Riemannian Structure of Some New Gradient Descent Learning Algorithms

Robert E. Mahony MAHONY@IEEE.ORG
Department of Electrical and Computer Systems Engineering, Monash University, Clayton, Victoria, 3800, Australia

Robert C. Williamson BOB.WILLIAMSON@ANU.EDU.AU
Department of Engineering, Australian National University, Canberra ACT 0200, Australia

Abstract

We consider some generalizations of the classical LMS learning algorithm including the Exponentiated Gradient (EG) algorithm. We show how one can develop these algorithms in terms of a prior distribution over the weight space. Our framework subsumes the notion of "link-functions". Differential geometric methods are used to develop the algorithms as gradient descent with respect to the natural gradient in the Riemannian structure induced by the prior distribution. This allows a Bayesian Riemannian of the EG and related algorithms. We relate our work to that of Amari and others who used similar tools in a different manner. Simulation experiments illustrating the behaviour of the new algorithms are presented.

1. Introduction

The LMS algorithm [2] is very widely used in signal processing and various learning problems. Recently some interesting variants of this algorithm including the *Exponentiated Gradient* algorithm have been developed [10]. This algorithm has been shown (both theoretically and experimentally) to have better performance in situations where the target weight vector is sparse.

More recently [5, 13, 11, 3, 4] a range of general families of gradient descent algorithms inspired by the EG algorithm have been analyzed for both classification and regression problems. In the present paper we will examine these algorithms from a new viewpoint: one which has a clear Bayesian interpretation and which, we believe, provides some nice new intuition.

2. Problem Formulation

Consider the class of linear model relationships with inputs in $\mathbb{R}^N$ and outputs in $\mathbb{R}$; the set of maps $x \mapsto \langle w, x\rangle, \quad x \in \mathbb{R}^N$, where $w \in \mathbb{R}^N$ and $\langle w, x\rangle = w^T x$. For a sequence of data $\{x_1, x_2, ...\}$ assume that the associated outputs are

$$y_k = \langle w_*, x_k\rangle + \eta_k, \tag{1}$$

for an unknown "true" system $x \mapsto \langle w_*, x\rangle$ perturbed by noise η_k. The problem considered is to learn the unknown w_* for the incoming data stream $S_t := \{(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)\}$.

A *learning rule* is a method of determining a sequence of estimates $\{w_1, w_2, \ldots, w_k\}$, where w_k depends on S_k, which "learns" the parameter w_*; that is $w_k \to w_*$ in some sense. Many learning algorithms are based on stochastic gradient descent. The *instantaneous* loss function

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

measures the mismatch between the training sample and the estimated output

$$\hat{y}_k := \langle w_k, x_k\rangle. \tag{3}$$

The Gradient Descent GD learning rule updates the present estimate w_k in the direction of steepest descent of the cost $\mathcal{L}(y_k, \hat{y}_k)$

$$x_{k+1} = w_k - s_k \frac{\partial \mathcal{L}}{\partial w}(y_k, \hat{y}_k). \tag{4}$$

Here $\frac{\partial \mathcal{L}}{\partial w}(y_k, \hat{y}_k)$ is the column vector of partial differentials The scalar $s_k \in \mathbb{R}$ is a step-size scaling factor (or learning rate) which is chosen to control how large an update to w_k is made; in this brief version of the paper we assume s_k is constant. The noise in (1) will locally perturb the convergence of the stochastic gradient algorithm. Under mild conditions the estimate w_k converges asymptotically to a neighbourhood of w_* [12]. In fact, the averaged performance of a stochastic gradient descent algorithm is related to that of a descent algorithm based on non-stochastic data. For this reason it is of interest to analyze the noise free case and in the remainder of the paper *we assume that the measurement* $y_k = \langle w_*, x_k\rangle$ *is unperturbed by noise*. This assumption simplifies the presentation of the principal contribution of the paper: a geometric interpretation of a preferential structure on the parameter space. The geometric structure proposed differs significantly from other recent results in learning theory which use information geometric concepts [7, 9]. Briefly, the key difference is that Amari derives all of his geometry from the likelihood function arising from the underlying statistical problem and takes no explicit account of prior information regarding w_*, whereas for us, prior information takes central place.

Variations on (4) have been proposed recently, including the *exponentiated gradient* (EG) algorithm [10]

$$w_{k+1} = \mathrm{diag}(w_k) \exp.(s_k \frac{\partial \mathcal{L}}{\partial w}(y_k, \hat{y}_k)), \tag{5}$$

where the exponential function $\exp.$ of a vector is simply the vector of the exponentials of the separate entries, and $\mathrm{diag}(w_k)$ is the diagonal matrix with diagonal entries given by the entries of w_k. Thus the ith entry of w_{k+1} is given by $w^i_{k+1} = w^i_k \exp\left(s_k \frac{\partial \mathcal{L}}{\partial w^i}(y_k, \hat{y}_k)\right)$.

