follows by summing over \( t \) as usual.

Consider now the implicit update. We have
\[ d_t = \Delta_q(u, w_{t-1}) - \Delta_q(u, w_t) = \eta(y_t - \hat{y}_t)(u \cdot x_t - w_{t-1} \cdot x_t) - \Delta_q(w_{t-1}, w_t) \]
where now \( \hat{y}_t = h(w_{t-1} \cdot x_t) \). We write
\[ (y_t - \hat{y}_t)(u \cdot x_t - w_{t-1} \cdot x_t) = (y_t - \hat{y}_t)(u \cdot x_t - w_{t-1} \cdot x_t) \]
\[ + (y_t - \hat{y}_t)(w_{t-1} \cdot x_t - w_{t-1} \cdot x_t). \]

Like above, we have
\[ (y_t - \hat{y}_t)(u \cdot x_t - w_{t-1} \cdot x_t) = L(y_t, h(u \cdot x_t)) \]
and
\[ -L(y_t, h(u \cdot x_t)) + L(y_t, \hat{y}_t), \]

Also, since \( w_t \) is the solution to
\[ w_t = \arg \min_w (\Delta_q(w, w_{t-1}) + \eta L(y_t, h(u \cdot x_t))) \]
we have either \( y_t \leq w_t \cdot x_t \leq w_{t-1} \cdot x_t \) or \( w_{t-1} \cdot x_t \leq w_t \cdot x_t \leq y_t \). In either case, \( (y_t - \hat{y}_t)(w_{t-1} \cdot x_t - w_{t-1} \cdot x_t) \leq 0 \). Hence, we have established
\[ d_t \geq \eta(L(y_t, h(u \cdot x_t)) - L(y_t, h(u \cdot x_t))) + L(y_t, \hat{y}_t) - \Delta_q(w_{t-1}, w_t) \]
and can proceed with the explicit update.

Because of how we defined \( \hat{y}_t \), the theorem gives an *a priori* filtering bound for the explicit update and *a posteriori* bound for the implicit update.

When \( h \) is the identity function, we get the results of Section IV with \( c = 1 \). For the logistic sigmoid, \( c = 1/4 \). Thresholded transfer functions, such as \( h(r) = \text{sign}(r) \), correspond to the limiting case \( c \to \infty \), which makes the bound vacuous.

This result generalizes to the nonstationary case (Section V) in the obvious manner; we omit the details.

Our main motivation for considering loss functions other than square loss was that they make the problem involving a nonlinear transfer function computationally simpler, which also allows strong worst case bounds. One might also prefer different loss functions if one assumes a non-Gaussian noise distribution [20]. This is quite different from our framework, where no statistical assumptions are made.

**VII. SIMULATION RESULTS**

The discussion following Theorem 2 suggests that having a sparse target favors having a large \( p \). We illustrate this with a simple filtering simulation.

At time \( t \), the sender sends a bit \( y_t \in \{-1, 1\} \) over a channel. The recipient is required to produce a binary prediction \( \hat{y}_t \in \{-1, 1\} \) about the sent bit. If \( \hat{y}_t \neq y_t \), we say that an error occurred. What the recipient actually observes is
\[ r_t = \sum_{i=0}^{k-1} u_{t+i} y_{t+i} + v_t \]
where \( u \in \mathbb{R}^k \) for some \( k \) describes the channel and \( v_t \) is zero-mean Gaussian noise. The prediction is then \( \hat{y}_t = \text{sign}(\mathbf{w}_{t-1} \cdot x_t) \), where \( x_t = (r_{t-m}, \ldots, r_t, \ldots, r_{t+m}) \in \mathbb{R}^{2m+1} \) and \( r_t \in [-10, 10] \) are distributed uniformly. In both cases, we then renormalize to make \( \|u\|_2 = 1 \). The targets \( u \) from the second distribution are “sparse” in the sense that most of the weight is concentrated on only few components, whereas the targets from the first distribution are “dense.” In both experiments, we
used $k = 10$, $m = 15$ and a signal-to-noise ratio of 10 dB. We compared the explicit update algorithm with $p = 2$ against $p = 2 \ln n \approx 6.9$. (As we remarked after Theorem 2, for $p = 2 \ln n$ we can estimate $(p - 1) ||x||_p^2 \leq 2 \varepsilon (\ln n) ||x||_c^2$.)

