Journal of Machine Learning Research 1 (2001) 311–355 Submitted 8/00; Published 9/01

Prior Knowledge and Preferential Structures in Gradient Descent Learning Algorithms[*]

Robert E. Mahony MAHONY@IEEE.ORG
Department of Engineering
Australian National University,
Canberra, ACT 0200, Australia.

Robert C. Williamson BOB.WILLIAMSON@ANU.EDU.AU
Department of Telecommunications Engineering
Research School of Information Sciences and Engineering
Australian National University,
Canberra, ACT 0200, Australia.

Editor: Michael Kearns

Abstract

A family of gradient descent algorithms for learning linear functions in an online setting is considered. The family includes the classical LMS algorithm as well as new variants such as the Exponentiated Gradient (EG) algorithm due to Kivinen and Warmuth. The algorithms are based on prior distributions defined on the weight space. Techniques from differential geometry are used to develop the algorithms as gradient descent iterations with respect to the natural gradient in the Riemannian structure induced by the prior distribution. The proposed framework subsumes the notion of "link-functions".

Keywords: Gradient descent, exponentiated gradient algorithm, natural gradient, link-functions, Riemannian metric

1. Introduction

The LMS (least mean-square or Widrow-Hoff algorithm) (Clarkson, 1993) is very widely used in signal processing and various learning problems (Duda, Hart, and Stork, 2001). Recently some interesting variants of this algorithm including the *Exponentiated Gradient* (EG) algorithm have been developed by Kivinen and Warmuth (1997). The EG algorithm has been shown (both theoretically and experimentally) to have better performance in situations where the target weight vector is sparse. The theoretical framework used in that analysis is also relatively new (the so called mistake-bounded framework). An alternative analysis (Hill and Williamson, 1999, 2001) (analogous to more traditional ways of viewing LMS) essentially reproduces the conclusion that EG works well with a sparse target. More recently Grove et al. (1997), Warmuth and Jagota (1997), Kivinen and Warmuth (2001, 1998), Gentile and Littlestone (1999) and Gordon (1999a,b) have analyzed a range of general families of gradient descent algorithms inspired by the EG algorithm for both

[*] An earlier version of parts of this work appeared in pages 197–202 of the Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communication and Control Symposium (AS-SPCC)), S. Haykin and J. Principe (Eds), IEEE Press, New Jersey, 2000.

classification and regression problems. There are several different viewpoints taken in these works, analyzing the algorithms in terms of Bregman divergences, matching loss functions and conjugate priors (for example). In general, such algorithms perform better on a particular class of problems related to prior knowledge of the target weight vector. It is natural to study the relationship between prior information and numerical learning algorithms in order to design effective learning algorithms for new classes of problems. Kivinen and Warmuth (1998) introduced the concept of 'link functions' in order to generalize the derivation of the EG algorithm. An alternative is the natural gradient learning framework developed by Amari (1998). Such a learning algorithm is well motivated as a stochastic gradient descent algorithm derived with respect to the maximally non-informative (or Fisher) information metric; see Amari (1985), Barndorff-Nielsen (1988), Murray and Rice (1993), Douglas and Amari (2000). These algorithms correspond to assuming a maximally non-informative (or Jeffery) prior distribution on the target weight vector.

In this paper we present a new method to derive stochastic gradient algorithms that is closely linked to Bayesian prior information. The approach taken is to link prior distributions to a Riemannian structure on weight space that is called a *preferential structure*. In the case of product prior distributions the connection between a preferential structure and a Bayesian prior distribution may be made relatively precise. The framework proposed leads to a constructive method to design stochastic gradient descent algorithms that are adapted to perform well under certain prior assumptions. Once a stochastic gradient descent algorithm has been designed for a certain application the numerical cost of its implementation is of the same order as that of the LMS algorithm. A theorem is proved showing that (subject to some technical conditions) a stochastic gradient algorithm designed according to the framework proposed is, on average and when the prior assumptions hold true, locally the most efficient stochastic gradient descent algorithm. The preferential structure proposed provides an interpretation of the EG algorithm in terms of a product prior distribution that heavily weights zero against non-zero weight vector entries. The intuition associated with the prior weighting fits with the observed numerical advantages of the EG algorithm. Using the framework of prior distributions and preferential structures several new numerical learning algorithms are derived based on prior distributions of particular interest. Simulations are given comparing relative performance of the algorithms proposed. The proposed framework also provides an interpretation of the role played by link functions in the development of Kivinen and Warmuth (1998). The key contribution of the paper is in providing a new tool in the optimization of stochastic gradient descent algorithms for real world applications.

Section 2 of the paper reviews the learning problem considered and relates the approach taken to the previous work of Amari (1997, 1998). In Section 3 general preferential structures are motivated and defined. Section 4 concentrates on the special case where the prior distribution in parameter space is a product distribution. The preferential (natural) gradient descent algorithm is introduced in Section 5. In Section 6 existing algorithms (such as the EG algorithm) are interpreted in terms of prior distributions and preferential structure. Section 7 shows the connection between "link functions" and the proposed framework. Some examples of different algorithms are presented in Section 8. An analysis of local performance of different learning algorithms is provided in Section 9. Results of simulation experiments for the examples considered in Section 8 are given in Section 10.

Finally, in order to aid readers unfamiliar with Riemannian geometry, in Appendix A we have presented a gentle introduction to the basic ideas used in this paper.

2. Problem Formulation

In this section the framework for the learning problem considered is presented. The conceptual difference between the proposed approach and that based on recent developments in statistical geometry (cf. Amari (1998)) is outlined. The gradient descent (or LMS) algorithm and the exponentiated gradient algorithms are presented.

Consider the class of linear model relationships with inputs in $\mathbb{R}^N$ and outputs in $\mathbb{R}$; the set of maps

$$x \mapsto \langle w, x\rangle, \ \ x \in \mathbb{R}^N,$$

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 there are associated outputs

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

generated by an unknown "true" system $x \mapsto \langle w_*, x\rangle$ perturbed by noise η_k. Depending on the problem setting and the type of analysis to be attempted, different assumptions can be made on the noise. We do not make any assumptions here, but merely mention that there usually is noise because it will ultimately govern the tradeoff between convergence speed of the algorithms considered and their steady-state error (how much the estimates "jiggle about" the true weight vector w_*).

The problem considered is to learn the unknown w_* for the incoming data stream

$$S_k := \{(x_1, y_1), (x_2, y_2), \dots , (x_k, y_k)\}. \tag{2}$$

In this paper we make no assumptions about the sequence (x_k) except in Section 9. (This is because apart from in that section we are not actually making a performance analysis.) This problem is known as a supervised learning problem since to obtain the training sequence S_k one needs both a set of trial data points $\{x_k\}$ and the measured outputs $\{y_k\}$ which would in practice be supplied by a "supervisor" (human or machine). In the presence of the noise η_k this problem becomes one of parametric statistical inference, that of finding the statistical model p_{w_*} which best explains the observed data.

A *learning rule* is a method of determining a sequence of estimates $\{w_1, w_2, \dots , w_k\}$, where w_k depends on S_k, which "learns" the parameter w_*, that is $w_k \to w_*$. Many practical learning algorithms proposed in the literature are based on stochastic gradient descent schemes; see for example Fine (1999), Hassoun (1995), Duda et al. (2001). Let

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

denote the estimated output. The *instantaneous* loss function

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

measures the mismatch between the training sample y_k and the estimated output $\hat{y}_k$.

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

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

Here $\frac{\partial \mathcal{L}}{\partial w}(y_k, \hat{y}_k)$ is the column vector of partial differentials $\frac{\partial \mathcal{L}}{\partial w^i}$, for $i = 1, \dots, N$. 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 most current learning applications the update is kept constant. The noise in (1) will locally perturb the convergence of the stochastic gradient algorithm, although, as long as the measurements are drawn from a non-degenerate distribution, the estimate w_k converges asymptotically to a neighbourhood of w_* (Solo and Kong, 1995). More specifically, it can be shown that under mild assumptions the stochastic gradient descent algorithm will follow a trajectory "close" to the trajectory of an "averaged equation". The exact meaning of "close" depends on the details of the analysis (Solo and Kong, 1995), but the important point is that they are closer as the step size gets smaller and the noise gets smaller. Thus it is common practice in analysing or developing stochastic gradient descent algorithms to pretend at first there is no noise present, but afterwards check performance with noise. This is exactly the route we will adopt in the present paper. Thus we 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.

Recent developments in statistical geometry (Amari, 1985, Murray and Rice, 1993, Barndorff-Nielsen, 1988, Douglas and Amari, 2000) are based on providing a geometric interpretation of the problem of parametric statistical inference, the problem of computing one model among a parametrized set of statistical models which best describes an observed set of noisy measurements. The underlying paradigm is to find an intrinsic geometry for the problem (based on statistical precepts) which is independent of the particular parametrization of the statistical model. The key developments in statistical geometry revolve around a geometry on the space of conditional probabilities derived from a likelihood function and the associated affine action of the set of random variables acting on the space of probabilities. The Riemannian metric used is typically the Fisher metric (also known as the maximally non-informative metric). This metric is derived from cross-correlation of random variables and leads to an associated prior distribution known as the Jeffery prior or maximally non-informative prior. The more sophisticated geometry developed (involving parallel transport and affine connections) is related to invariance of various statistical divergence measures under the action of covariant differentiation.

In the present paper we take a different approach. We assume that there is significant prior information available and that this information is coded directly in terms of the weight vectors for the learning problem. As a conseqeunce, the given parametrization (in this case the linear weight vector) of the problem contains important information. It is clear that the maximally non-informative geometry (generated by the Fisher metric) is not a suitable structure for analysis of the learning problem considered. To further simplify the development we restrict our analysis to the deterministic learning problem where the only statistical information that needs to be considered is the prior information. This perspective on the problem considered is quite different from that proposed by Amari (1998) and appears to provide a means to understand a number of known learning algorithms in a generic manner.