3. Preferential Structures and Prior Knowledge

In this section an approach for encoding prior knowledge into learning algorithms by imposing a Riemannian geometric structure on parameter space is proposed; we call this a *preferential structure* for reasons that will become clear below.

Riemannian Metrics A Riemannian metric on $\mathbb{R}^n$ is a bilinear, positive definite inner product on each tangent space $T_w\mathbb{R}^n \cong \mathbb{R}^n$ which varies smoothly in w. We denote a metric by $\langle \cdot, \cdot \rangle_w : T_w\mathbb{R}^n \times T_w\mathbb{R}^n \to \mathbb{R}^n$ and its explicit representation in the natural co-ordinates on $\mathbb{R}^n$ by a positive definite matrix $G_w > 0$ at each point. Thus, for tangent vectors $X, Y \in T_w\mathbb{R}^n$, $\langle X, Y \rangle_w = X^T G_w Y$, and $G_w > 0$ is a smooth matrix function on $\mathbb{R}^n$. At the point $w \in \mathbb{R}^n$ the metric can be thought of a way to measure length of vectors and angles between vectors in $T_w\mathbb{R}^n$.

A Riemannian metric $\langle \cdot, \cdot \rangle_w$ on each $T_w\mathbb{R}^n$ can be used to measure curve length on $\mathbb{R}^n$. Let $\gamma : [0,1] \to \mathbb{R}^N$ be a smooth curve on $\mathbb{R}^n$, then the length of γ is defined to be

$$L(\gamma) := \int_0^1 \sqrt{\langle \dot{\gamma}(\tau), \dot{\gamma}(\tau) \rangle_{\gamma(\tau)}} d\tau. \tag{6}$$

This extends to a classical metric $\delta(u, w)$ measuring distance between two points $u, w \in \mathbb{R}^N$ via the infimum

$$\delta(u, w) := \inf_{\gamma \in \mathcal{H}(u,w)} L(\gamma), \tag{7}$$

$\mathcal{H}(u, w) := \{\gamma : [0,1] \to \mathbb{R}^N : \gamma(0) = u, \gamma(1) = w\}$. That is, $\delta(u, w)$ is the length of the shortest curve connecting u and w. To avoid confusion between Euclidean $\mathbb{R}^N$ we will use $\tilde{\mathbb{R}}^N$ to denote $\mathbb{R}^N$ equipped with the new geometric structure.

Encoding Prior Information Imagine that locally around some point w the Riemannian metric is a constant matrix with diagonal structure, $G_z := \mathrm{diag}(\mu_1^2, \ldots, \mu_n^2)$, $z \in N$ a neighbourhood of w and where $\mu_i > 0$. Then choosing two end points $w(0)$ and $w(1)$ that only vary in the ith component it is easily verified that the shortest length curve between these points is the straight line lying along the co-ordinate axis connecting them. The length of a curve lying along a co-ordinate axis w^i is simply $\mu_i |w^i(0) - w^i(1)| = \delta(w(0), w(1))$. Thus, taking a unit length step in direction w^i with respect to the new geometry translates into a scaled step of length $\frac{1}{\mu_i}$ in the original co-ordinates. That is $\delta(w(0), w(1)) = \mu_i |w^i(0) - w^i(1)| = 1$ if and only if $|w^i(0) - w^i(1)| = \frac{1}{\mu_i}$.

Suppose now that one has some prior knowledge that indicates w^i is likely to be a fairly good estimate of the ith component of the true parameter whereas w^j may be a poor estimate. Then we may choose the metric G_z with $\mu_i \gg \mu_j$ so that a unit step (with respect to the new metric) in direction w^i results in a relatively small change in the Euclidean distance while a unit step in direction w^j results in a significant change Euclidean distance. Intuitively, if our prior knowledge is good and can be coded in this manner then a learning algorithm derived with respect to this new geometric structure should perform better than one which does not incorporate the prior knowledge in any manner. The insight provided by this example is directly applicable to infinitesimal learning steps at a point $w \in \mathbb{R}^n$ since G_w is symmetric and can always be diagonalized locally (to second order terms in a neighbourhood of w).

4. Product Distributions and Diagonal Preferential Metrics

Suppose a prior distribution has a density $\phi : \mathbb{R}^N \to \mathbb{R}_+$ for the true parameter w_* is given and has the form

$$\phi(w) := \prod_{i=1}^N \phi_i(w^i), \tag{8}$$

where each $\phi_i : \mathbb{R} \to \mathbb{R}_+$ is itself a probability density for w^i. Thus given a set (event) $\Omega \subseteq \mathbb{R}^n$ then the probability that $w_* \in \Omega$

$$P(w_* \in \Omega) = \int_\Omega \phi(w) dw. \tag{9}$$

Of course the total probability weight for all of $\mathbb{R}^N$ is $\int_{\mathbb{R}^N} \phi(w) dw = 1$.

