Exploiting Sparsity in Adaptive Filters

R. K. Martin
School of Electrical and Computer Engineering
Cornell University
Ithaca, NY 14853, USA
e-mail:
frodo@ece.cornell.edu

W. A. Sethares
Department of Electrical and Computer Engineering
University of Wisconsin
Madison, WI 53706, USA
e-mail:
sethares@ece.wisc.edu

R. C. Williamson
Department of Telecommunications Engineering, RSISE,
Australian National University
Canberra, 0200, Australia
e-mail:
Bob.Williamson@anu.edu.au

C. R. Johnson, Jr.
School of Electrical and Computer Engineering
Cornell University
Ithaca, NY 14853, USA
e-mail:
johnson@ece.cornell.edu

Abstract — **This paper studies a class of algorithms called Natural Gradient (NG) algorithms, and their approximations, known as ANG algorithms. The LMS algorithm is derived within the NG framework, and a family of LMS variants that exploit sparsity is derived.**

Mean squared error (MSE) analysis of the family of ANG algorithms is provided, and it is shown that if the system is sparse, then the new algorithms will often converge faster for a given total asymptotic MSE. Simulations are provided to confirm the analysis. In addition, Bayesian priors matching the statistics of a database of real channels are given. Actual channels are identified by algorithms that exploit these priors and by LMS, showing a realistic application of these algorithms.

I. Introduction

Transmission channels are often sparse, meaning that most of the taps are small and a few taps are large. Optimal equalizers often reflect this sparsity, and typical equalization methods such as the Least Mean Square (LMS) algorithm, the Constant Modulus Algorithm (CMA), and the Decision Directed (DD) algorithm do not exploit this a priori information. Typical approaches to exploiting sparsity are motivated by complexity reduction (at the expense of a small performance loss), which is often accomplished by only updating a subset of the channel model or equalizer taps [1], [2]. In constrast, the Exponentiated Gradient (EG) algorithm [3] has recently been shown to have better performance than typical gradient methods when the target weight vector is sparse [4], [5], though it does not require fewer computations than LMS. We will study algorithms that perform similar to the EG algorithm.

This paper studies a general class of algorithms known as approximate natural gradient (ANG) algorithms, including variants of the EG algorithm. Section II gives a few examples of these algorithms, and gives intuitive reasons for why they are sometimes preferable to standard algorithms such as LMS. Section III analyzes the asymptotic MSE of ANG algorithms. Section IV shows how (Bayesian) prior knowledge of the distribution of the unknown parameters can be used to create natural gradient (NG) algorithms [6], and how a suitable approximation to the general NG algorithm yields the ANG algorithms which we are studying. Section V uses this framework to formally derive a few algorithms. Section VI shows the performance of ANG algorithms as used to identify actual transmission channels, and Section VII concludes.

This work was supported by Fox Digital.

II. Motivation

The goal in this section is to motivate the study of EG-like algorithms. The simplest of the EG algorithms (equation (3.5) in [3]) estimates y_k by

$$\hat{y}_k = \sum_i w_k^i x_k^i. \tag{1}$$

In this section, the weights are assumed positive, though this assumption is relaxed in Section V. The weight vector $\mathbf{w}_k = [w_k^1, ..., w_k^N]$ is updated at each iteration k by

$$w_{k+1}^i = w_k^i + \mu\ w_k^i\ (y_k - \hat{y}_k)\ x_k^i \tag{2}$$

where μ is a small positive stepsize. This can be derived from the EG perspective by an approximation (discussed in section 4.4 of [3]) of the general update strategy

$$w_{k+1}^i = w_k^i\ \exp(\mu \frac{\partial L(y_k, \hat{y}_k)}{\partial w_k^i}) \tag{3}$$

where the loss (or cost) function is

$$L(y_k, \hat{y}_k) = \frac{1}{2}(y_k - \hat{y}_k)^2, \tag{4}$$

The approximation involves taking a Taylor series expansion of the exponential and dropping terms of μ^2 or higher.

One of the main contributions of this paper is showing how EG-like algorithms can be derived as gradient descent algorithms. Consider (2), for example. It can also be derived by estimating y_k by equation (1), but with

$$w_k^i = \gamma(z_k^i) = \frac{1}{4}\ (z_k^i)^2 \tag{5}$$

for some parameter vector $\mathbf{z}$. Now if we think of the algorithm as adapting over z-space, we can use the Euclidean gradient descent

$$z_{k+1}^i = z_k^i - \mu\ \frac{\partial L(y_k, \hat{y}_k)}{\partial z_k^i}. \tag{6}$$

The gradient term becomes

$$\frac{\partial L}{\partial(\hat{y}_k)} \frac{\partial(\hat{y}_k)}{\partial w_k^i} \frac{\partial(w_k^i)}{\partial z_k^i} = -(y_k - \hat{y}_k)\ x_k^i\ \dot{\gamma}(z_k^i).$$