Notice that due to the constant learning rate, the weight vectors of the algorithms end up oscillating around the optimum, so the algorithms converge to a nonzero error rate. By using a smaller learning rate, one can reduce the oscillations and thus achieve a smaller final error rate, but this makes the initial convergence slower. The choice of learning rate is thus not straightforward.

We used for $p = 2 \ln n$ the value $\eta = 1/((p - 1)X_p^2)$ as suggested by Theorem 2. This gave final error rates 0.02 in the first experiment and 0.01 in the second one. For $p = 2$ we then chose $\eta$ so that these same final error rates were achieved. For the first experiment, this resulted in $\eta = 0.45/X_2^2$, and for the second one, $\eta = 0.4/X_2^2$.

The development of the error rates over time is shown in Fig. 1. As expected, $p = 2$ gives a faster convergence for dense targets and $p = 2 \ln n$ for sparse targets. The differences here are not large, but they become more apparent if the filter length (i.e., dimensionality of inputs) is increased.

We did not include the implicit updates in this comparison. In other experiments we noticed that for any fixed $p$ and $\eta$, the implicit update has slower initial convergence and smaller final error rate than the explicit one. This can be understood by noticing that by (17), the implicit update always makes a smaller step. Hence, as a crude first approximation, the implicit update is similar to the explicit update with a smaller $\eta$.

VIII. DISCUSSION AND CONCLUSION

We have shown how Bregman divergences based on $p$-norms can be used to derive generalizations of the classical LMS algorithm. This is a direct application of methods recently introduced in machine learning. The resulting $p$-norm algorithms have for large $p$ quite different behavior from the LMS, which is the special case $p = 2$. In particular, both theoretical bounds and preliminary simulations suggest that the large $p$ version has better performance when the target weight vector is sparse. We apply further methods from machine learning to show that also in filtering, the $p$-norm algorithms can be made robust against target shift and can be adapted for generalized linear systems.

The question of applying these techniques to genuinely nonlinear problems remains unsolved. Recently much work has been done in machine learning on applying linear algorithm to nonlinear problems using the so-called kernel trick. This trick works for a large class of algorithms, such as LMS, the support vector machine, or more generally any rotation invariant algorithm [16], [17], [21]. The $p$-norm algorithm for $p \neq 2$ is not rotation invariant, and it remains an open problem whether it can be efficiently nonlinearized with some technique analogous to the kernel trick. For algorithms with similar performance to the $p$-norm algorithm with large $p$, efficient techniques have been found for some kernels [22], but for other kernels the problem is known to be intractable [23]. Further, the computational requirements in signal processing applications may even rule out kernel-style approaches that rely on storing a large number of data points. Thus, the prospects of finding a general nonlinear version of the $p$-norm algorithms do not seem good.
Since $1/p + 1/q = 1$, a straightforward calculation shows that $||u||_p = ||f(w)||_q$ and $w \cdot f(w) = ||w||_2^2$ for all $w \in \mathbb{R}^n$ [8, Lemma 1]. Fix now $\theta = f(w)$ and $\theta' = f(w')$, with $x = \theta' - \theta$. Based on the above, it is easy to verify that $\Delta_p(w, w') = \Delta_p(\theta', \theta)$. Notice the order of the arguments.) Let $G(\theta) = (1/2)||\theta||_p^2$. Since $\Delta_p$ is defined as the error of a first-order Taylor approximation for $G$, we can write

$$\Delta_p(\theta + x, \theta) = \frac{1}{2} x^T H x$$

(25)

where $H_{ij} = \partial^2 G(\xi)/\partial \xi_i \partial \xi_j$ and the derivatives are evaluated at some point $\xi$ on the line between $\theta$ and $\theta + x$. We now estimate the right-hand side of (25) as in [7, Theorem 7.1]. We have

$$H = (2-p) \text{sign}(\xi) |\xi|^{p-2} \left( \sum_i \text{sign}(\xi_i) |\xi_i|^{p-1} - 1 \right) + \delta_i(p-1) |\xi_i|^{p-2} |\xi_i|^{2-p}.$$  