Now suppose there exists a preferential metric (represented by G_w) such that

$$\det(G_w) = \phi(w)^2. \tag{10}$$

Once again we can take a set $\Omega \subseteq \mathbb{R}^n$ (Lesbegue measurable) and compute the area of Ω with respect to the preferential metric. The area is given by the integral [1, pg. 240]

$$A_p(\Omega) := \int_\Omega \sqrt{\det(G_w)} dw. \tag{11}$$

Thus, due to the assumed form of G_w,

$$A_p(\Omega) = \int_\Omega \phi(w) dw = P(w_* \in \Omega).$$

In this sense area with respect to the preferential metric is equivalent to density with respect to the p.d.f. ϕ.

Thus, if ϕ is large in a region then the associated metric should also be large, corresponding to large relative area of the region with respect to the preferential structure. Consequently, unit step updates (with respect to the preferential structure) in a gradient descent learning algorithm should translate into small updates of the parameters w. For example, even if the instantaneous cost indicates large changes should be made to the present estimates (perhaps due to noisy data), the prior knowledge (in the form of the preferential structure) ensures that only small steps (relative to the Euclidean metric) are made in areas corresponding to uniformly high p.d.f. ϕ. If the actual true parameter is such that it causes the descent steps to continue to force a change in an unlikely direction with respect to the preferential structure then convergence of the parameter $w_k \to w_*$ will be considerably slower than if the preferential structure was not present. This corresponds to having made the wrong prior assumptions about w_*.

Freedom in choosing G_w from ϕ Equation 10 will be satisfied by arbitrarily many Riemannian metrics for a given p.d.f. ϕ. However, for a product distribution (8) we propose the particular preferential metric

$$G_w := \begin{pmatrix} \phi_1(w)^2 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \phi_N(w^N)^2 \end{pmatrix}. \quad (12)$$

Certainly G_w satisfies (10). Moreover, it generalizes the product distribution structure of ϕ.

We will say that a preferential structure agrees with a prior distribution ψ if for any embedded submanifold of $\mathbb{R}^n$ then $P(\Omega|M) = A_p^M(\Omega)$ If this is true then as seen above in the case of a one-dimensional embedded submanifold of $\mathbb{R}^N$ then the concept of length defined by the preferential metric (which is important for generating learning algorithms) agrees with conditional probability distributions on 1-dimensional subspaces. Computing conditional probabilities for general lower dimensional sets is difficult and we shamelessly dodge this problem in the next lemma by restricting our attention to embedded manifolds lying orthogonal to the co-ordinate axis and exploiting the structure of the product measure.

Lemma 4.1 *Let ϕ be a product p.d.f. of the form (8) and let G_w be the preferential metric given by (12). Then for any embedded manifold $M \hookrightarrow \mathbb{R}^N$ which is orthogonal to the co-ordinate axis and any subset $\Omega \subseteq M$ one has $P(\Omega|M) = A_p^M(\Omega)$. Furthermore, G_w is the unique metric for which this identity holds.*

5. Learning Algorithms on Preferentially Structured Parameter Space

We consider a general learning algorithm of the form

$$w_{k+1} = F(s_k, \frac{\partial \mathcal{L}}{\partial w}, w_k), \quad (13)$$

where $F : \mathbb{R} \times \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$. To make the learning algorithm sensible one would expect that if either $s_k = 0$ or $\frac{\partial \mathcal{L}}{\partial w}(y_k, \hat{y}_k) = 0$ then $F(s_k, \frac{\partial \mathcal{L}}{\partial w}, w_k) = w_k$ and that F is a continuous (or even differentiable) function of its arguments. If F is differentiable in the first variable then

$$\tilde{\gamma}(\tau) := F(\tau, \frac{\partial \mathcal{L}}{\partial w}, w_k),$$

is a $\mathcal{C}^1$ curve in $\mathbb{R}^N$ which passes through w_k for $\tau = 0$. This leads one to study the class of curves

$$\gamma_{(w_k, V_k)} : [0, s_k] \to \mathbb{R}^N$$

such that $\gamma_{(w_k,V_k)}(0) = w_k$, $\dot{\gamma}_{(w_k,V_k)}(0) = V_k$ and V_k is a function of the derivative information $\frac{\partial \mathcal{L}}{\partial w}$.