Substituting,

$$z_{k+1}^i = z_k^i + \mu\ \dot{\gamma}(z_k^i)\ (y_k - \hat{y}_k)\ x_k^i. \tag{7}$$

What we are truly interested in is the effective update rule for $\mathbf{w}$, not $\mathbf{z}$, since that is what is used to generate the estimate $\hat{y}_k$ in (1). Since

$$w_{k+1}^i = \gamma(z_{k+1}^i) = \gamma(z_k^i + small\ term),$$

a first order Taylor expansion gives the effective $\mathbf{w}$ update as

$$w_{k+1}^i = \gamma(z_k^i) + \dot{\gamma}(z_k^i)(small\ term), \tag{8}$$

which results in

$$w_{k+1}^i = w_k^i + \mu\ \dot{\gamma}^2(z_k^i)\ (y_k - \hat{y}_k)\ x_k^i. \tag{9}$$

For the γ given by (5), $\dot{\gamma}^2(z_k^i) = \frac{1}{4}\ (z_k^i)^2 = w_k^i$, which gives us equation (2) once again. Thus the EG algorithm can also be thought of as a Euclidean gradient descent algorithm, but the gradient descent is taking place in z-space. This can be contrasted with LMS by noting that LMS is of the form (9), but with $\dot{\gamma}^2(z_k^i) = 1$.

The effect of the extra factor of w in (2) is to scale the stepsize independently for each parameter. When the current estimate of w_k^i is small, it's update is small, and when the current estimate of w_k^i is large, it's update is large. In terms of sparsity, this means that small taps will contribute less to the total Mean Squared Error (MSE), because they will not be wiggling around as much.

This can be validated quantitatively. In [4], the MSE (defined as $E(|y - \hat{y}|^2)$) of a simple EG algorithm with a white input signal is shown to be

$$\xi = \left(1 + \mu\left(\|\mathbf{w}^*\|_1 - \frac{\|\mathbf{w}^*\|_2}{\|\mathbf{w}^*\|_1}\right)\sigma_x^2\right)\sigma_n^2, \tag{10}$$

where $\mathbf{w}^*$ is the target weight vector, while the MSE for LMS is

$$\xi = \left(1 + \mu N \sigma_x^2\right)\sigma_n^2, \tag{11}$$

The key term is $(||\mathbf{w}||_1 - \frac{||\mathbf{w}||_2}{||\mathbf{w}||_1})$, which provides a measure of the sparsity of the target weight vector. When there is considerable sparsity, this term is small, and the stepsize can be made larger for faster convergence without adversely affecting the excess MSE. When this term is large there is not much sparsity, and the stepsize must remain small in order not to increase the MSE. For example, consider the channel $[1, a, a, a, a, 0.5]$. For $|a| = 0.05$, the "measure of sparsity" equals 1.04, while for $|a| = 0.5$, it equals 3.07.

Again consider the alternative parameterization of the linear predictor given by (5). In most signal processing and communications applications the weights are not constrained to be positive, as is required by both methods above of deriving the EG update rule (2). In [3], a modification called the $EG^{\pm}$ algorithm is proposed for this task, but a simpler method (first suggested in [7]) is to modify (5) to

$$w_k^i = \frac{1}{2}\mathrm{sgn}(z_k^i)\ (z_k^i)^2 + \epsilon z_k^i. \tag{12}$$

The Euclidean gradient algorithm in z-space then becomes

$$z_{k+1}^i = z_k^i + \mu\ (|z_k^i| + \epsilon)\ (y_k - \hat{y}_k)\ x_k^i. \tag{13}$$

which allows both positive and negative values of $\mathbf{z}$, and retains the same kinds of sparsity and stepsize advantages as (2). This also allows $\mathbf{w}$ to have positive and negative elements. The effective update rule for $\mathbf{w}$ is

$$w_{k+1}^i = w_k^i + \mu\ (2|w_k^i| + \epsilon^2)\ (y_k - \hat{y}_k)\ x_k^i. \tag{14}$$

Compared to (2), the algorithm has hardly changed, though it is now more useful for signal processing applications. The reason for the ϵ is to keep the update term from vanishing for small $\mathbf{z}$. This allows initialization by zeros, and allows the weights in the model to pass across zero if need be.

III. Analysis

This section derives a theoretical expression for the MSE of the general class of algorithms introduced in section II, in a fashion similar to that in [8] and [9]. Consider the vector update rule (compare to [8]) of

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mu\ D\ (-\bigtriangledown_k + \mathbf{\Gamma}_k), \tag{15}$$

where $\bigtriangledown_k$ is the gradient of the cost surface at time k with respect to $\mathbf{w}_k$; $\mathbf{\Gamma}_k$ is the gradient noise (the difference between the true gradient and the estimate of the gradient used by the algorithm), as in [8]; and D is a diagonal matrix. (In LMS, $D = I$.) Examples of algorithms of this form can be found in the previous section, as in equations (2), (9), and (14); and a more general form is given in the next section, by equation (32). In general, D depends on the current value of the weight estimates, $\mathbf{w}$, but in this section we will assume that $\mathbf{w} \cong \mathbf{w}^*$, which is reasonable when we are considering asymptotic MSE.