Variations on the classical gradient descent or LMS algorithm (5) have been recently proposed as prototype learning algorithms. Of particular interest is the *exponentiated gradient* (EG) algorithm (Kivinen and Warmuth, 1997)

$$w_{k+1} = \mathrm{diag}(w_k) \exp.(-s_k \frac{\partial \mathcal{L}}{\partial w}(y_k, \hat{y}_k)), \tag{6}$$

where the exponential function exp. of a vector is 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

$$w_{k+1}^i = w_k^i \exp\left(-s_k \frac{\partial \mathcal{L}}{\partial w^i}(y_k, \hat{y}_k)\right).$$

It is known that the EG algorithm performs better than the GD algorithm if the true parameter w_* contains relatively few non-zero entries (Kivinen and Warmuth, 1997). It is not surprising that it performs worse than the GD algorithm if the true parameter has many non-zero entries. Thus, to choose the most efficient learning algorithm for a given application one would exploit any prior knowledge available regarding the nature of the true weight vector to choose between the GD or the EG algorithm.

The above discussion leads one to pose the question: *How is it possible to use prior knowledge about the true weight vector in a given application to design a more efficient learning algorithm?* The remainder of the paper is devoted to presenting *an approach* to answering this question. Before entering into the technicalities it is worth mentioning how the proposed results may be used in practice. For a real world problem involving a class of linear model relationships it is often very difficult to understand and model prior information based on physical arguments. However, it is generally a simple, if time consuming, matter to acquire data in real world operating environments. An estimate of the distribution of true weight vectors can then be inferred from the data by running a standard gradient descent algorithm and recording the weights it converges to and then estimating a distribution over those weights. The experimental prior distribution can then be used as the basis of an optimized design using the theory presented in the sequel. In particular, if the prior distribution obtained is a product distribution the preferential structure is given by (13) and the algorithm is given by (40) or an approximation of this equation. An example of this determination and use of an “empirical prior” has been presented by Martin et al. (2001).

3. Preferential Structures and Prior Knowledge

In this section an approach for encoding prior knowledge into learning algorithms by imposing a Riemannian geometry on parameter space is proposed. In the context of learning theory we propose to call this geometric structure a preferential structure.

3.1 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

$$\langle \cdot, \cdot \rangle_w : T_w\mathbb{R}^N \times T_w\mathbb{R}^N \to \mathbb{R}^N$$

that may be represented explicitly 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 as a way to measure the 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

$$L(\gamma) := \int_0^1 \sqrt{\langle \dot{\gamma}(\tau), \dot{\gamma}(\tau) \rangle_{\gamma(\tau)}} d\tau. \tag{7}$$

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

$$\delta(u, w) := \inf_{\gamma \in \mathcal{H}(u,w)} L(\gamma), \tag{8}$$

where

$$\mathcal{H}(u, w) := \{\gamma : [0, 1] \to \mathbb{R}^N : \gamma(0) = u, \gamma(1) = w\}.$$

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 a non-Euclidean geometric structure.

3.2 Encoding Prior Information

By a linearization locally around any point $w(0)$, a Riemannian metric may be written

$$G_z := \operatorname{diag}(\mu_1^2, \dots, \mu_N^2) + \mathbf{O}(\|z - w\|)$$

for $z \in U$ a neighbourhood of w and where $\mu_i > 0$. 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

$$\delta(w(0), w(1)) = \mu_i |w^i(0) - w^i(1)| = 1 \quad \Leftrightarrow \quad |w^i(0) - w^i(1)| = \frac{1}{\mu_i}$$

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.[1] Thus even if the instantaneous cost indicates large changes should be made in the direction w^j (perhaps due to noisy data), the prior knowledge (in the form of the chosen metric) would ensure that only small steps (relative to the Euclidean metric) are made. 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 $\mathbf{O}(w)$ terms in a neighbourhood of w). In Section 5 we show how to generate practical learning algorithms that respect the geometric structure generated by an arbitrary Riemannian metric.

Definition 1 *Consider a learning problem of the form outlined in Section 2. A* preferential structure *is a Riemannian metric on parameter space, called the* preferential metric*, that encodes certain prior knowledge for the learning problem. A learning problem along with a preferential structure is said to have a* preferentially structured *parameter space.*

This definition is clearly inadequate (so far) as a technical tool since it does not provide any quantitative manner to generate a preferential structure. In Section 4 a connection is drawn between product prior distributions and diagonal preferential structures that provides a quantitative connection for a large class of interesting problems.

Remark 2 *The definition of preferential structure proposed (Definition 1) makes no explicit reference to the underlying statistics of the problem (e.g. via the likelihood function). The information geometric structure presented by Amari (1985), Murray and Rice (1993), Barndorff-Nielsen (1988) satisfies Definition 1 given prior knowledge of the noise characteristics of the measurements and no prior distribution on the weights (Amari, 1998). The class of algorithms considered in the present paper follows from the assumption of deterministic measurements and a prior distribution on the weights.*

4. Product Distributions and Diagonal Preferential Metrics

In this section we consider the situation when prior knowledge is quantified via a Bayesian prior probability distribution. There are numerous arguments (Robert, 1994) why this is a “good” way of encoding prior knowledge. Our goal in the present section is to relate a prior probability distribution over the parameter space to a preferential metric.

We focus on the particular case where the prior is a product distribution in which case (as we show) there is a unique preferential metric that naturally recodes the prior distribution into a preferential structure. The actual application of these structures to gradient descent learning algorithms is considered in the next section.

[1] An illustrative example of this effect is given in Figures 7 and 8 for the particular geometry of the EG algorithm.

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

$$\phi(w) := \prod_{i=1}^{N} \phi_i(w^i), \tag{9}$$

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

$$P(w_* \in \Omega) = \int_{\Omega} \phi(w) dw. \tag{10}$$

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

$$\det(G_w) = \phi(w)^2. \tag{11}$$

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 (Boothby, 1986, pg. 240)

$$A_p(\Omega) := \int_{\Omega} \sqrt{\det(G_w)} dw \tag{12}$$

(the p denotes "preferential".) That is, the volume element in the new geometry is scaled by the factor $\sqrt{\det(G_w)}$ with respect to Lesbegue integration in the co-ordinates w. 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.

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_*.

We will show below (section 9) that the above intuitive justification is sound in a more precise sense: modification of a gradient descent algorithms by a preferential metric satisfying (11) leads to an algorithm with faster local convergence.

4.1 Freedom in choosing G_w from ϕ

Equation 11 will be satisfied by arbitrarily many Riemannian metrics for a given p.d.f. ϕ. However, for a product distribution (9) we propose the particular preferential metric

$$G_w := \begin{pmatrix} \phi_1(w)^2 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \phi_N(w^N)^2 \end{pmatrix}. \tag{13}$$

Certainly G_w satisfies (11). Moreover, it generalizes the product distribution structure of ϕ. To see this one must consider the probability of sets (events) Ω which have zero measure in $\mathbb{R}^N$. We will restrict our attention to sets Ω which are embedded submanifolds of $\mathbb{R}^N$. In general, let M be an embedded submanifold of $\mathbb{R}^N$ and let $\Omega \subseteq M$ be a subset of M. With respect to the p.d.f. ϕ on $\mathbb{R}^N$ then for any event Ω of this form $P(\Omega) = 0$. However, the conditional probability $P(\Omega|M)$ should not be zero. Moreover, this should be related to an area integral on M relative to the geometric structure inherited on M via the embedding $M \hookrightarrow \mathbb{R}^N$ from the preferential structure.

We will say that a preferential structure agrees with a prior distribution ϕ if for any embedded submanifold of $\mathbb{R}^N$ then

$$P(\Omega|M) = A_p^M(\Omega)$$

where

$$A_p^M(\Omega) := \int_\Omega \sqrt{\det(G_w^M)}dw$$

where G_w^M is the metric projected onto the manifold M; see (15) below. If this is true then the concept of length defined by the preferential metric (which is important for generating learning algorithms) agrees with conditional probability distributions in the particular case of 1-dimensional subspaces. Computing conditional probabilities for arbitrary 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 3 *Let ϕ be a product p.d.f. of the form (9) and let G_w be the preferential metric given by (13). Then for any embedded manifold $M \hookrightarrow \mathbb{R}^N$ which is orthogonal to a subset of the co-ordinate axes and any subset $\Omega \subseteq M$ one has*

$$P(\Omega|M) = A_p^M(\Omega). \tag{14}$$

Furthermore, G_w is the unique metric for which this identity holds.

Proof If an m-dimensional manifold $M \hookrightarrow \mathbb{R}^N$ is orthogonal to the co-ordinate axes then locally one can choose m co-ordinates $z = (w^{i_1}, \dots, w^{i_m})^T \in \mathbb{R}^m$ which act as local co-ordinates on M. Note that M is characterised locally by holding the remaining $N - m$ co-ordinates constant (Boothby, 1986, pg. 75). Due to the nature of the product distribution the conditional p.d.f. on M is given by integrating out ϕ with respect to the $N-m$ 'non-active' co-ordinates to obtain

$$\phi^M(w^{i_1}, \dots, w^{i_m}) := \prod_{k=1}^{m} \phi_{i_k}(w^{i_k})$$

Thus, written in terms of the local co-ordinates z the probability of Ω conditioned on M is the Lesbegue integral

$$P(\Omega|M) = \int_{\Omega} \phi^M(z)dz.$$

We deliberately avoid the rigorous approach to dealing with conditional probabilties when the conditioning event has probability is zero; see for example (Shiryaev, 1996, Section II.7) or (Hoffmann-Jorgensen, 1994, Chapter 10). Chang and Pollard (1997) have presented an alternative formal approach.