Ignoring for the moment the question of how to choose s_k and V_k, then one may ask exactly what is the best "curve" $\gamma_{(w_k,V_k)}(s_k)$ to choose given a known preferential structure. For stochastic gradient descent algorithms, the aim is to converge as fast as possible to a neighbourhood of w_* and then stay there. Setting $w_{k+1} = \gamma_{(w_k,V_k)}(s_k)$ for fixed s_k and V_k then one would like to maximize the distance $\delta(w_k, w_{k+1})$ taken at every step (cf. (7)) measured relative to the preferential structure! .

Thus, given that the vectors V_k and the scaling factors s_k are chosen together to guarantee the learning algorithm is *well behaved* (for example in the sense that $\{w_k\}$ will converge to w_* for reasonable data samples) then the curve γ should be chosen to maximize the *distance* traveled in the 'direction' V_k. By measuring distance relative to the preferential structure, prior information is directly incorporated into the update step.

To derive a curve that generates an efficient learning algorithm it is important that the length of the curve is directly related to the step size s_k and to the size of the vector V_k. This is natural for the step size since it is the path length parameter of the curve. However to ensure that length of the update curve is properly related to the size of the vector V_k then it is necessary to further require that the update curve γ evolves at a constant speed with respect to the preferential metric:

$$\sqrt{\langle \dot{\gamma}(\tau), \dot{\gamma}(\tau) \rangle_{\gamma(\tau)}} = \langle V_k, V_k \rangle_{w_k}, \quad \tau \geq 0. \quad (14)$$

By this argument, the 'best' curve to choose for the purpose of generating a learning algorithm is one that satisfies (14) whilst maximizing $\delta(w_k, w_{k+1})$ for given s_k and V_k. Thinking of the question in reverse, then given two points w_k and w_{k+1} one is searching for a curve γ of minimum length and constant velocity (with respect to the preferential structure) that connects the two points. Such length minimizing curves on a general Riemannian manifold are known as *geodesics* and are the analogues of straight lines in Euclidean space.

Geodesics Denote the ijth entry of G_w by g_{ij} and the ijth entry of G_w^{-1} by g^{ij} where the base point w of g_{ij} (resp. g^{ij}) is inferred from context. Define

$$\Gamma_{ij}^k = \frac{1}{2} \sum_{s=1}^{N} g^{ks} \left(\frac{\partial g_{si}}{\partial w^j} - \frac{\partial g_{ij}}{\partial w^s} + \frac{\partial g_{js}}{\partial w^i} \right). \quad (15)$$

The functions Γ_{ij}^k are known as the Christoffel symbols and define the Levi-Civita connection on the Riemannian manifold [1, pg. 322]. A geodesic curve $\gamma := \gamma_{(w_k,V_k)}(\tau)$ satisfies the set of coupled second order ODEs

$$\frac{d^2\gamma^k}{d\tau^2} + \sum_{i,j=1}^{N} \Gamma_{ij}^k \frac{d\gamma^i}{d\tau} \frac{d\gamma^j}{d\tau} = 0, \quad (16)$$

with initial conditions $\gamma(0) = w_k, \frac{d\gamma}{d\tau} = V_k$. Uniqueness and well definedness (at least for small τ) follows from the classical theory of ODEs.

Generating the geodesic requires knowledge of tangent direction V_k. It seems natural to choose V_k equal to the derivative $\frac{\partial \mathcal{L}}{\partial w}$. Formally, however, this derivative is not actually an element of the tangent space of $\mathbb{R}^N$. Defining $D_w\mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial w}\right)^T \in \mathbb{R}^{1\times N}$, then $D_w\mathcal{L}$ is known as the differential of $\mathcal{L}$ and is a row vector (co-tangent vector) and

not a column vector (tangent vector). The differential of $\mathcal{L}$ should be thought of as a differential operator

$$\begin{aligned} D_w\mathcal{L} \quad &: T_w\mathbb{R}^N \to T_{\mathcal{L}(y_k,\hat{y}_k)}\mathbb{R}, \\ &D_w\mathcal{L}[V] := D_w\mathcal{L}\cdot V. \end{aligned}$$

Note that writing $V \in T_w\mathbb{R}^N$ as column vectors then $D_w\mathcal{L}\cdot V$ is just matrix multiplication. There is a one to one correspondence between tangent vectors $V \in T_w\mathbb{R}^N \approx \mathbb{R}^{N\times 1}$ and cotangent vectors $W \in T_w^*\mathbb{R}^N \approx \mathbb{R}^{1\times N}$ induced by the Riemannian metric via the unique correspondence of linear maps

$$\langle V, X\rangle_w = WX, \quad \text{for any } X \in T_w\mathbb{R}^N \approx \mathbb{R}^{N\times 1}.$$

For $W = D_w\mathcal{L}$, the corresponding tangent vector is known as the gradient and is given in local co-ordinates by

$$\mathrm{grad}\mathcal{L} = G_w^{-1}\frac{\partial\mathcal{L}}{\partial w}. \tag{17}$$