Using the definition of the error system $\mathbf{v}_k = \mathbf{w}_k - \mathbf{w}^*$ and the fact that the gradient can be expressed as $2R\mathbf{v}_k$ where $R = E[\mathbf{X}\mathbf{X}^T]$ (see [9]),

$$\mathbf{v}_{k+1} = \mathbf{v}_k + \mu\ D\ (-2R\mathbf{v}_k + \mathbf{\Gamma}_k)$$

or

$$\mathbf{v}_{k+1} = (I - 2\mu DR)\ \mathbf{v}_k + \mu\ D\ \mathbf{\Gamma}_k. \tag{16}$$

At this point, [8] rotates the coordinate system by diagonalizing R. That is not feasible here, because if we diagonalize DR to become $Q^{-1}DRQ$ for an appropriate Q, $Q^{-1}DQ$ will not necessarily be diagonal.

To determine the MSE, we need to find the covariance matrix of $\mathbf{v}_k$. Define $C = E[\mathbf{v}_k\ \mathbf{v}_k^T]$, and assume the system is near convergence (hence $\mathrm{cov}(\mathbf{v}_{k+1}) \cong \mathrm{cov}(\mathbf{v}_k)$), then

$$C = (I - 2\mu DR)\ C\ (I - 2\mu DR)^T + \mu^2 D\mathrm{cov}(\mathbf{\Gamma}_k)D.$$

It is easily shown (see [8]) that $\mathrm{cov}(\mathbf{\Gamma}_k) = 4\xi_{min}R$, where ξ_{min} is the minimum MSE (the MSE for $\mathbf{w} = \mathbf{w}^*$ exactly, and $\mu = 0$). Inserting this and absorbing the 2's and the 4 into μ and μ^2 gives

$$C = (I - \mu DR)\ C\ (I - \mu DR)^T + \mu^2\ \xi_{min}\ DRD. \tag{17}$$

The $(\mu DR)^2$ term can be dropped on the grounds that μ is very small. (An exact solution is given in [10].) Simplifying yields

$$C(-DR)^T + (-DR)\ C \cong -(\mu\ \xi_{min}\ DRD). \tag{18}$$

The negative signs have been included to cast Equation (18) in the form of the Lyapunov equation [11]. In general, this equation cannot be solved for C in closed form. However, due to the structure of the right hand side, a solution is possible, and is given by

$$C = \frac{1}{2}\ \mu\ \xi_{min}\ D. \tag{19}$$

An exact computation of the solution to (17) without the approximation leading to (18) is given in [10], namely

$$C = \frac{1}{2}\ \mu\ \xi_{min}\ D\ \left(I - \frac{\mu}{2}RD\right)^{-1}, \tag{20}$$

which holds provided that all eigenvalues of $\frac{\mu}{2}RD$ are less than one in magnitude. We see that for small μ, the solution given by (19) is indeed valid. For simplicity, (19) will henceforth be used instead of (20).

Widrow et. al. [9] show that the average excess MSE for standard LMS is given by

$$E[\hat{\mathbf{v}}_k^T\ \Lambda\ \hat{\mathbf{v}}_k] = \sum_{p=1}^{n} \lambda_p E[(\hat{v}_{p,k})^2]$$

where $Q^{-1}RQ = \Lambda$, $\hat{\mathbf{v}}_k = Q^{-1}\mathbf{v}_k$, and λ_p is the p^{th} diagonal element of Λ. To use this, note that $\mathrm{cov}(\hat{\mathbf{v}}) = Q^{-1}\mathrm{cov}(\mathbf{v})Q = Q^{-1}CQ$. From (19), the theoretical MSE ξ is then

$$\xi = \xi_{min}\ (1 + \frac{1}{2}\mu \sum_{p=1}^{n} \lambda_p [Q^{-1}\ D\ Q]_{p,p}) \tag{21}$$

where $[.]_{p,p}$ indicates the p^{th} diagonal element.

In the special case when R is diagonal, Q reduces to I, yielding

$$\xi = \xi_{min}\ (1 + \frac{1}{2}\mu \mathrm{tr}(DR)). \tag{22}$$

When D is just the Jacobian used by [4], this simplifies to the result derived in [4]. However, R is often not diagonal, such as in an equalization setting.

Equation (21) provides a basis for a fair comparison between traditional LMS and ANG algorithms. For a given algorithm and system, it is possible to compute the asymptotic total MSE as a function of μ. The stepsizes for the different algorithms can then be adjusted such that at convergence, all have the same MSE. Then other factors (such as convergence time) can be compared fairly.