Think of the local co-ordinates as maps into $\mathbb{R}^N$ via the correspondence $w^i : \mathbb{R} \to \mathbb{R}^N$,

$$w^{i_k}(z^k) := (0, \dots, 0, z^k, 0, \dots, 0)^T$$

where z^k occurs in the i_kth entry of the vector. Thus, the local co-ordinate map around a point $w_0 \in M$ can be written as a vector function $w_M : \mathbb{R}^m \to \mathbb{R}^N$

$$w_M(x) := w^{i_1}(z^1) + \cdots + w^{i_m}(z^m) + w_0.$$

We write $w_M = (w_M^1, \dots, w_M^N)^T \in \mathbb{R}^N$ for the co-ordinates of $w_M \in \mathbb{R}^N$. The induced preferential metric on M with respect to the local co-ordinates is

$$G_w^M = dw_M^T \, G_w \, dw_M = \begin{pmatrix} \phi_{i_1}(z^1)^2 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \phi_{i_N}(z^m)^2 \end{pmatrix}. \tag{15}$$

Here $dw_M \in \mathbb{R}^{N \times m}$ with entries $(dw_M)_{pq} = \frac{\partial w_M^p}{\partial z^q}$, $p = 1, \dots, N$, $q = 1, \dots, m$. Now observe that

$$\det(G_w^M) = \prod_{k=1}^{m} \phi_{i_k}(z^k)^2 = (\phi^M(z))^2.$$

It follows from the definition of area on a Riemannian manifold (12) that $P(\Omega|M) = A_p^M(\Omega)$.

To show uniqueness assume that $G_w > 0$ (with entries g_{ij}) is an arbitrary preferential structure that satisfies (14). Observe firstly that the diagonal elements of G_w are uniquely defined by considering the case of one-dimensional manifolds orthogonal to the co-ordinate axis. This follows since the induced preferential metric is just the diagonal element g_{ii} while the conditional probability is $\phi^M(w^i)$.

Consider an arbitrary two-dimensional submanifold orthogonal to the co-ordinate axis with local co-ordinates w^i and w^j. Then

$$G_w^M = \begin{pmatrix} g_{ii} & g_{ij} \\ g_{ji} & g_{jj} \end{pmatrix}.$$

To ensure that (14) holds on all subsets Ω then locally on M one must have $\phi^M(w^i, w^j)^2 = \det(G_w^M)$ where $\phi^M(w^i, w^j) = \phi_i(w^i)\phi_j(w^j)$ is the product measure induced on the subspace spanned by the i and j co-ordinates. Computing the determinant yields

$$\begin{aligned}\det(G_w^M) &= g_{ii}g_{jj} - g_{ij}g_{ji}\\ &= \phi_i(w^i)^2\phi_j(w^j)^2 - g_{ij}^2\\ &= \phi^M(w^i, w^j)^2 - g_{ij}^2,\end{aligned}$$

where we have used symmetry of G_w as well as the identification of the diagonal elements with $\phi_i(w^i)^2$. Since M and Ω are arbitrary it follows that $g_{ij} = 0$ for all $i \neq j$. ■

Remark 4 *It should be noted that except for some changes in notation the development undertaken above is valid for any p.d.f. of the form*

$$\phi(w) = \prod_{i=1}^{N} \phi_i(w)$$

where each $\phi_i(w)$ is a probability distribution for w^i on $\mathbb{R}$ but may depend on all the variables w. This is important since product measures $\phi_i(w) := \phi_i(w^i)$ induce a preferential structure which is isometric to Euclidean space (cf. Section 7) whereas a more general product distribution, even though it induces a diagonal preferential metric, will not induce a Euclidean preferential structure.

4.2 General Improper Priors

In this subsection we explore the implications of $\phi(w)$ being an *improper* distribution (i.e. one for which $\int_\Omega \phi(w)dw \neq 1$ and may in fact be infinite). There are two key reasons for doing this:

1. The two main example algorithms we consider (standard Gradient Descent where $\phi(w) = 1$ and the EG algorithm where $\phi(w) = 1/|w|$) correspond to improper priors;

2. In practice only a local comparison between performance of stochastic gradient descent learning algorithms is possible (a global comparison seems analytically intractable; cf. Section 9) and consequently only conditional probabilities, in a neighbourhood of a given point are considered. Thus, the question of improper or proper priors does not truly affect the conclusions of the paper.

The above development has been undertaken for proper probability distributions ϕ and has looked for Riemannian metrics which correspond to these probability distributions. However, one may look at the problem in the opposite direction and ask the question, *given a Riemannian metric $g(w)$, then does the distribution*

$$\phi(w) := \sqrt{\det(g(w))} \tag{16}$$

mean anything in a probabilistic sense? Below we will argue that it does. Note that for an arbitrary Riemannian metric G_w the distribution ϕ generated is not in general a proper probseeability distribution; it is highly unlikely that

$$A = \int_{\mathbb{R}^N} \sqrt{G_w} dw = 1.$$

It can be interpreted as an *improper* distribution. Bayesian statisticians often deal with improper prior distributions; the most obvious one is the uniform measure over $\mathbb{R}$. Whilst there are undoubtably technical difficulties arising from such improper priors (see e.g. Robert, 1994), one can generally get away with their use as long as the posterior distributions are well defined. Furthermore, it is possible to reformulate the basis of Bayesian statistics to handle improper priors in a rigorous manner (Hartigan, 1983) albeit at the expense of considerable technicalities.

Of course if A is finite then simply rescaling G_w provides a direct analogy of the development given above. However if A is infinite then a different argument is needed. If one wanted to compare the one event $\Omega \subseteq \mathbb{R}^N$ with respect to a uniform distribution (an improper distribution on $\mathbb{R}^N$) or a non-uniform distribution ϕ then one could look at the ratio

$$R_{(\phi,1)}(\Omega) := \frac{\int_\Omega \phi(w) dw}{\int_\Omega 1 dw}.$$

Thus, $R_{(\phi_1,\phi_2)}(\Omega)$ is a ratio of likelihoods rather than an absolute probability. As long as the events considered always have finite non-zero weight with respect to the set Ω, then the ratio $R_{(\phi_1,\phi_2)}(\Omega)$ is well defined. We will call distributions ϕ of the form (16) *relative probability distributions.* Such a distribution should always be thought of relative to a second distribution; in the sequel unless otherwise mentioned the relative distribution will always be the uniform distribution. Thus, in this context the ratio of $\phi(w)/1$ can be thought of as the relative likelihood of the point w compared to the uniform distribution. We shall see in subsection 7.1 that rescaling $\phi(w)$ by multiplication by a scalar is equivalent to a simple change in the step size used.

The machinery associated with improper distributions and relative probability distributions is natural in the analysis undertaken in this paper since the basic GD learning algorithm can be thought of a being associated with the uniform prior distribution. This can be seen heuristically by observing that the direction of update for the GD algorithm is totally unbiased by any modification of the direct steepest descent direction. In a probabilistic sense this is saying that $w_* \in \mathbb{R}^N$ is equally likely to be any point in $\mathbb{R}^N$ and hence the best estimate of w_* is based on minimizing the loss associated with the latest received data. By contrast, the EG algorithm update step is approximately taken in the direction of steepest descent for very small update steps, but the larger the update step then the more the distortion of the exponential tends to change the next estimate. We interpret this as taking account of prior information in the update step. (An explicit form for the prior associated with the EG algorithm is determined in section 6.2).

In the sequel we will always be dealing with the performance of a given algorithm *relative to* the performance of some other algorithm. We will relate this relative performance back to relative probability densities and then to certain associated Riemannian metrics. Since the entire analysis is relative we will choose to adopt the following premise:[2] *The GD algorithm is the "best" or most approrpriate learning algorithm given that one has no prior knowledge of the true parameter. The GD algorithm is associated with the uniform improper prior and the Euclidean preferential metric.*

5. Learning Algorithms on Preferentially Structured Parameter Space

In this section a method for generating learning algorithms is proposed based on a given preferential structure.

Both the GD and EG learning algorithms (5) and (6) are stochastic gradient descent algorithms and use only first order differential information of $\mathcal{L}$ and a step size (learning rate) s_k at each update. Consider a general learning algorithm of the form

$$w_{k+1} = F(s_k, \frac{\partial \mathcal{L}}{\partial w}, w_k), \tag{17}$$

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 locally 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" $\gamma_{(w_k,V_k)}(\cdot)$ 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 maximise the distance

$$\delta(w_k, w_{k+1})$$

taken at every step (cf. (8)) measured relative to the preferential structure!

[2] The term "best" in this definition is certainly dependent on the context of what constitutes a learning algorithm. Certainly if one were to consider simply the optimization problem of minimizing the loss over all target parameters then some more sophisticated update method (for example a Newton update) would tend to display better performance. Here we will restrict 'learning algorithms' to be effectively the class of algorithms which we generate in the sequel. Though this may appear to be a circular definition it is effectively the set of linear descent algorithms with respect to general Riemannian geometry. Only direct first order derivatives of the cost are used to generate the update information. Further modification of the update step is entirely due to prior information. The key point is not which sense GD is best in, but rather how we can modify GD simply taking it as the starting point — the appropriate algorithm for a uniform prior.

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 the sequence $\{w_k\}$ will converge to w_* for reasonable data samples) then the curve γ should be chosen to maximise 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:

$$\sqrt{\langle \dot\gamma(\tau), \dot\gamma(\tau) \rangle_{\gamma(\tau)}} = \langle V_k, V_k \rangle_{w_k}, \quad \tau \geq 0. \tag{18}$$

By this argument, the 'best' curve to choose for the purpose of generating a learning algorithm is one that satisfies (18) whilst maximising $\delta(w_k, w_{k+1})$ 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.

5.1 Geodesics

Geodesic curves on a Riemannian manifold can be defined as the solution of a ODE (Ordinary Differential Equation) in local co-ordinates which essentially ensures straightness of the solution curve with respect to the metric (Lee, 1997, pg. 68).