When the metric is simply the identity matrix then one obtains the classical *Euclidean* gradient $\mathrm{grad}\mathcal{L} = \frac{\partial\mathcal{L}}{\partial w}$. As Amari [8, 9] has shown (in a slightly different setting) there are advantages to using $V_k = \mathrm{grad}\mathcal{L}$ where the gradient is taken with respect to the preferential structure. Of course the Euclidean gradient $\frac{\partial\mathcal{L}}{\partial w}$ will always provide a descent direction for the cost $\mathcal{L}$ and will generate a sensible learning algorithm. Indeed, for any positive definite matrix $Q > 0$, $V_k = Q\frac{\partial\mathcal{L}}{\partial w}$, generates a descent direction.

Natural Gradient Descent Algorithms The general form of the learning algorithms studied in the rest of the paper is now presented. Let $\mathcal{L}$ be an instantaneous loss function associated with the learning problem given in Section 2. Let G_w be a preferential metric and let s_k be a sequence of scalars which can be thought of as the effective learning rate. Then the learning algorithm we study is given by

$$\boxed{w_{k+1} = \gamma_{(w_k,-\mathrm{grad}\mathcal{L})}(s_k),} \tag{18}$$

where $\gamma_{(w_k,-\mathrm{grad}\mathcal{L})}$ is a geodesic curve with respect to the preferential structure (cf. (16)). The significance of this general algorithm will become more apparent when we consider particular cases of it, which we now do.

6. Preferential Structures for Some Common Learning Algorithms

In this section we interpret some common learning algorithms in terms of preferential structures.

Gradient Descent Algorithm Consider the un-biased or Euclidean preferential structure given by $G_w = I_N$ the Euclidean metric. This metric is associated with a uniform (improper) prior product distribution. The Christoffel symbols are zero, since the metric entries are constant, and geodesics are given by solutions $\frac{d^2\gamma^k}{d\tau^2} = 0$, which are of course just straight lines $\gamma_{(w_k,V_k)}(\tau) := w_k + \tau V_k$. The gradient of the loss $\mathcal{L}$ is simply $\mathrm{grad}\mathcal{L} = \frac{\partial\mathcal{L}}{\partial w}$ Thus, comparing with (4) it is easily verified that the GD algorithm is simply

$$\begin{aligned} w_{k+1} \quad &= \quad \gamma_{(w_k,-\mathrm{grad}\mathcal{L})}(s_k) \\ = \mathrm{diag}(w_k) + \frac{\partial\mathcal{L}}{dw}&(y_k, y\hat{y}_k). \end{aligned}$$

Exponentiated Gradient Algorithm We now derive the EG (Exponentiated Gradient) algorithm within our framework. Previous work [10, 6] has shown that the EG algorithm performs well when only a few entries of w_* are nonzero. Now consider choosing a preferential structure that emphasizes regions where only a few co-ordinates are nonzero. We will choose such a structure and show the EG algorithm (almost) follows from such a choice.

Let $\mathbb{R}^N_* = \{u \in \mathbb{R}^N : u > 0\}$. Consider the following preferential metric defined on $\mathbb{R}^N_*$

$$G_w = \begin{pmatrix} \frac{1}{(w^1)^2} & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \frac{1}{(w^N)^2} \end{pmatrix} \tag{19}$$

This metric is associated with an (improper) product distribution on $\mathbb{R}^N_*$ of the form $\psi(w) := \prod_{i=1}^N \psi_i(w)$, where $\psi_i(w) := \frac{1}{w^i}$. Recalling (16) the geodesic satisfies

$$\frac{d^2\gamma^p}{d\tau^2} - \frac{1}{\gamma^p}\left(\frac{d\gamma^p}{d\tau}\right)^2 = 0, \quad \text{for } p = 1,\ldots,N, \tag{20}$$

with initial conditions $\gamma(0) = w$ and $\dot{\gamma} = V$ for arbitrary $w \in \mathbb{R}^N$ and $V \in T_w\mathbb{R}^N$. The particular structure of the Riemannian metric ensures that the N second order ODEs for the co-ordinates of γ are decoupled. The solution of this equation for each co-ordinate is

$$\gamma^i(\tau) = w^i \exp\left(\frac{\tau}{w^i}V^i\right). \tag{21}$$

Consider the descent direction

$$V_k = \mathrm{diag}(w_k)\frac{\partial\mathcal{L}}{\partial w}(y_k, \hat{y}_k). \tag{22}$$

where w_k denotes the kth iteration of the learning algorithm. This is certainly a descent direction since $\mathrm{diag}(w) > 0$ is a positive definite matrix for $w \in \mathbb{R}^N_*$. Along with the geodesic equation obtained above for the preferential structure chosen this choice of descent direction in (18) leads to the EG algorithm (5):

$$w_{k+1} = \gamma_{(w_k,\mathrm{diag}(w_k)\frac{\partial\mathcal{L}}{\partial w})}(s_k) = (5).$$