IV. Exploiting Prior Knowledge

This section derives a general form of reparameterized gradient algorithms that can be understood in terms of prior knowledge or in terms of an underlying cost function. Let

$$w_k^i = \gamma(z_k^i) \tag{23}$$

as in Equation 8 of [4], with $\gamma(z_k^i)$ invertible and differentiable over its domain, though isolated points of discontinuity may be allowed. Each of the different algorithms will have different $L(\cdot,\cdot)$ and/or different $\gamma(\cdot)$ functions. The algorithms we consider will be algorithms which update the entries of $\mathbf{z}$, and thus $\mathbf{w}$.

Mahony and Williamson [7] provide a general discussion of how to encode prior knowledge into learning algorithms using a geometric "preferential structure." The essence of this is to define a metric so that the algorithm evolves over an error surface that is shaped to incorporate the known prior information. For instance, if the ith component is considered to be reliable while the jth component is not, then the algorithm should take larger steps in the jth direction.

Mathematically, the preferential metric is a family of functions $\phi_i(z^i)$ which represent the a priori knowledge (the Bayesian prior) of the ith parameter. The idea of a Bayesian prior is that an unknown parameter that is to be estimated is viewed as a random variable with a (known) probability density function [12]. In this case, $\phi_i(z^i)$ may be viewed as the probability density function (pdf) for the unknown parameter z^i, although we will not insist that the integral of $\phi_i(z^i)$ be normalizable to 1.

Using this concept of priors, Mahony and Williamson [7] show that when the standard parameterization is used ($\gamma(z_k^i) = z_k^i$, so $\mathbf{w} = \mathbf{z}$) the "natural gradient" (NG) algorithm is

$$z_{k+1}^i = \Phi_i^{-1}(\ \Phi_i(z_k^i) - \mu\ \frac{\partial L}{\partial z_k^i}\ \frac{1}{\phi_i(z_k^i)}) \tag{24}$$

where Φ is the indefinite integral of ϕ.

The updates of the NG algorithm (24) can be quite complicated due to the presence of the nonlinearities Φ and Φ^{-1}. A more tractable algorithm can be derived as a first order approximation to (24) by rewriting the update as

$$\Phi_i(z_{k+1}^i) = \Phi_i(z_k^i) - \mu\ \frac{\partial L}{\partial z_k^i}\ \frac{1}{\phi_i(z_k^i)}. \tag{25}$$

Expanding $\Phi_i(z_{k+1}^i)$ in a Taylor Series about z_k^i gives

$$\Phi_i(z_{k+1}^i) = \Phi_i(z_k^i) + \phi(z_k^i)(z_{k+1}^i - z_k^i) + o(\mu^2),$$

When μ is sufficiently small, the higher order terms may be neglected. Substituting into the left hand side of (25) gives

$$\Phi_i(z_k^i) + \phi(z_k^i)(z_{k+1}^i - z_k^i) = \Phi_i(z_k^i) - \mu\ \frac{\partial L}{\partial z_k^i}\ \frac{1}{\phi_i(z_k^i)}.$$

Finally, dividing both sides by $\phi(w_k^i)$ and rearranging gives the algorithm

$$\boxed{z_{k+1}^i = z_k^i - \mu\ \frac{\partial L}{\partial z_k^i}\ \frac{1}{\phi_i^2(z_k^i)}.} \tag{26}$$

At this point, we are still assuming $\gamma(z) = z$, so all of the z's in equation (26) are equivalent to w's. This "approximate natural gradient" (ANG) algorithm may be preferred to (24) in applications since the updates are simpler and can be more readily analyzed.

The use of (26) requires knowledge of the Bayesian prior of the target weight vector, in the form of $\phi_i(z^i)$. For example, the notion of sparsity can be captured by the supposition that with high probability the tap will have have a small value, while with low probability the tap will have a large value. Still assuming $z > 0$, One prior that implies sparsity is

$$\phi(z) = \frac{1}{\sqrt{z}}, \tag{27}$$

since it is large for small z and small for large z. As will be shown in Section V, this prior leads to algorithm (2).

Suppose we have an ANG algorithm of the form (26), with $\gamma(z) = z$. The same ANG algorithm can also be derived as a Euclidean gradient descent ($\phi(z) = 1$) on a modified cost function (i.e. $\gamma(z)$ not necessarily the identity). This is a bit subtle – since the parameterizations are different, the algorithms may be evolving over different z-spaces, but the effective evolution in w-space will be the same. Thus, the prior can be stated directly in terms of ϕ, indirectly in terms of γ, or as some combination of the two. The following proposition makes this precise.

Proposition IV.1 *Let ϕ and γ represent the priors and parameterizations of an ANG algorithm (26) with $\hat{y}$ parameterized as in (23), and let the cost function be given by $L(y, \hat{y})$. If there are functions $\bar{\gamma}$ and $\bar{\phi}$ with*

$$\frac{\dot{\gamma}^2}{\phi^2} = \frac{\dot{\bar{\gamma}}^2}{\bar{\phi}^2}, \tag{28}$$

then $\bar{\gamma}$ and $\bar{\phi}$ are an alternate set of priors and parameterizations that yield the same effective update rule for $\mathbf{w}$.