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

$$\Gamma^k_{ij} = \frac{1}{2} \sum_{s=1}^{N} g^{ks} \left(\frac{\partial g_{si}}{\partial w^j} - \frac{\partial g_{ij}}{\partial w^s} + \frac{\partial g_{js}}{\partial w^i} \right). \tag{19}$$

The functions Γ^k_{ij} are known as the Christoffel symbols (Boothby, 1986, Lee, 1997). A geodesic curve $\gamma := \gamma_{(w_k, V_k)}(\tau)$ satisfies the set of coupled second order ODEs (Lee, 1997, pg. 58)

$$\frac{d^2\gamma^k}{d\tau^2} + \sum_{i,j=1}^{N} \Gamma^k_{ij} \frac{d\gamma^i}{d\tau} \frac{d\gamma^j}{d\tau} = 0, \tag{20}$$

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 $\mathbb{R}^N$. Rather, $D_w\mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial w}\right)^T$ is the differential of $\mathcal{L}$ and is a row vector (co-tangent vector) and not a column vector (tangent vector). 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

$$\mathrm{grad}\mathcal{L} = G_w^{-1}\frac{\partial \mathcal{L}}{\partial w}. \tag{21}$$

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 (1997, 1998) 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 negative 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 $Q > 0$

$$V_k = -Q^{-1}\frac{\partial \mathcal{L}}{\partial w},$$

generates a descent direction. The general form of the learning algorithms studied in the remainder of the paper may now be 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 studied is given by

$$\boxed{w_{k+1} = \gamma_{(w_k, -\mathrm{grad}\mathcal{L})}(s_k),} \tag{22}$$

where $\gamma_{(w_k, -\mathrm{grad}\mathcal{L})}$ is a geodesic curve with respect to the preferential structure (confer (20)).

6. Preferential Structures for Two Common Learning Algorithms

In this section some common learning algorithms are analyzed in terms of preferential structures. The gradient descent (GD) algorithm has an interpretation based on the standard Euclidean (or unbiased) preferential structure on $\mathbb{R}^n$ while the exponential gradient algorithm is related to a preferential metric of the form (13).

6.1 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

$$\frac{d^2\gamma^k}{d\tau^2} = 0, \tag{23}$$

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 grad$\mathcal{L} = \frac{\partial \mathcal{L}}{\partial w}$ Thus, comparing with (5) it is easily verified that the GD algorithm is simply

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

6.2 Exponentiated Gradient Algorithm

In this subsection the EG (Exponentiated Gradient) algorithm is derived within the framework of preferential structures.

Previous work (Kivinen and Warmuth, 1997, Hill and Williamson, 2001) has shown that the EG algorithm tends to perform well in the situation where only a few entries of the true parameter w_* are non-zero. Considering the question in reverse one would heuristically wish to choose a preferential structure that emphasises regions where only a few co-ordinates are non-zero. We will choose such a structure and show the EG algorithm (almost) follows from such a choice.

Let $\mathbb{R}^N_*$ denote the positive cone in $\mathbb{R}^N$

$$\mathbb{R}^N_* = \{u \in \mathbb{R}^N : u > 0\}. \tag{24}$$

Consider the following preferential metric defined on $\mathbb{R}^N_*$

$$G_w = \begin{pmatrix} \frac{1}{(w^1)^2} & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \frac{1}{(w^N)^2} \end{pmatrix} \tag{25}$$

According to Section 4.2 this metric is associated with an improper product distribution on $\mathbb{R}^N_*$ of the form

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

where each $\phi_i(w)$ is given by

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

Thus, each component prior distribution is weighted more heavily near $w^i = 0$ and thus the overall product density is weighted strongly near $w = 0$. (Some graphs illustrating this metric are given in Section A.)

The diagonal structure of the preferential metric G_w makes it particularly easy to compute the Christoffel symbols. Recalling (19) the Christoffel symbols are

$$\Gamma^p_{ij} = \begin{cases} \frac{-1}{w^p} & \text{if } i = j = p. \\ 0 & \text{otherwise.} \end{cases}$$

Finally, recalling (20) the equation for the geodesic is

$$\frac{d^2\gamma^p}{d\tau^2} - \frac{1}{\gamma^p}\left(\frac{d\gamma^p}{d\tau}\right)^2 = 0, \quad \text{for } p = 1, \dots, N, \tag{26}$$

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. It is easily verified by substitution that the solution of this equation for each co-ordinate is

$$\gamma^i(\tau) = w^i \exp\left(\frac{\tau}{w^i} V^i\right). \tag{27}$$

Consider the descent direction

$$V_k = -\mathrm{diag}(w_k)\frac{\partial \mathcal{L}}{\partial w}(y_k, \hat{y}_k) = -G_w^{-1/2}\frac{\partial \mathcal{L}}{\partial w}. \tag{28}$$

where w_k denotes the kth iteration of the learning algorithm. Note that $V_k \neq -\mathrm{grad}\mathcal{L}_k$ with respect to the preferential structure G_w, however, V_k 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 (22) leads to the EG algorithm (6)

$$w_{k+1} = \gamma_{(w_k, -\mathrm{diag}(w_k)\frac{\partial \mathcal{L}}{\partial w})}(s_k) = \mathrm{diag}(w_k)\exp.\left(-s_k\frac{\partial \mathcal{L}}{\partial w}(y_k, \hat{y}_k)\right) = (6).$$

Although V_k is not actually the negative 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

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

This choice results in a the learning algorithm that we will call the ‘natural EG algorithm’

$$\boxed{w_{k+1} = \gamma_{(w_k, -\mathrm{grad}\mathcal{L})}(s_k) = \mathrm{diag}(w_k)\exp.\left(-s_k\mathrm{diag}(w_k)\frac{\partial \mathcal{L}}{\partial w}(y_k, \hat{y}_k)\right).} \tag{29}$$

This raises the question: what is the effective difference between (6) and (29), or is one to be preferred somehow. One remark we can offer in this regard is that (29) might be preferred in so far as it allows a comparison with other gradient algorithms which use the natural gradient and thus differ *only* in the choice of ϕ. It is also apparent by inspecting the two algorithms and their behaviour in simulation that (29) pays more attention to the prior.

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 (Kivinen and Warmuth, 1998) to analyse the EG algorithm can be obtained as a direct generalization of normal co-ordinates with respect to the preferential structure (25).

Consider a product preferential metric of the form

$$G_w = \begin{pmatrix} \phi_1(w^1)^2 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \phi_N(w^N)^2 \end{pmatrix}, \tag{30}$$

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 situation where G_w is a preferential metric associated with a product prior

$$\phi(w) = \prod_{i=1}^{N} \phi_i(w^i).$$

This structure has some special consequences for the geometry of learning algorithms derived according to the procedure outlined in Section 5. To distinguish between the preferential geometry and the classical Euclidean geometry on $\mathbb{R}^N$ we denote the preferentially structured space by $\tilde{\mathbb{R}}^N$.

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 is inferred from context. Thus, $g_{ij} = 0 = g^{ij}$ except when $i = j$ and

$$g_{ii} = \phi_i(w^i)^2, \quad g^{ii} = \frac{1}{\phi_i(w^i)^2}.$$

Recalling (19) the Christoffel symbols are easily verified to be

$$\Gamma^p_{ij} = \begin{cases} \frac{1}{\phi_i(w^i)} \frac{d\phi_i}{dw^i}(w^i) & \text{if } i = j = p. \\ 0 & \text{otherwise.} \end{cases} \tag{31}$$

Thus the Christoffel symbols for a product preferential metric always have a diagonal structure. Recalling (20) the general equation for a geodesic curve, then it is easily shown that the general geodesic equation in this case is a set of N *decoupled* second order ODEs

$$\ddot{\gamma}^i + \frac{1}{\phi_i(\gamma^i)} \frac{d\phi_i(\gamma^i)}{d\gamma^i} \left(\dot{\gamma}^i\right)^2 = 0. \tag{32}$$

As a consequence the general geodesics are made up of independent evolution equations in each of the co-ordinates.

Equation 32 can be simplified to obtain a set of simple first order, single variable ODEs. Note that

$$\frac{d\phi_i(\gamma^i)}{d\gamma^i}\dot{\gamma}^i = \frac{d\phi_i(\gamma^i)}{dt} =: \dot{\phi}_i$$

where we now think of $\phi_i(t) = \phi_i(\gamma^i(t))$ as a function of t. Thus, the above equation for the geodesic becomes

$$\ddot{\gamma}^i + \frac{\dot{\phi}^i}{\phi_i}\dot{\gamma}^i = 0.$$

Let $z^i = \phi_i\dot{\gamma}^i$ (i.e. $z^i(t) = \phi_i(\gamma^i(t))\dot{\gamma}^i(t)$). Then

$$\dot{z}^i = \dot{\phi}_i\dot{\gamma} + \phi_i\ddot{\gamma}^i = \phi_i\left(\frac{\dot{\phi}^i}{\phi_i}\dot{\gamma}^i + \ddot{\gamma}^i\right) = 0,$$

where the factorization by ϕ_i is always possible since the ϕ_i are positive definite functions. Substituting back into the definition of z^i and rearranging terms yields the ODE

$$\begin{aligned} \dot{\gamma}^i &= \frac{z^i(0)}{\phi_i(\gamma^i)} && (33) \\ z^i(0) &= \phi_i(\gamma^k(0))\dot{\gamma}^i(0); \qquad \gamma^i(0) = w^i; \qquad \dot{\gamma}^i(0) = V^i && (34) \end{aligned}$$

Lemma 5 *Suppose $\phi_i(w) > 0$ for all $w \in \mathbb{R}$. Then the solution of (33,34) is*

$$\gamma^i(t) = \Phi_i^{-1}(tV^i\phi_i(w^i) + \Phi_i(w^i)) \tag{35}$$

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

Proof We write $\phi = \phi_i$ for simplicity. The condition on ϕ implies Φ is invertible and thus (35) implies

$$\Phi(\gamma(t)) = tV\phi(w) + \Phi(w).$$

Differentiating both sides we obtain

$$\begin{aligned} \frac{\partial}{\partial t}\Phi(\gamma(t)) &= \frac{\partial}{\partial t}(tV\phi(w) + \Phi(w)) && (36) \\ \Rightarrow \quad \phi(\gamma(t))\dot{\gamma}(t) &= V\phi(w) && (37) \\ \Rightarrow \quad \dot{\gamma}(t) &= \frac{V\phi(w)}{\phi(\gamma(t))} = \frac{\phi(\gamma(0))\dot{\gamma}(0)}{\phi(\gamma(t))}. && (38) \end{aligned}$$

Furthermore $\gamma(0) = \Phi^{-1}(\Phi(w)) = w$ and $\dot{\gamma}(0) = \frac{V\phi(w)}{\phi(\gamma(0))} = V$. ■

One can readily check (by replacing $\Phi(w)$ by $\Phi(w)+c$ and $\Phi^{-1}(x)$ by $\Phi^{-1}(x-c)$) that the constant of integration c effectively omitted in the definition of Φ does not change the solution $\gamma(t)$.