An interesting point here is that the descent direction chosen to recover the EG algorithm is actually $V_k = G_w^{-\frac{1}{2}}\frac{\partial\mathcal{L}}{\partial w}$. Thus, though V_k is not actually the gradient $\mathrm{grad}\mathcal{L}$ it is, however, closely related. According to the development undertaken in Section 5 it may be preferable to choose $w_{k+1} = \gamma_{(w_k,\mathrm{grad}\mathcal{L})}(s_k)$. which would be equivalent to choosing $V_k = \mathrm{grad}\mathcal{L} = G_w^{-1}\frac{\partial\mathcal{L}}{\partial w}$. Substituting in (18) gives

$$w_{k+1} = \mathrm{diag}(w_k)\exp.(s_k\mathrm{diag}(w_k)\frac{\partial\mathcal{L}}{\partial w}(y_k,\hat{y}_k)).$$

7. Link Functions and Flat Preferential Structures

In this section the general properties of product preferential structures are studied. It is shown that the link function analysis used in recent literature to analyze the EG algorithm can be obtained as a direct generalization of normal co-ordinates.

Consider a product preferential metric of the form

$$G_w = \begin{pmatrix} \phi_1(w^1)^2 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \phi_N(w^N)^2 \end{pmatrix},$$

where $\phi_i : \mathbb{R} \to \mathbb{R}$ are positive definite functions $\phi_i > 0$. Note that each $\phi_i := \phi_i(w^i)$ is chosen to depend only on its associated variable. This is the case when the ϕ_i are independent prior distributions associated with a product prior $\phi(w) = \prod_{i=1}^N \phi_i(w^i)$. This structure has some special consequences for the structure of learning algorithms derived according to the procedure outlined in Section 5.

Theorem 7.1 *Suppose $\phi_i(w) > 0$ for all $w \in \mathbb{R}$. Then the solution of the geodesic equation is*

$$\gamma^i(t) = \Phi_i^{-1}(tV^i\phi_i(w^i) + \Phi_i(w^i)) \tag{23}$$

where $\Phi_i = \int \phi_i$ (the indefinite integral of ϕ).

By setting $V = \mathrm{grad}\mathcal{L}$, and since

$$\mathrm{grad}\mathcal{L} = G_w^{-1}\frac{\partial \mathcal{L}}{\partial w} = \mathrm{diag}(\phi_1^{-2}(w^1), \ldots, \phi_N^{-2}(w^N))\frac{\partial \mathcal{L}}{\partial w}$$

(18) takes the general form

$$\begin{aligned} w_{k+1} &= \gamma_{(w_k, -\mathrm{grad}\mathcal{L})}(s_k) \\ \Rightarrow \quad w_{k+1}^i &= \Phi_i^{-1}(-s_k \frac{\partial \mathcal{L}}{\partial w^i}\frac{1}{\phi_i(w^i)} + \Phi_i(w^i)) \end{aligned}$$

When $\mathcal{L}$ is the squared loss (2), $\frac{\partial \mathcal{L}}{\partial w^i} = -x_k^i(y_k - \hat{y}_k)$ and

$$\boxed{w_{k+1}^i = \Phi_i^{-1}\left(\frac{s_k x_k^i (y_k - \hat{y}_k)}{\phi_i(w_k^i)} + \Phi_i(w^i)\right)} \tag{24}$$

8. Examples

EG (natural) Here we choose $\phi(w) = 1/w$ and thus from (24) and Table 1 we obtain the algorithm:

$$\begin{aligned} w_{k+1}^i &= \exp\left(s x_k^i (y_k - \hat{y}_k)/w_k^i + \ln(w_k^i)\right) \\ &= w_k^i \exp\left(s x_k^i (y_k - \hat{y}_k)/w_k^i\right). \end{aligned}$$

This is the EG algorithm utilizing the natural gradient. It is only valid for $w > 0$. In order to use this algorithm to learn targets u that are not componentwise sign definite, the $\pm$ trick as presented in [10] could be used.