Since we assume $p \geq 2$, we get

$$x^T H x = (2-p) ||x||_p^{2-p} \left( \sum_i \text{sign}(\xi_i) |\xi_i|^{p-1} x_i \right)^2$$

$$+ (p-1) ||\xi||_p^{2-p} \left( \sum_i |\xi_i|^{p-2} x_i^2 \right)$$

$$\leq (p-1) ||\xi||_p^{2-p} \xi \cdot \hat{x}$$

where $\hat{x} = |\xi|^{p-2} \frac{\xi}{|\xi|}$ and $\hat{x}_i = x_i^2$. Since $(1/(p/(p-2))) + (1/(p/2)) = 1$, Hölder’s inequality gives us

$$|\xi \cdot \hat{x}| \leq ||\xi||_{p/(p-2)} ||\hat{x}||_p^2 = ||\xi||_p^{p-2} ||\hat{x}||_p^2$$

and the claim follows.
For $z \in \mathbb{R}^n$, define now $g(z)$ by
\[ g_i(z) = \sum_{j=1}^{\infty} \frac{e^{z_j}}{\sum_{j=1}^{\infty} e^{z_j}}. \] (27)

Let $z_{t-1}$ be such that $g(z_{t-1}) = w_{t-1}$. It is easy to see that such a $z_{t-1}$ exists assuming $w_{t-1,i} > 0$ for all $i$ and $\sum_{i=1}^{n} w_{t-1,i} = 1$. Further, if $z_{t-1}$ is another vector satisfying $g(z_{t-1}) = w_{t-1}$, then $z_{-1}$ and $z_{t-1}$ are the same up to an additive constant, i.e., $z_{t-1,i} = z_{-1,i} + b$ for some $b$ that does not depend on $i$. Equation (26) can now be written as $w = g(z)$, where $z = z_{t-1} - \eta((w_i - x_i) x_t)$. Notice that because of the normalization, the choice of the representative $z_{t-1}$ (i.e., the constant $b$) makes no difference.

Again, we define the implicit and explicit versions of the update. We use an additional parameter vector $z_t$ to present the algorithm, the actual weights being given by $w_t = g(z_t)$. In both cases, we start with $z_1 = 0$. For the implicit exponentiated gradient algorithm, we define $z_t$ by
\[ z_t = z_{t-1} - \eta((w_t - x_t) x_t) x_t \]
and for explicit exponentiated gradient (EG) algorithm by
\[ z_t = z_{t-1} - \eta((w_{t-1} - x_t) x_t) x_t. \]

Thus the implicit update uses as $w_t$ the minimizer of $C_t$, while the explicit update uses an approximation thereof. These updates are analogous to the implicit and explicit updates given previously, with $g$ now replacing $f^{-1}$. However, in this case $g$ is not one-to-one, so we write the update in terms of $z_t$ (which corresponds to $\tilde{f}(w_t)$ in the previous setting) and not directly in terms of $w_t$.

The following lemma gives the analogues of (11) and (15) for relative entropy.

**Lemma 1**: Let $w = g(z)$ and $w' = g(z')$ for some $z, z' \in \mathbb{R}^n$. Then
\[ \Delta_{\text{rel}}(w, w') \leq \frac{1}{8} \left( \max_i (z'_i - z_i) - \min_i (z'_i - z_i) \right)^2 \] (28)
and for any $u \in \mathbb{R}^n$ with $u_i \geq 0$ and $\sum_i u_i = 1$, we have
\[ \Delta_{\text{rel}}(u, w') = \Delta_{\text{rel}}(u, w) + \Delta_{\text{rel}}(u, w') + (z' - z) \cdot (w - u). \] (29)

**Proof**: Equation (29) follows directly from the definition. To prove (28), we first write
\[ \Delta_{\text{rel}}(w, w') = G(z') - (G(z) + g(z) \cdot (z' - z)) \]
where $G(z) = \ln(\sum e^{z_i})$. Notice that $g = \nabla G$. Therefore, $\Delta_{\text{rel}}(w, w')$ is the error in the first-order Taylor approximation of $G(z')$ around $G(z)$, and we have $\Delta_{\text{rel}}(w, w') = (1/2)(z' - z)^T H(z' - z)$, where $H$ is the Hessian of $G$ evaluated at some point between $z$ and $z'$. We have
\[ \frac{\partial^2 G(z)}{\partial z_i \partial z_j} = \delta_{ij} g_i(z) - g_i(z) g_j(z). \]
Therefore we can write $H_{ij} = \delta_{ij} p_i - p_i p_j$ for some $p$ that satisfies $p_i > 0$ and $\sum_i p_i = 1$. Denote now by $X$ a random variable that is obtained by choosing the value $x_i = z_i' - z_i$ with probability $p_i$. Then
\[ (z' - z)^T H(z' - z) = \sum_i p_i x_i'^2 - \sum_{i,j} x_i x_j p_i p_j \]
\[ = \mathbb{E}[X^2] - \mathbb{E}[X]^2 \]
\[ = \text{Var}[X] \]
\[ \leq \frac{1}{4} \max x_i - \min x_i. \]