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

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

(22) takes the general form

$$w_{k+1}^i = \Phi_i^{-1}\left(-s_k \left.\frac{\partial \mathfrak{L}}{\partial w^i}\right|_{w^i=w_k^i} \frac{1}{\phi_i(w_k^i)} + \Phi_i(w_k^i)\right) \tag{39}$$

When $\mathfrak{L}$ is the squared loss (4), $\frac{\partial \mathfrak{L}}{\partial w^i} = -x_k^i(y_k - \hat{y}_k)$ one obtains

$$\boxed{w_{k+1}^i = \Phi_i^{-1}\left(\frac{s_k x_k^i(y_k - \hat{y}_k)}{\phi_i(w_k^i)} + \Phi_i(w_k^i)\right)} \tag{40}$$

7.1 Rescaling of $\phi(w)$

As a sanity check, now consider what happens if we replace $\phi(w)$ by

$$\tilde{\phi}(w) := \beta\phi(w) \tag{41}$$

for some $\beta > 0$. Clearly we have

$$\tilde{\Phi}(w) = \beta\Phi(w) \tag{42}$$

and

$$\tilde{\Phi}^{-1}(x) = \Phi^{-1}(x/\beta) \tag{43}$$

Substituting (41), (42) and (43) into (40) we see that we recover the orginial algorithm by setting $\tilde{s}_k = \beta^2 s_k$. This makes sense when one recalls (see (30)): clearly $\tilde{G}_w = \beta^2 G_w$.

This simple analysis shows there is no intrinsic reason to insist that the distributions $\phi(w)$ are normalized as proper probability distributions.

8. Examples of Possible New Learning Algorithms

In this section we examine some examples of learning algorithms generated by certain specific preferential structures. In all cases the underlying learning problem is that presented in Section 5 (cf. (22)). The algorithms are in fact simply (40) for different choices of $\phi(w)$. The resulting ϕ, Φ and Φ^{-1} are collected together in table 1.

Algorithm	$\phi(w)$	Conditions	$\Phi(w)$	$\Phi^{-1}(x)$
EGnatural	$\frac{1}{w}$	$w > 0$	$\ln(w)$	e^x
EG(α)	$\frac{1}{w^\alpha}$	$w > 0$ $\alpha \neq 1$	$\frac{1}{(1-\alpha)} \frac{1}{w^{\alpha-1}}$	$\frac{1}{(1-\alpha)} \frac{1}{x^{1/(\alpha-1)}}$
EGclipped(c)	$\min\left(\frac{1}{\lvert w\rvert}, c\right)$	$c > 0$	$\begin{cases} cw, & \lvert w\rvert \leq \frac{1}{c} \\ \operatorname{sgn}(w)(1+\ln(c\lvert w\rvert)), & \lvert w\rvert > \frac{1}{c} \end{cases}$	$\begin{cases} \frac{x}{c}, & \lvert x\rvert \leq 1 \\ \frac{\operatorname{sgn}(x)e^{\lvert x\rvert-1}}{c}, & \lvert x\rvert \leq 1 \end{cases}$
Cauchy$^{1/2}$	$\frac{1}{\sqrt{1+w^2}}$		$\operatorname{arcsinh}(w)$	$\sinh(x)$
Cauchy	$\frac{1}{1+w^2}$		$\arctan(w)$	$\tan(x)$
Cauchy$^{3/2}$	$\frac{1}{(1+w^2)^{3/2}}$		$\frac{w}{\sqrt{1+w^2}}$	$\sqrt{\frac{x^2}{1-x^2}}$
exp(α)	$e^{-\alpha\lvert w\rvert}$	$\alpha > 0$	$\frac{\operatorname{sgn}(w)}{\alpha}\left(1-e^{-\alpha\lvert w\rvert}\right)$	$-\frac{\operatorname{sgn}(x)}{\alpha}\ln(1-\alpha\lvert x\rvert)$
Gaussian	$e^{-\alpha^2 w^2}$	$\alpha > 0$	$\frac{\sqrt{\pi}\operatorname{erf}(\alpha w)}{2\alpha}$	$\frac{\operatorname{erf}^{-1}(2\alpha x/\sqrt{\pi})}{\alpha}$

Table 1: Some possible choices of ϕ, Φ and Φ^{-1} for algorithm (40).

8.1 EG (natural)

Choose $\phi(w) = 1/w$ and thus from (40) and Table 1 one obtains the algorithm:

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

which is equivalent to (29) with $\mathfrak{L}$ being squared loss.

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 ± trick as presented by Kivinen and Warmuth (1997) could be used.

8.2 EG(α)

Choose $\phi(w) = 1/w^\alpha$ ($\alpha \neq 1$) and thus from (40) and Table 1 one obtains the algorithm:

$$w_{k+1}^i = \frac{1}{1-\alpha}\left(s_k x_k^i (y_k - \hat{y}_k)(w_k^i)^\alpha + \frac{1}{1-\alpha}(w_k^i)^{\alpha-1}\right)^{\frac{1}{1-\alpha}} \tag{44}$$

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

Numerically implementing the algorithm it turns out that weights can tunnel through the infinite barrier at the origin due to the non-infintesimal step size used. In order to avoid problems which this causes (more likely for larger values of s), we have found it necessary to modify the algorithm to

$$\begin{aligned} t_k &= \operatorname{sgn}(w_k^i)|w_k^i|^{1-\alpha} + (|w_k^i| + \epsilon)^\alpha x_k^i s_k (y_k - \hat{y}_k) \\ w_{k+1}^i &= \operatorname{sgn}(t_k)|t_k|^{1/(1-\alpha)}. \end{aligned}$$

Here ϵ is some small number; we have used $\epsilon = 2 \times 10^{-13}$.

8.3 EGclipped(c)

In order to avoid the difficulties of the singularity at the origin with ϕ_{EG}, one can simply clip ϕ to $c > 0$ giving

$$\phi_c(w) = \min\left(c, \frac{1}{|w|}\right). \tag{45}$$

One can show that

$$\Phi_c(w) = \begin{cases} cw & |w| \leq \frac{1}{c} \\ \operatorname{sgn}(w)(1 + \ln(c|w|)) & |w| > \frac{1}{c}. \end{cases} \tag{46}$$

$$\Phi_c^{-1}(w) = \begin{cases} \dfrac{x}{c} & |x| \leq 1 \\ \frac{\operatorname{sgn}(x)e^{|x|-1}}{c} & |x| > 1 \end{cases} \tag{47}$$

Observe that on a bounded domain, for all $c < \infty$, $\phi_c(w)$ can be made into a proper distribution (by rescaling). The limit as $c \to \infty$ is improper though. This is analogous to how improper priors are sometimes treated in the Bayesian literature, as the limit of a sequence of proper priors (Akaike, 1980).

8.4 exp(α)

Choose $\phi_\alpha(w) = \exp(-\alpha|w|)$, $\alpha > 0$. By considering positive and negative cases separately one show

$$\Phi_\alpha(w) = \frac{\text{sgn}(w)}{\alpha} \ln\left(1 - e^{-\alpha|w|}\right) \tag{48}$$

$$\Phi_\alpha^{-1}(x) = \frac{-\text{sgn}(x)}{\alpha} \ln\left(1 - \alpha|x|\right), \qquad |x| < \frac{1}{\alpha} \tag{49}$$

8.5 Cauchy Product Distribution

Choose

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

which is the Cauchy distribution (unnormalized). The variance of the Cauchy distribution is not defined and it is a classic example of a distribution with heavy tails. 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 $\mathbb{R}^N$ and there is no need to use ± algorithms like those developed for the EG algorithm (Kivinen and Warmuth, 1997).