Proof: From (26), the ANG corresponding to ϕ, γ, and $L(y, \hat{y})$ is

$$z_{k+1}^i = z_k^i - \mu \frac{\partial L}{\partial z_k^i} \frac{1}{\phi_i^2(z_k^i)}.$$

Applying the chain rule yields

$$z_{k+1}^i = z_k^i - \mu \frac{\partial L}{\partial \hat{y}_k} \frac{\dot{\gamma}(z_k^i)}{\phi_i^2(z_k^i)} x_k^i.$$

Note that γ appears here implicitly, since L is a function of $\hat{y}_k$, which in turn is a function of γ. Since y is generated according to $\hat{y}_k = \sum_i w_k^i x_k^i$, what we ought to compare is the effective change of $\mathbf{w}$ as $\mathbf{z}$ changes. Note that

$$w_{k+1}^i = \gamma(z_{k+1}^i) = \gamma(z_k^i + small\ term),$$

so a first order Taylor expansion gives

$$w_{k+1}^i = \gamma(z_k^i) + \dot{\gamma}(z_k^i)(small\ term), \tag{29}$$

which results in

$$w_{k+1}^i = w_k^i + \mu \frac{\partial L}{\partial \hat{y}_k} x_k^i \left(\frac{\dot{\gamma}^2(z_k^i)}{\phi_i^2(z_k^i)} \right). \tag{30}$$

Similarly, the ANG corresponding to $\bar{\gamma}$ and $\bar{\phi}$ is

$$w_{k+1}^i = w_k^i + \mu \frac{\partial L}{\partial \hat{y}_k} x_k^i \left(\frac{\dot{\bar{\gamma}}^2(z_k^i)}{\bar{\phi}_i{}^2(z_k^i)} \right). \tag{31}$$

By (28), these are the same algorithm. Δ

Note that the left-hand and right-hand sides of (28) are both functions of z, yet z is different for each side since the parameterization is different. Thus, both sides of the equation must be separately represented as functions of w, then compared. For example, the algorithm (2) can be thought of as having the standard parameterization $w = \gamma_1(z) = z$ and prior $\phi_1(z) = \frac{1}{\sqrt{z}}$. Thus,

$$\frac{\dot{\gamma_1}^2}{\phi_1^2} = \frac{1}{\phi_1^2(z)} = \frac{1}{\phi_1^2(w)} = w.$$

On the other hand, if we reparameterize w via $\gamma_2(z) = \frac{1}{4}z^2$ and use the uniform prior of $\phi_2(z) = 1$, we have

$$\frac{\dot{\gamma_2}^2}{\phi_2^2} = \dot{\gamma_2}^2(z) = \frac{1}{4}z^2 = w.$$

These two ratios used z differently, but in terms of w, the ratios are the same. Consequently, the update rules given by equations (2) and (9) were the same.

In the course of our proof, we have shown that algorithms incorporating priors and reparameterizations can be written in the general form of (15) (compare to (30), where D_k is given by

$$[D_k]_{i,j} = \left(\frac{\partial \gamma_i}{\partial (z_k^j)} \right)^2 \left(\frac{1}{\phi_i(z_k^j)} \right)^2. \tag{32}$$

This shows how ANG algorithms can be analyzed using the framework of Section III. Also note the similarity between D_k and what Equation (15) of [4] calls the Jacobian. The difference is that the Jacobian in [4] only includes the first factor from (32), whereas our use here is intended to capture the effects of the prior as well.

Using this proposition, the ANG may be derivable from an alternative prior ϕ using the standard parameterization $\gamma(z) = z$. This prior will be called the 'true' prior because it represents the prior beliefs without the confounding influence of the reparameterization function. Alternatively, the ANG may be derivable from a reparameterization using the standard prior $\phi = 1$. In this case, γ can be used to give the cost function over which the algorithm is evolving (in z-space) under the standard Euclidean gradient. These are useful because sometimes it is easier to understand the behavior of an algorithm from the viewpoint of priors, while sometimes it is easier from the perspective of the cost function. This is an important feature of the ANG algorithms, since this translation is not possible with the NG algorithms.

Even so, this proposition should be used with caution. When we obtained equation (29), we ignored terms of order μ^2 or higher. However, if $\dot{\gamma}(z_k^i)$ is small compared to μ, which often occurs near $\mathbf{z} = \mathbf{0}$, the higher order terms are of comparable magnitude to the first order term, and the approximations used in the proposition are invalid.

V. Algorithms

This section uses the framework of the previous section to derive standard algorithms and new algorithms that exploit sparsity. The prior belief that corresponds to the LMS algorithm is that all parameter values are equally likely. Hence $\phi(z) = 1$, $\Phi(z) = z$, and $\Phi^{-1}(v) = v$. The cost function is the mean square cost $L(y_k, \hat{y}_k) = (y_k - \hat{y}_k)^2$, and the parameterization is the standard one, $\gamma(z) = z$. Hence $\frac{\partial L}{\partial w_k^i} = 2(y_k - \hat{y}_k)x_k^i$. Substituting into (24) or (26) gives the LMS algorithm. (In general, when $\phi(z) = 1$, there is no difference between the NG and the ANG algorithms.)