**Theorem 8**: Assume that $\max_i x_{t,i} - \min_i x_{t,i} \leq R$ for all $l$. Then for any $u \in \mathbb{R}^n$ with $u_i \geq 0$ and $\sum_i u_i = 1$, the explicit EG algorithm with learning rate $\eta = 4/R^2$ satisfies
\[ \sum_{t=1}^{T} (u \cdot x_t - u_{t-1} \cdot x_t)^2 \leq \sum_{t=1}^{T} (u \cdot x_t - y_t)^2 + \frac{1}{4} R^2 \Delta_{\text{rel}}(u, w_0) \]
where $w_0 = g(0)$ is the uniform weight vector.

**Proof**: We analyze the progress $d_t = \Delta_{\text{rel}}(u, w_{t-1}) - \Delta_{\text{rel}}(u, w_t)$. By substituting the explicit EG update into (29) and then using (28), we get
\[ d_t = \eta(y_t - u_{t-1} \cdot x_t) x_t \cdot (u_t - w_{t-1}) - \Delta_{\text{rel}}(u_t, w_t) \]
\[ \geq \eta(y_t - u_{t-1} \cdot x_t) u_t \cdot x_t - \Delta_{\text{rel}}(u_t, w_{t+1}) \]
\[ - \frac{1}{8} R^2 (y_t - u_{t-1} \cdot x_t)^2 R^2. \]

By rearranging terms, we can write this as
\[ d_t \geq \frac{\eta R^2}{2} - \frac{R^2}{2} \eta^2 + \frac{\eta R^2}{2} \left( 1 - \frac{R^2}{4} \right) \]
where $s_t = u \cdot x_t - u_{t-1} \cdot x_t$ and $r_t = u \cdot x_t - y_t$. Since $R^2/4 = 1$, we can apply $\Delta_{\text{rel}}(u, w_{T+1}) \geq 0$ to get
\[ \Delta_{\text{rel}}(u, w_0) \geq \Delta_{\text{rel}}(u, w_1) - \Delta_{\text{rel}}(u, w_{T+1}) \]
\[ = \sum_{t=1}^{T} d_t \]
\[ \geq \frac{4}{R^2} \left( \sum_{i=1}^{T} s_t^2 - \sum_{i=1}^{T} r_t^2 \right) \]
from which the claim follows.

The above theorem assumes the comparison vector $u$ is a probability vector. To deal with arbitrary vectors $u$ with $\|u\|_1 \leq U_1$ for some given bound $U_1 > 0$, we define the scaled explicit EG algorithm as explicit EG with each input $x_t$ replaced by $x_t' = (U_1 x_{t,1}, \ldots, U_1 x_{t,n}) - U_1 x_{t,1} \ldots, - U_{x_{t,n}} \in \mathbb{R}^{2n}$.

**Corollary 1**: Assume $\|x_t\|_\infty \leq X_{\infty}$ for all $t$. Then for any $u \in \mathbb{R}^n$ with $\|u\|_1 \leq U_1$, the scaled explicit EG algorithm satisfies
\[ \sum_{t=1}^{T} (u \cdot x_t - u_{t-1} \cdot x_t)^2 \leq \sum_{t=1}^{T} (u \cdot x_t - y_t)^2 + \ln(2n) X_{\infty}^2 U_1^2. \]

**Proof**: There is some $u' \in \mathbb{R}^{2n}$ with $u'_t \geq 0$ for all $i$ and $\sum_i u'_i = 1$ such that $u' \cdot x_t = u \cdot x_t$ for all $t$. Thus we can apply Theorem 8 with this $u'$. We have $\max_i x_{t,i} - \min_i x_{t,i} = 2U_1 \|x_t\|_\infty$. Since $u_{t,i}$ is the uniform $2n$-dimensional probability vector, we have $\Delta_{\text{rel}}(u, w_{t,0}) \leq \ln(2n)$. The bounds for the implicit EG algorithm can be proven analogously.