From (40) and Table 1 we obtain the algorithm:

$$w_{k+1}^i = \tan\left(s_k x_k^i (y_k - \hat{y}_k)(1 + (w_k^i)^2) + \arctan(w_k^i)\right). \tag{50}$$

8.6 Elementwise Link Functions and Flat Preferential Structures

Let $E_1, E_2, \ldots, E_N$ be the unit vectors in $\mathbb{R}^N$. Let $\gamma_{(w,V)}(s)$ denote the particular geodesic curve obtained as a solution of (33) for initial conditions $\gamma_{(w,V)}(0) = w$ and $\dot{\gamma}_{(w,V)}(0) = V$. Then one can define a map

$$f : \mathbb{R}^N \to \tilde{\mathbb{R}}^N$$
$$f(x) := \gamma_{\left(w, \sum_{i=1}^N x^i E_i\right)}(1)$$

The map $f(x)$ provides a set of local co-ordinates for $\tilde{\mathbb{R}}^N$ known as normal co-ordinates (Boothby, 1986). As we show below the map f is closely related to the concept of link functions commonly used to analyse learning algorithms such as the EG algorithm. The local co-ordinate frames induced are denoted

$$\frac{\partial}{\partial x^i} := df|_x(E_i), \quad i = 1, \ldots, N,$$

For any two vector fields X, Y on $\tilde{\mathbb{R}}^N$ then let $\nabla_X Y$ denote the action of the Levi-Civita connection of X on Y (Boothby, 1986, pp. 317). Since f is constructed from solutions to the geodesic equations then it follows directly that

$$\nabla_{\frac{\partial}{\partial x^i}} \frac{\partial}{\partial x^i} = 0, \quad i = 1, \ldots, N.$$

In fact, using the structure of Γ^k_{ij} it is easily verified that

$$\nabla_{\frac{\partial}{\partial x^i}} \frac{\partial}{\partial x^j} = 0. \quad i, j = 1, \dots, N.$$

This property is not true of general Riemannian manifolds and is important since it implies that the underlying geometry of $\tilde{\mathbb{R}}^N$ is effectively Euclidean. In particular, taking a straight line $x_k + s_k V_k$ on $\mathbb{R}^N$ which is a geodesic in the Euclidean geometry then f maps this to a geodesic

$$\gamma(s) = f(x_k + s_k V_k)$$

on $\tilde{\mathbb{R}}^N$. The fact that the base point x_k need not be $\mathbf{0}$ is crucial in this relationship since it implies that the structure is unchanged for translations in $\mathbb{R}^N$. This will only occur if the intrinsic 'curvature' of $\tilde{\mathbb{R}}^N$ is zero.

A further consequence of the flatness of $\tilde{\mathbb{R}}^N$ is that the normal co-ordinate mapping f is an isometry. That is that for any two vectors $X, Y \in T_x\mathbb{R}^N$ then

$$X^T Y = \langle X, Y \rangle = \langle df X, df Y \rangle_{f(x)} = (df X)^T G_{f(x)} df Y.$$

Conseqeuntly, the mapping f preserves the metric distance given by length of curves. Since f is an isometry one may as well take a standard learning algorithm on the simple Euclidean space $\tilde{\mathbb{R}}^N$ given in local Euclidean co-ordinates $x \in \mathbb{R}^N$ (cf. Subsection 6.1)

$$x_{k+1} = x_k + s_k \tilde{V}_k \tag{51}$$

and obtain its associated learning algorithm on the desired space directly by the mapping $w_k = f(x_k)$. Of course the descent direction $\tilde{V}_k$ is the descent direction in local co-ordinates x and must map via df to a chosen descent direction $V_k \in T_{f(x_k)}\tilde{\mathbb{R}}^N$. Thus,

$$df(\tilde{V}_k) := V_k.$$

The function f is serving the same role as the link functions commonly used to analyse and extend the EG algorithm (Kivinen and Warmuth, 1998).

For example consider the EG algorithm where $f = \exp.$, the vector exponential, is the standard link function used. Direct computations show $df = \operatorname{diag}(w_k) \exp.(X) = w$ and hence $\tilde{V}_k = \operatorname{diag}(w)^{-1} V_k$. Choosing $V_k = -G_w^{-\frac{1}{2}} \frac{\partial \mathfrak{L}}{\partial w}$ according to (28) then it follows that $\tilde{V}_k = -\frac{\partial \mathfrak{L}}{\partial w}$. The learning algorithm obtained on $\tilde{\mathbb{R}}^N = \mathbb{R}^N_*$ from (51) is

$$\begin{aligned} w_{k+1} &= f(x_{k+1}) = f(x_k + s_k \tilde{V}_k) \\ &= \exp.\left(x_k + s_k \tilde{V}_k\right) = \operatorname{diag}(\exp.(x_k)) \exp.(s_k \tilde{V}_k) \\ &= \operatorname{diag}(w_k) \exp.(-s_k \frac{\partial \mathfrak{L}}{\partial w}). \end{aligned}$$

Comparing with (6) it is seen that this is another to way to derive the EG algorithm.

Remark 6 *In some recent literature (Warmuth and Jagota, 1997) the concept of link function has been developed in a co-ordinate by co-ordinate context. Thus, a link function is considered to be a map*

$$f^i : \mathbb{R} \to \mathbb{R},$$

where the same function $f^i := g$ is generally used for each co-ordinate. The development here is closely related since the product preferential structure ensures that the normal co-ordinates can always be decomposed into independent co-ordinate functions $f^i := f^i(x^i)$ obtained as solutions to the decoupled geodesic equations (33). The unified view presented in this paper provides a way of generalizing the link function results to obtain a better understanding of learning algorithms.

Remark 7 *Gordon (1999a, section 3.9.2) has also presented a (quite different) interpretation of algorithms such as GD and EG via Bayesian priors. He makes use of* conjugate priors *(see Robert, 1994) and his framework is rather different to ours. We have been unable to draw a formal connection between his work and ours. Furthermore, the machinery of conjugate priors is intrinsically restricted — it can not cover the range of prior distributions we can deal with. Whether there is a useful connection is left as an open problem.*

9. Comparing the Performance of Stochastic Descent Learning Algorithms.

In this section the relative performance of two stochastic gradient descent learning algorithms derived with respect to diagonal preferential structure (associated with product distributions) are compared. The algorithms are compared for a target weight vector w^* drawn from the true product prior distribution that we assume is associated with one of the algorithms.

The relative performance of two learning algorithms must be compared according to some criterion that is both computable and relevant to the desired qualitative behaviour of the algorithms. Kivinen and Warmuth (1997) proposed the 'mistake-bounded framework' for analysing the EG algorithm. A more traditional analysis based on classical LMS analysis techniques has also been considered for the EG algorithm (Hill and Williamson, 1999, 2001). Both approaches attempt to quantify the global performance of the algorithm considered. In this paper generic algorithms based on the stochastic gradient descent concept are considered. A stochastic gradient descent algorithm may be considered to be a sequence of Bayesian estimates of the true weight vector conditioned on a local neighbourhood of the previous estimate and based on the most recent data. That is, at each step one searches for the parameter $w_{k+1} \in B_r(w_k)$ that best describes the latest data received. The learning rate s_k of a stochastic gradient algorithm is linked directly to the radius r of the ball which is used to condition the new estimate. A consequence of this perspective is that one should only seek to compare the *relative local* performance of stochastic gradient descent algorithms. It is to be hoped, however, that for most sensible classes of prior distributions good local performance will translate into good global performance. This justification, along with the practical simplicity and ease of implementation, motivates the use of stochastic gradient descent algorithms in learning applications. Conversely, if one wishes to solve the problem optimally then it is necessary to return to a full Bayesian analysis.[3]

All stochastic gradient descent algorithms (for sufficiently small learning rate) display the same qualitative asymptotic convergence properties (Solo and Kong, 1995). The asymptotic error may differ depending on the relative sensitivity of the schemes and the particular learning rate used. High asymptotic sensitivity is usually linked to a fast learning rate and better transient performance of an algorithm. As a consequence asymptotic error analysis provides a poor measure of relative performance of two stochastic gradient descent learning algorithms.

A tempting comparison of the transient performance between two learning algorithms is to take the expectation (with respect to the given prior distribution) of the rate of decrease of the loss function for the two algorithms across all possible weight vectors and all target vectors. Unfortunately, due to the local nature of the Bayesian interpretation of stochastic gradient algorithms a global average will not provide a suitable comparison, although, in certain specific cases the global average may well indicate the desired result. *In this paper we will compare the expected decrease of the loss function over possible measurements and true weight vectors locally around each point in weight space.*

Assume that samples x_k are drawn from a normalized uniform iid distribution with density ϕ and that the measurements y_k are deterministic functions of x_k. The expected value of the loss function over possible samples x_k is

$$\begin{aligned}\psi(w) &:= E^{\phi}_{x_k}\left[\mathcal{L}(y_k, \hat{y}_k)\right] \\ &= \frac{1}{2}\int_{\mathbb{R}^N} (y_k - \hat{y}_k)^2 \phi(x_k) dx_k \\ &= \frac{1}{2}\int_{\mathbb{R}^N} \langle w - w_*, x_k\rangle^2 \phi(x_k) dx_k \\ &= \frac{1}{2}(w - w_*)^T E^{\phi}_{x_k}[x_k x_k^T](w - w_*).\end{aligned} \tag{52}$$

Since ϕ is normalized uniform iid then

$$E^{\phi}_{x_k}[x_k x_k^T] = I_N$$

where I_N is the $N \times N$ identity matrix.

Let p and q be two proper product prior distributions on a compact subset $\Omega \subset \mathbb{R}^N$ of weight space

$$p(w) = \prod_{i=1}^{N} \phi_p^i(w), \quad q(w) = \prod_{i=1}^{N} \phi_q^i(w). \tag{53}$$

Let $G_p(w) := \operatorname{diag}((\phi_p^1)^2, \ldots, (\phi_p^N)^2)$ and $G_q(w) := \operatorname{diag}((\phi_q^1)^2, \ldots, (\phi_q^N)^2)$ be diagonal preferential structures associated with the product distributions p and q.