Now consider the prior $\phi(z) = \frac{1}{z}$. This means $\Phi(z) = \ln(z)$ and $\Phi^{-1}(v) = \exp(v)$. Using the standard MSE cost of (4) and the standard parameterization $\gamma(z) = z$ gives the NG algorithm

$$z_{k+1}^i = \exp(\ln(z_k^i) + \mu(y_k - \hat{y}_k)x_k^i z_k^i)$$

which can be rewritten

$$z_{k+1}^i = z_k^i \exp(\mu\ z_k^i\ (y_k - \hat{y}_k)\ x_k^i). \tag{33}$$

This is called the exponentiated gradient algorithm in [5]. The associated ANG algorithm is

$$z_{k+1}^i = z_k^i + \mu\ (z_k^i)^2\ (y_k - \hat{y}_k)\ x_k^i. \tag{34}$$

An equivalent way to derive this ANG algorithm is to let $\gamma(z) = \exp(z)$ and $\phi(z) = 1$. In this case, both the NG and ANG algorithms are

$$z_{k+1}^i = z_k^i + \mu \exp(z_k^i)\ (y_k - \hat{y}_k)\ x_k^i. \tag{35}$$

To see the equivalence of (34) and (35), convert them into w-space. (34) is already in that format (since there we had $\mathbf{w} = \mathbf{z}$), so expand (35) in a Taylor series as in (8). This yields

$$w_{k+1}^i = w_k^i + \mu\ (\exp(z_k^i))^2\ (y_k - \hat{y}_k)\ x_k^i.$$

Since $w = \gamma(z) = \exp(z)$, this is effectively

$$w_{k+1}^i = w_k^i + \mu\ (w_k^i)^2\ (y_k - \hat{y}_k)\ x_k^i,$$

which is the same as equation (34).

Now consider an algorithm with $\gamma(z) = z$ and a prior $\phi(z) = \frac{1}{\sqrt{z}}$. The prior is qualitatively similar to the previous example, but slightly different. The NG update is

$$z_{k+1}^i = \left(\sqrt{z_k^i} + \frac{1}{2}\mu\sqrt{z_k^i}(y_k - \hat{y}_k)x_k^i\right)^2,$$

and the simpler ANG update is

$$z_{k+1}^i = z_k^i + \mu\ z_k^i\ (y_k - \hat{y}_k)\ x_k^i. \tag{36}$$

This is exactly the example introduced in Section II, since $\mathbf{w} = \mathbf{z}$. An equivalent way to derive this ANG algorithm comes from setting $\gamma(z) = \frac{1}{4}z^2$ and $\phi(z) = 1$. This leads to NG and ANG algorithms given by

$$z_{k+1}^i = z_k^i + \mu\ (\frac{1}{2}z_k^i)\ (y_k - \hat{y}_k)\ x_k^i.$$

Converting the ANG algorithm into w-space as in (8) gives

$$w_{k+1}^i = w_k^i + \mu\ (\frac{1}{2}z_k^i)^2\ (y_k - \hat{y}_k)\ x_k^i. \tag{37}$$

Since $(\frac{1}{2}z_k^i)^2 = \frac{1}{4}(z_k^i)^2 = w_k^i$, equations (36) and (37) are the same in w-space. This was also derived in Section II, although the framework here is more formal.

Now we will modify $\gamma(z)$ to allow both positive and negative weights. Suppose that a prior form of $\phi(z) = \frac{1}{\sqrt{z}}$ (for positive weights) appears to fit well. Then the discussion above suggests that an algorithm of the general form of (2) may be effective in a sparse environment. As discussed in Section II, this is equivalent to a Euclidean gradient descent in z-space with $\gamma(z) = \frac{1}{4}z^2$. In order to allow positive and negative coefficients as well as to forbid the algorithm from becoming locked at $w = 0$, the parameterization can modified to be

$$\gamma(z) = \frac{1}{4}\text{sgn}(z)\ (z)^2 + \sqrt{\epsilon}z \tag{38}$$

where $\epsilon > 0$. What is the effect of this modification?

Since $\dot{\gamma}(z) = \frac{1}{2}|z| + \sqrt{\epsilon} > 0$, this is always an increasing function, and equilibrium in the corresponding ANG algorithm can only occur when $y_k = \hat{y}_k$ (or in the degenerate case when $x = 0$). Specifically, ϵ keeps the update term from vanishing for small z, which would have prevented coefficients from changing sign. Now consider the question of how such modifications influence the sparsity prior. The parameterization given by (38) and a Euclidean gradient over z ($\phi(z) = 1$) is equivalent to $\dot{\bar{\gamma}}(z) = 1$ and $\bar{\phi}(z) = \sqrt{\frac{1}{|z|+\epsilon}}$ by Proposition IV.1. This can be shown as follows. Note that

$$(\dot{\gamma}(z))^2 = \frac{1}{4}|z|^2 + \sqrt{\epsilon}|z| + \epsilon, \tag{39}$$

and

$$|\gamma| = \frac{1}{4}|z|^2 + \sqrt{\epsilon}|z|. \tag{40}$$

Thus,

$$\left(\frac{\dot{\gamma}}{\phi}\right)^2 = (\dot{\gamma}(z))^2 = |\gamma| + \epsilon = |w| + \epsilon, \tag{41}$$

and we must also have

$$\left(\frac{\dot{\bar{\gamma}}}{\bar{\phi}}\right)^2 = |w| + \epsilon. \tag{42}$$