	$\phi(w)$	$\Phi(w)$	$\Phi^{-1}(x)$
$w > 0$	$\frac{1}{w}$	$\ln(w)$	e^x
$w > 0$	$\frac{1}{w^\alpha}$ $(\alpha \neq 1)$	$\frac{1}{1-\alpha}\frac{1}{w^{\alpha-1}}$	$\frac{1}{1-\alpha}\frac{1}{x^{1/(\alpha-1)}}$
	$\frac{1}{\sqrt{1+w^2}}$	$\mathrm{arcsinh}(w)$	$\sinh(x)$
	$\frac{1}{1+w^2}$	$\arctan(w)$	$\tan(x)$
	$\frac{1}{(1+w^2)^{3/2}}$	$\frac{w}{\sqrt{1+w^2}}$	$\sqrt{\frac{x^2}{1-x^2}}$
$\alpha, w > 0$	$e^{-\alpha w}$	$\frac{-1}{\alpha}e^{-\alpha w}$	$\frac{-\ln(-\alpha x)}{\alpha}$
$\alpha > 0$	$e^{-\alpha^2 w^2}$ $(\alpha > 0)$	$\frac{\sqrt{\pi}\mathrm{erf}(\alpha w)}{2\alpha}$	$\frac{\mathrm{erf}^{-1}(2\alpha x/\sqrt{\pi})}{\alpha}$

Table 1. Possible choices of ϕ, Φ and Φ^{-1} for algorithm (24).

EG(α) Here we choose $\phi(w) = 1/w^\alpha$ $(\alpha \neq 1)$ and thus from (24) and Table 1 we obtain the algorithm:

$$w_{k+1}^i = \frac{1}{1-\alpha}\left(s x_k^i (y_k - \hat{y}_k)(w_k^i)^\alpha + \frac{1}{1-\alpha}(w_k^i)^{\alpha-1}\right)^{\frac{1}{1-\alpha}}$$

Like the EG (natural) algorithm, this algorithm is only valid for $w_k^i > 0$. One can easily check that this algorithm approaches the GD algorithm as $\alpha \to 0$.

Cauchy Product Distribution In this case we consider

$$\phi_i(w^i) := \frac{1}{1 + (w^i)^2}$$

which is the classic (unnormalized) Cauchy distribution. The product distribution $\phi(w) := \prod_{i=1}^N \phi_i(w^i)$ is a proper p.d.f. on $\mathbb{R}^N$. Since the Cauchy distribution does not have a singularity at $w^k = 0$ then the preferential structure is defined on all $\mathbb{R}^N$. From (24) and Table 1 we obtain the algorithm:

$$w_{k+1}^i = \tan\left(s x_k^i (y_k - \hat{y}_k)(1 + (w_k^i)^2) + \arctan(w_k^i)\right).$$

9. Simulations

In order to present simulation results, it was necessary to decide on an appropriate way to compare the different algorithms, and in particular how to choose the step size parameter s. We set their steady state MSE to be equal, and then examined their speed of convergence.

Figures 1 and 2 illustrate several of the algorithms developed above. We set $N = 100$ and chose a target vector w_* with only 4 non-zero coefficients. 2000 examples x_k were generated at random from a uniform distribution on $\{-1, 1\}^N$. Then y_k was generated via $y_k = \langle w_*, x_k \rangle + \eta_k$ where η_k was an iid Gaussian random variable with mean zero and standard deviation 0.06. We ran the EG algorithm, GD, Cauchy and EG(α). The step size s was chosen for each algorithm to (approximately) give the same steady state MSE. Reading the plots row-wise, we used $s_{\mathrm{EG}} = \{0.014, 0.026, 0.040\}$, $s_{\mathrm{GD}} = s_{\mathrm{Cauchy}} = s_{\mathrm{EG}(\alpha)} = 0.003$. The MSE estimates given in the figures were obtain from the last 400 steps of the algorithms.

There are several things to observe in these figures:

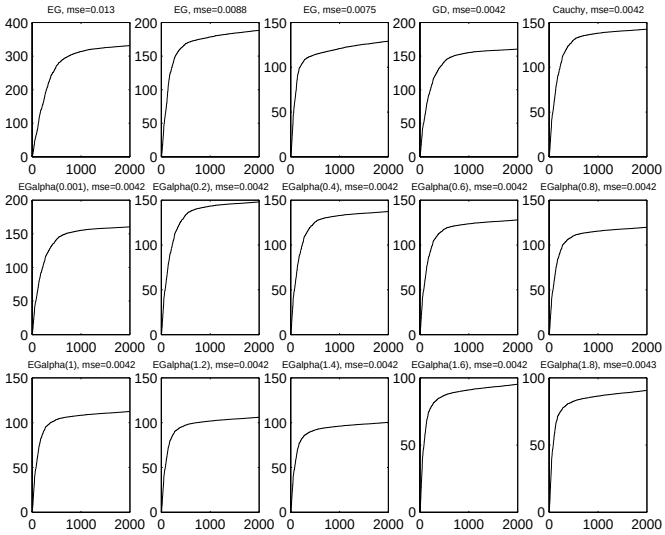

Figure 1. Cumulative Loss for various algorithms (see text).