We will compare the behaviour of the averaged learning algorithms

$$w_{k+1} = \gamma_{\left(w_k, -G_\dagger(w)^{-1}(w_k - w_*)\right)}(s_\dagger^k), \tag{54}$$

[3] Even if the reader does not completely buy the above argument, we have a backup: "we have performed a local analysis because a global analysis seems dauntingly difficult!".

derived from Eq. 22 for the metric $G_\dagger(w)$ equal to either $G_p(w)$ or $G_q(w)$. The step-size $s^k_\dagger$ is either s^k_p or s^k_q depending on the algorithm. The step-size scaling factor is chosen to scale the relative volume induced by each probability distribution to equal a constant

$$\frac{(s^k_p)^N}{p(w)} = \frac{(s^k_q)^N}{q(w)} = \frac{\mathrm{vol}B_r(w_k)}{\mathrm{vol}\Omega} = r^N\frac{\mathrm{vol}B_1(w_k)}{\mathrm{vol}\Omega} \tag{55}$$

where $\mathrm{vol}B_r(0)$ denotes the (Euclidean) volume of a ball of radius $r > 0$. For r sufficiently small the probability of the event $B^\dagger_{s^k_\dagger}(w_k)$ (denoting the ball with respect to the preferential structure †) under the relevant prior distribution is normalized

$$\begin{aligned}P_p(B^p_{s^k_p}(w_k)) &= \int_{B^p_{s^k_p}(w_k)} p(z)dz \approx \prod_{i=1}^N \frac{s^k_p}{(\phi^i_p)^2}p(w_k) = \frac{(s^k_p)^N}{p(w_k)}\\ &= \frac{\mathrm{vol}B_r(0)}{\mathrm{vol}\Omega} = \frac{(s^k_q)^N}{q(w)} \approx P_q(B^q_{s^k_q}(w_k)).\end{aligned}$$

Thus, the step-size scaling factor is adjusted so that, based on the prior information under which an algorithm is derived, there is a fixed probability that the true weight vector lies in the set in which the next update of the algorithm is chosen. The scaling factor r which is associated with the uniform distribution on the compact set Ω provides a useful 'knob' with which to tune the performance of a class of learning algorithms. In the simulations (cf. Section 10) we compare two new algorithms with standard Gradient Descent (which has $\phi(w) = 1$) and thus choose $s^k_{\mathrm{new}} = s^k_{\mathrm{GD}}(\phi_{\mathrm{new}}(w))^{1/N}$.

The following approximation of (54) holds for sufficiently small step-size $s^k_\dagger$

$$w_{k+1} = w_k - s^k_\dagger G_\dagger(w_k)^{-1}(w_k - w_*) + \mathbf{O}\left((s^k_\dagger)^2||G_\dagger||^2_2\right),$$

To bound the error due to the approximation it is necessary to reduce the scalar r that bounds maximum step length. Note that $s^k_\dagger \propto r$ (cf. Eq. 55) and the above equation may be written

$$w_{k+1} = w_k - s^k_\dagger G_\dagger(w_k)^{-1}(w_k - w_*) + \mathbf{O}\left(r^2\right), \tag{56}$$

where the constant associated with the $\mathbf{O}(r^2)$ term scales as $\sup_{w\in\Omega}||G^{-1}_\dagger(w)||^2_2$.

Let

$$\Delta\psi_\dagger(w_k) := \psi_\dagger(w_{k+1}) - \psi_\dagger(w_k) \tag{57}$$

denote the decrease of the averaged loss function at step $k+1$ with respect to the preferential structure $G_\dagger$. From (56) we obtain

$$\begin{aligned}\Delta\psi_\dagger(w_k) &= \frac{1}{2}\left(w_k - s^k_\dagger G_\dagger(w)^{-1}(w_k - w_*) - w_*\right)^2 - \frac{1}{2}(w_k - w_*)^2 + \mathbf{O}(r^2)\\ &= -s^k_\dagger(w_k - w_*)^T G_\dagger(w)^{-1}(w_k - w_*) + \mathbf{O}(r^2).\end{aligned}$$

Since $G_\dagger(w) = \mathrm{diag}\{(\phi^i_\dagger)^2\}$ is a diagonal preferential structure then we can write

$$\Delta\psi_\dagger(w_k) = -s^k_\dagger\sum_{i=1}^N\frac{(w^i - w^i_*)^2}{(\phi^i_\dagger)^2} + \mathbf{O}(r^2) \tag{58}$$

Theorem 8 *Consider the learning problem outlined in Section 2 restricted to a compact subset $\Omega \subset \mathbb{R}^n$. Assume that the samples are chosen according to a normalized uniform iid process and let ψ be the averaged loss function (52). Let p and q be two product prior distributions (Eqn's 53) for the true weight vector w_* with associated diagonal preferential structures $G_p(w)$ and $G_q(w)$. Assume that $\sup_{w\in\Omega} ||G_p^{-1}(w)||_2$ and $\sup_{w\in\Omega} ||G_q^{-1}(w)||_2$ are bounded from above. Let $\Delta\psi_\dagger$ be defined by (57). For any point $w_k \in \Omega$ set*

$$\Sigma_p(w_k) = \operatorname{diag}\left(\int_\Omega (w_k^i - w_*^i)^2 p(w_*) dw_*\right)$$

If for all $w_k \in \Omega$

$$p(w_k) < q(w_k) \tag{59}$$

and

$$\frac{1}{N}\operatorname{tr}\left(G_q^{-1}(w_k)\Sigma_p(w_k)\right) \leq \det\left(G_q^{-1}(w_k)\Sigma_p(w_k)\right)^{\frac{1}{N}} \tag{60}$$

then there exists $r > 0$ sufficiently small such that when s_p^k and s_q^k are chosen to satisfy (55) one has

$$E_{w_*}^p\left[\Delta\psi_p(w_k)\right] < E_{w_*}^p\left[\Delta\psi_q(w_k)\right] < 0. \tag{61}$$

Proof For each $w_k \in \Omega$ there exists an $r_1(w_k) > 0$ sufficiently small such that both $E_{w_*}^p\left[\Delta\psi_p(w_k)\right]$ and $E_{w_*}^p\left[\Delta\psi_q(w_k)\right]$ are negative. Since Ω is compact there exists $r_1 := \inf_{w\in\Omega}\{r_1(w)\} > 0$. Let

$$\begin{aligned}
F &:= E_{w_*}^p\left[\Delta\psi_p(w_k)\right] - E_{w_*}^p\left[\Delta\psi_q(w_k)\right] \\
&= -\int_\Omega s_p^k \sum_{i=1}^N \frac{(w_k^i - w_*^i)^2}{(\phi_p^i(w_k))^2} p(w_*) dw_* + \int_\Omega s_q^k \sum_{i=1}^N \frac{(w_k^i - w_*^i)^2}{(\phi_q^i(w_k))^2} p(w_*) dw_* + \mathbf{O}(r^2) \\
&= \sum_{i=1}^N \left(\frac{s_q^k}{(\phi_q^i(w_k))^2} - \frac{s_p^k}{(\phi_p^i(w_k))^2}\right) \int_\Omega (w_k^i - w_*^i)^2 p(w_*) dw_* + \mathbf{O}(r^2).
\end{aligned}$$

To improve readability the superscripts and subscripts k are dropped in the remainder of the proof and the base point $w = w_k$ is used. If no other argument is specified all probability distributions are evaluated at the point $w = w_k$. Let

$$\begin{aligned}
\mu_i &:= \frac{s_q}{(\phi_q^i)^2} \int_\Omega (w^i - w_*^i)^2 p(w_*) dw_* \\
a_i &:= \frac{s_q}{s_p} \frac{(\phi_p^i)^2}{(\phi_q^i)^2}.
\end{aligned}$$

Observe that $\mu_i \geq 0$ for all i. Using these definitions then define

$$F(a) := \sum_{i=1}^N \mu_i \left(1 - \frac{1}{a_i}\right).$$

where we consider $F(a)$ as a function of the new variables $\{a_i\}$. The variables a_i may in turn be thought of as functions of the product prior distributions ϕ_p^i. Note that

$$F(a) = F + \mathbf{O}(r^2).$$

The approach taken is to show that for a fixed $q = \prod \phi_q^i$ distribution then the result holds for all product distributions $p = \prod \phi_p^i$ satisfying the theorem conditions. The set of all such distributions is parameterized by the variables $a_i > 0$, $i = 1, \dots, N$. By inspection, it is easily verified that

$$F(a) \to -\infty \text{ for } a_i \to 0, \quad i = 1, \dots, N.$$

Since we need consider only the set $\{a_i > 0 \colon i = 1, \dots, N\}$ then there are no positive asymptotes of $F(a)$ and consequently $F(a)$ must have a global supremum.

Note that $\mu_i > 0$, $i = 1, \dots, N$ are the scaled variances of the w_*^i around the arbitrary reference point w^i. Recalling Eq. 55 one has

$$\prod_{i=1}^{N} a_i = \frac{s_q^N}{s_p^N} \frac{\prod (\phi_p^i)^2}{\prod (\phi_q^i)^2} = \frac{s_q^N}{s_p^N} \frac{p^2}{q^2} = \frac{p}{q} \tag{62}$$

This is the constraint on the variables a_i introduced by the conditioning associated with the step-size selection. Using this constraint one may write

$$F(a) = \sum_{i=1}^{N-1} \mu_i \left(1 - \frac{1}{a_i}\right) + \mu_N \left(1 - \frac{q}{p} \prod_{j=1}^{N-1} a_j\right).$$

Computing the partial derivative of $F(a)$ with respect to a_i yields

$$\frac{\partial F(a)}{\partial a_i} = \mu_i \frac{1}{(a_i)^2} - \mu_N \frac{q}{p} \frac{\prod_{j=1}^{N-1} a_j}{a_i}.$$

Thus, the critical points of $F(a)$ are characterized by the condition

$$\frac{\mu_i}{a_i} = \mu_N \frac{q}{p} \prod_{j=1}^{N-1} a_j = \frac{\mu_N}{a_N}, \quad i = 1, \dots, N. \tag{63}$$