If we choose $\bar{\gamma}(z) = z$, then $\dot{\bar{\gamma}} = 1$ and $w = z$, so

$$\frac{1}{(\bar{\phi}(z))^2} = |z| + \epsilon. \tag{43}$$

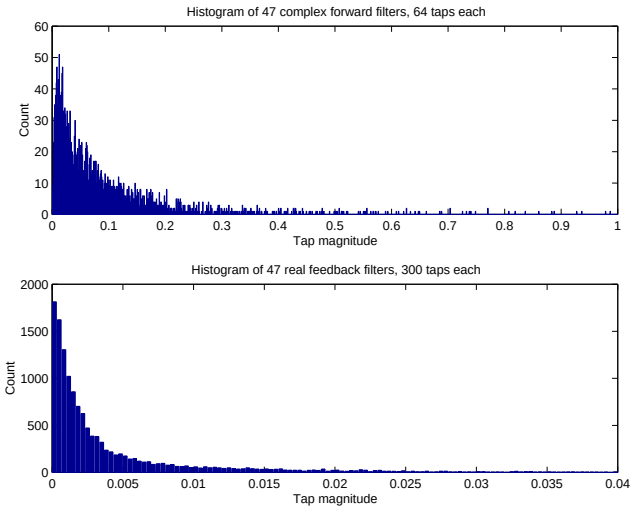

Figure 1: Histograms of the DFE.

This yields $\bar{\phi}(z) = \sqrt{\frac{1}{|z|+\epsilon}}$. Compared to the previous prior of $\phi(z) = \sqrt{\frac{1}{z}}$, there is little difference. The ϵ is a small modification that only changes the algorithm near $z = 0$. The resulting ANG algorithm (which we will call "signed sparse LMS", since it allows for weights of either sign of $\pm$) is

$$w_{k+1}^i = w_k^i + \mu\ (|w_k^i| + \epsilon)\ (y_k - \hat{y}_k)\ x_k^i. \tag{44}$$

Similar derivations can be done for algorithms using the CMA or decision directed cost functions, or for algorithms using ARMA model parameterizations. For a full treatment, see [10].

VI. Performance on Measured Channels

In March 2000, researchers from Cornell University, University of Wisconsin, Australian National University, Applied Signal Technology, and NxtWave Communications met in Philadelphia for field measurements of digital television signals. The results have been compiled into a database of identified channels and associated MMSE decision feedback equalizers. The data from 47 of these channels were used to produce histograms of the magnitudes of the equalizer coefficients. The histograms of the forward filter of the equalizer are well modeled by both exponential priors ($c\ e^{-\alpha|z|}$) and inverse power law (IPL) priors ($\frac{c}{|z|^\alpha+\epsilon}$). The feedback filter was well modeled by an exponential prior, and the channel was well modeled by an IPL prior.

The top plot in Figure 1 shows a histogram of the magnitudes of all the complex taps from the forward equalizers for all of the channels. The bottom plot in Figure 1 is similar, but it is for the (real) feedback equalizer.

Figure 2 shows the histogram of the magnitudes of the complex channel taps, as well as exponential and IPL curves that have been fitted to the histogram. Note the logarithmic scale.

The gain constant c appears in the update rule in such a way that it can be absorbed by the stepsize. The parameter ϵ in the IPL prior and the α that appears in the exponential prior are both subject to changes in the scale of $\mathbf{w}$. Thus,

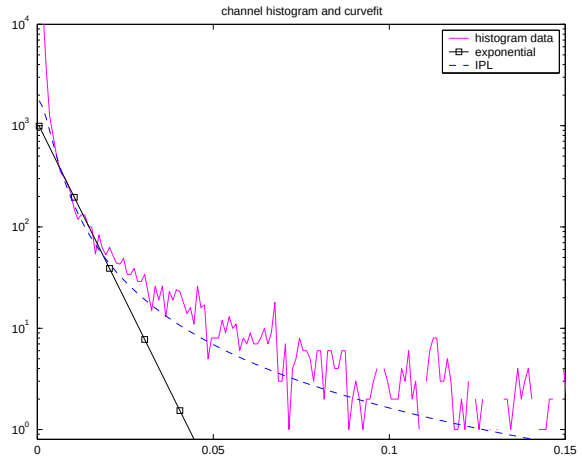

Figure 2: Histogram and curve fits for the channel.