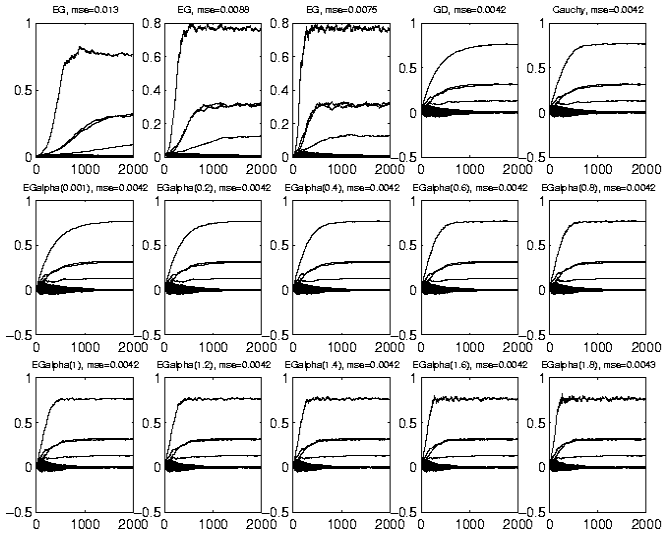

Figure 2. Evolution of weights for various algorithms (see text).

- The EG algorithm takes a very long time to get to complete steady-state, although the large weight vectors converge very fast. This is why increasing the step size decreases the "steady-state" MSE (recall how we measured that).
- The variation of the weights in steady state is homogeneous for GD, but varies according to the magnitude of the weight for the other algorithms. The effect is just noticeable for the Cauchy algorithm, and very noticeable for EG(1.8).
- As expected EG(0.001) behaves essentially identically to GD.
- Although all the algorithms have (roughly) the same steady-state MSE, the detailed dynamics of their convergence is quite different. Inspection of Figure 1 alone suggests EG (with $s = 0.04$) converges faster than GD, and makes a smaller cumulative loss by convergence. But note that the small weight vectors take a very long time to converge for EG.
- The algorithm EG(1.8) incurs the smallest cumulative loss at the "knee" of its cumulative loss curve. This is not really a useful effect though: the knee in this case corresponds to when the largest weight vector reaches steady state (around $k = 200$). At that time none of the other weights has converged. This effect is even more visible for the EG algorithm (Figure 2).

10. Conclusions

We have shown how some new variants on the classical LMS algorithm can be interpreted in terms of a prior over the parameter space. The tools used to do so were based on a natural Riemannian structure. The results complement those developed in the area of information geometry. The simulation experiments illustrate that the interpretation via a prior is easy to reconcile with the actual behaviour of the algorithms. Previous work [10, 6] has shown how the EG algorithm can perform well in real situations where the target weigth vector w_* is sparse. The viewpoint developed here may well serve as a means for fine tuning the venerable LMS algorithm to better exploit prior knowledge one may have in a real problem.

In the full version of this paper we present considerably more detailed connections with Riemannian geometry and relate our work more precisely to that of Amari [8, 9].

References

[1] W.M. Boothby. *An Introduction to Differentiable Manifolds*. Academic Press, London, 1986.

[2] Peter M. Clarkson. *Optimal and Adaptive Signal Processing*. CRC Press, Boca Raton, 1993.

[3] Claudio Gentile and Nick Littlestone. The robustness of the p-norm algorithms. Preprint, DSI, University of Milan, 1999.

[4] Geoffrey J. Gordon. Regret bounds for prediction problems. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, pages 29–40, 1999.

[5] Adam J. Grove, Nick Littlestone, and Dale Schuurmans. General convergence results for linear discriminant updates. In *Proceedings of the Tenth Annual Conference on Computational Learning Theory (COLT-97)*, pages 171–183. ACM Press, 1997.

[6] Simon I. Hill and Robert C. Williamson. Convergence of exponentiated gradient algorithms. Submitted to *IEEE Transactions on Signal Processing*. http://spigot.anu.edu.au/~williams/papers/P128.ps, 1999.

[7] Shun ichi Amari. *Differential-Geometrical Methods in Statistics*. Springer, Berlin, 1985.

[8] Shun ichi Amari. Neural learning in structured parameter spaces — natural riemannian gradient. In *Advances in Neural Information Processing Systems 9*, 1997.

[9] Shun ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10:251–276, 1998.

[10] Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient descent versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–64, 1997.

[11] Jyrki Kivinen and Manfred K. Warmuth. Relative loss bounds for multidimensional regression problems. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 287–293. MIT Press, 1998.

[12] Victor Solo and Xuan Kong. *Adaptive Signal Processing Algorithms*. Prentice-Hall, Englewood Cliffs, 1995.

[13] Manfred K. Warmuth and A.K. Jagota. Continuous and discrete-time nonlinear gradient descent: Relative loss bounds and convergence. Preprint, University of California, Santa Cruz, September 1997.