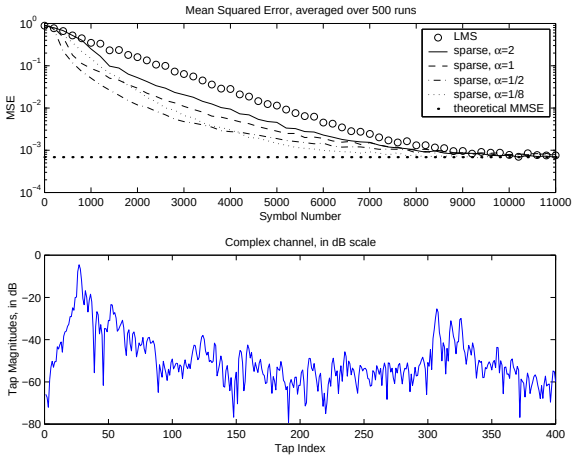

Figure 3: A measured complex channel and plot of convergence rates for channel identification.

a different receiver with a different automatic gain controller will have different values for these parameters.

It is easy to extend the algorithms in this paper to the complex case. Figure 3 shows an example of identification of one of the complex channels used for the histogram. The top plot shows the MSE versus time for both LMS and sparse LMS (with an IPL prior), with a variety of values of α. The bottom plot shows the complex channel in dB scale. A large value of ϵ was used, so as to speed initial convergence (since the model is initialized to zero). Similar simulations were done using nine other channels from the database, with almost identical results.

VII. Conclusions

We have motivated the study of EG-like algorithms by giving intuitive reasons why they often converge faster than traditional algorithms. An analysis of the asymptotic MSE was provided to allow a fair comparison of the algorithms, and a framework for algorithm development was presented. Using this framework we have derived the LMS algorithm and LMS variants that exploit sparsity. The new algorithms were shown to have component-wise modifications to the step size, and the important idea is how the ANG algorithms determine these modifications.

Simulations verify that if one has accurate knowledge of the prior (even without knowledge of the locations of the small taps), then substantial performance gains can be achieved, depending on the initialization. Conversely, if a false prior is assumed, performance degradation often occurs.

Future work may involve a more detailed approach to choosing the stepsize when the system is unknown, analyzing the stability and convergence behavior of the algorithms in greater detail, considering equalization applications in more depth, examining sensitivities of the algorithms to incorrect priors, and theoretically determining the priors of a channel from statistical propagation models.

Acknowledgments

Jaiganesh Balakrishnan and Wonzoo Chung of the Cornell University Blind Equalization Research Group provided many helpful comments.

References

[1] T. J. Endres, R. A. Casas, S. N. Hulyalkar, and C. H. Strolle, "On Sparse Equalization Using Mean-Square-Error and Constant Modulus Criteria," in *The 34th Annual Conference on Information Sciences and Systems*, Princeton, NJ, 2000, vol. 1, pp. TA7b–7–12.

[2] S. Ariyavisitakul, N. R. Sollenberger, and L. J. Greenstein, "Tap-Selectable Decision-Feedback Equalization," *IEEE Transactions on Communications*, vol. 45, no. 12, pp. 1497–1500, Dec. 1997.

[3] J. Kivinen and M. K. Warmuth, "Exponentiated Gradient Versus Gradient Descent for Linear Predictors," *Information and Computation*, vol. 132, no. 1, pp. 1–64, Jan. 1997.

[4] S. I. Hill and R. C. Williamson, "Convergence of Exponentiated Gradient Algorithms," Accepted for publication in *IEEE Transactions on Signal Processing*.

[5] R. E. Mahony and R. C. Williamson, "Riemannian Structure of Some New Gradient Descent Learning Algorithms," in *Adaptive Systems for Signal Processing, Communication and Control Symposium (AS-SPCC)*, Lake Louise, Alberta, Canada, 2000, pp. 197–202.

[6] S. Amari, "Natural Gradient Works Efficently in Learning," *Neural Computation*, vol. 10, no. 2, pp. 251–276, Feb. 1998.

[7] R. E. Mahony and R. C. Williamson, "Prior Knowledge and Preferential Structures in Gradient Descent Learning Algorithms," Submitted to Journal of Machine Learning Research, on August 4, 2000.

[8] B. Widrow, J. R. Glover, Jr., J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, E. Dong, Jr., and R. C Goodlin, "Adaptive Noise Cancelling: Principles and Applications," *Proceedings of the IEEE*, vol. 63, no. 12, pp. 1692–1716, Dec. 1975.

[9] B. Widrow, J. McCool, M. G. Larimore, and C. R. Johnson, Jr., "Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filter," *Proceedings of the IEEE*, vol. 64, no. 8, pp. 1151–1162, Aug. 1976.

[10] R. K. Martin, "Exploiting Sparsity in Adaptive Filters," M.S. thesis, Cornell University, 2001.

[11] T. Kailath, *Linear Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[12] C. P. Robert, *The Bayesian Choice*, Springer, New York, NY, 1994.