In particular, there is a unique critical point of $F(a)$. It follows that this critical point must be the global maximum of $F(a)$ on the set $\{a_i > 0 \mid i = 1, \dots, N\}$. Evaluating the function F at the critical point one obtains

$$\begin{aligned} F_{\text{crit}}(w) &:= F(a_{\text{crit}}) + \mathbf{O}(r^2) \\ &= \sum_{i=1}^{N-1} \mu_i - (N-1)\frac{\mu_N}{a_N} + \mu_N - \frac{\mu_N}{a_N} + \mathbf{O}(r^2) \\ &= \sum_{i=1}^{N} \mu_i - N\frac{\mu_N}{a_N} + \mathbf{O}(r^2) \end{aligned}$$

where $F_{\text{crit}}(w)$ denotes the dependence on the base point $w \in \Omega$ of each critical value of F. Premultiplying the constraint in (62) by $1/\prod \mu_i$ and evaluating at the critical point one obtains

$$\frac{\prod_{i=1}^N a_i}{\prod_{i=1}^N \mu_i} = \left(\frac{a_N}{\mu_N}\right)^N = \frac{p}{q\prod_{i=1}^N \mu_i}.$$

Consequently, at a critical point one has

$$\frac{1}{a_N} = \left(\frac{q}{p}\right)^{\frac{1}{N}} \left(\prod_{i=1}^N \frac{\mu_i}{\mu_N}\right)^{1/N}.$$

It follows that

$$\begin{aligned} F_{\text{crit}}(w) &\leq \sum_{i=1}^N \mu_i - N\mu_N \left(\prod_{i=1}^N \frac{\mu_i}{\mu_N}\right)^{1/N} \left(\frac{q}{p}\right)^{\frac{1}{N}} + \mathbf{O}(r^2) \\ &= \sum_{i=1}^N \mu_i - N \left(\prod_{i=1}^N \mu_i\right)^{1/N} \left(\frac{q}{p}\right)^{\frac{1}{N}} + \mathbf{O}(r^2). \end{aligned}$$

Multiplying condition (60) in the theorem statement by s_q and exploiting the diagonal structure of the metric and the covariance matrix Σ_p yields

$$\begin{aligned} &\frac{1}{N} s_q \text{tr}\left(G_q^{-1}(w_k)\Sigma_p(w_k)\right) \\ &\qquad = \frac{1}{N}\sum_{i=1}^N \frac{s_q}{(\phi_q^i)^2}\int_\Omega (w^i - w_*^i)^2 p(w_*)dw_* \\ &\qquad = \frac{1}{N}\sum_{i=1}^N \mu_i \\ &\qquad \leq s_q \det\left(G_q^{-1}(w_k)\Sigma_p(w_k)\right)^{\frac{1}{N}} \\ &\qquad = \left(\prod_{i=1}^N \frac{s_q \int_\Omega (w^i - w_*^i)^2 p(w_*)dw_*}{(\phi_q^i)^2}\right)^{\frac{1}{N}} \\ &\qquad = \left(\prod_{i=1}^N \mu_i\right)^{\frac{1}{N}} \end{aligned}$$

Using this condition the value of $F_{\text{crit}}(w)$ may be overbounded by

$$F_{\text{crit}}(w) \leq \sum_{i=1}^N \mu_i \left(1 - \left(\frac{q}{p}\right)^{\frac{1}{N}}\right) + \mathbf{O}(r^2)$$

Applying Eq. 59 it follows that for all $w \in \Omega$ there exists a $r_2(w) > 0$ such that $F_{\text{crit}}(w) < 0$. Set $r_2 = \inf_{w\in\Omega}\{r_2(w)\} > 0$. Choose $r \leq \min\{r_1, r_2\}$. It follows that $F_{\text{crit}}(w) < 0$ and $E_{w_*}^p\left[\Delta\psi_q(w_k)\right] < 0$ for all $w < 0$. This completes the proof. ■

Remark 9 *The set Ω used in Theorem 8 need not be the full set on which the probability distributions p and q are defined. In practice, for $\epsilon > 0$ sufficiently small, one may choose*

$$\Omega_p = \{w \in \Omega \,|\, p(w) < q(w) - \epsilon \text{ and Eq. 60 holds}\}$$

The set Ω_p is then the set of target weight vectors w_* on which the learning algorithm based on the prior distribution p outperforms that based on the prior distribution q.

It is of interest to consider in more detail the two Conditions 59 and 60. Equation 59 is a condition on the local value of the prior distribution. Ignoring the other conditions of theorem then this condition states that if the relative probability that the true weight vector is close to the present estimate is low then the algorithm designed according to the true prior distribution out performs its competitor. This should lead the algorithm to converge more quickly into a region in which the relative probability is comparable. If the relative probability that the true weight vector is close to the present estimate is high then nothing may be said about the relative performance of the algorithms. It should be noted that the reverse implication on performance is not a consequence of the theorem since the average descent properties of the algorithms are conditioned with respect to the true prior distribution. The second condition (Eq. 60) provides a link between the global and local properties of the distribution and the preferential structure. Condition (60) can be replaced by the underlying condition

$$\frac{q}{p} > \frac{\left(\sum_{i=1}^{N} \mu_i\right)^N}{N^N \prod_{i=1}^{N} \mu_i}$$

that is a sufficient condition to ensure that F_{crit} is negative (for sufficiently small $r > 0$). We do not have a good interpretation of this condition.

10. Simulations

In this section we present some simulation results. In order to do so, 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_k. The options available to choose s_k include:

- Following Amari (1998) one may argue that the asymptotic stability of the algorithm is an adaquate measure of performance. Thus, any sufficiently small step-size selection is satisfactory.

- Pick the "optimal" $s_k = s$ according to Kivinen and Warmuth (1997). The way they do this depends on knowledge of the sequence of examples and the true weight vector. Even assuming knowledge of the process generating the examples, and the true weight vector, this optimal choice will depend on the *length* of the training sequence. Whilst there are ways of dealing with the fact that this choice requires knowledge of things impossible to know (see Kivinen and Warmuth, 1997, Section 5.1), it does seem a difficult way to proceed. After all, as Kivinen and Warmuth (1997, Section 9.2) say

  > "In applying a learning algorithm, one is usually not so much concerned with the cumulative loss as with the quality of the final hypothesis."

- Adopt the standard signal processing method of comparison (cf. Hill and Williamson, 1999, 2001): determine steady-state Mean Square Error (MSE) as a function of s, and then choose s to achieve a fixed steady-state MSE. Then compare algorithms in terms of their speed of convergence.
- Adopt the choice implied by equation 55. This is in fact what we do in the present paper.

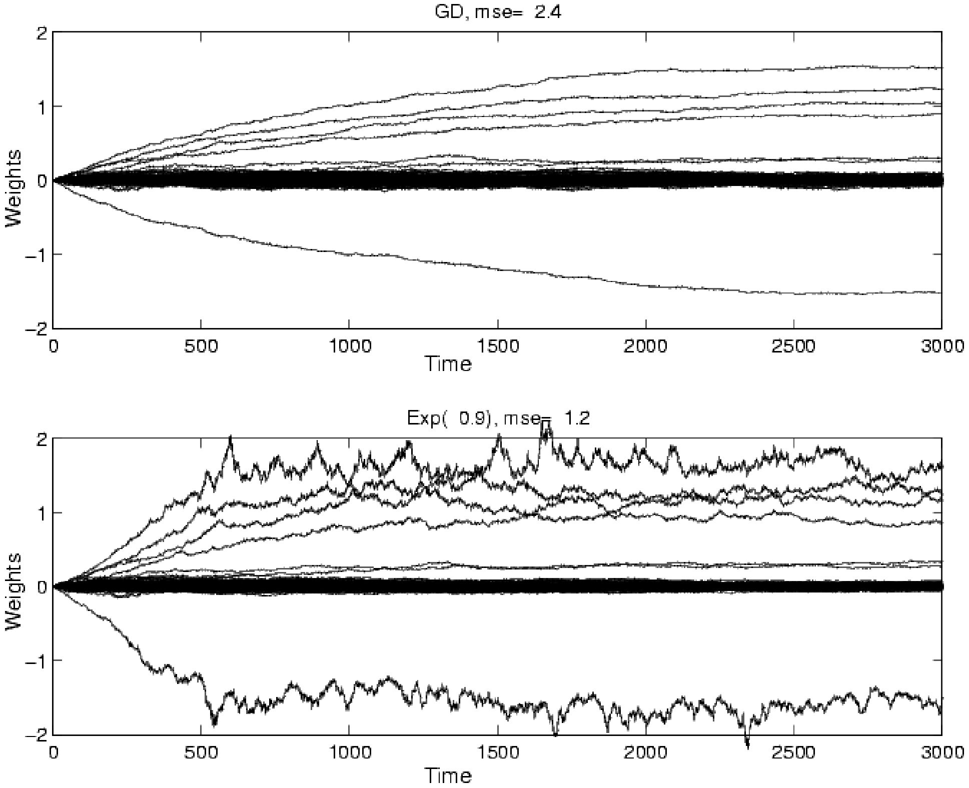

Figure 1: Comparison of exp(α) ($\alpha = 0.9$) algorithm with standard Gradient Descent. x_k was drawn independently from a uniform distribution on $[-\frac{1}{2}, \frac{1}{2}]^M$, $M = 1500$. The step-size for the GD algorithm was $s_{\mathrm{GD}} = 0.0001$. Step size for exp(α) chosen according to (55) (see text). The target had the non-zero values inferrable from the diagram (approximately 1.6, 1.3, 1.1, 0.8,0.25, 0.2, -1.5), with all the remaining values zero (a fraction of 0.005 of the dimensions of w_* were non-zero). Gaussian noise of standard deviation 0.06 was added to the y_k sequence. Both algorithms were started from the initial condition $(\frac{1}{3000}, \dots, \frac{1}{3000})'$. In the graphs, only the first 100 dimensions of w_k have been plotted for the sake of clarity. (The remaining dimensions all had target values of zero.) The indicated Mean Square Errors (MSE) were estimated from the final fifth of the run.

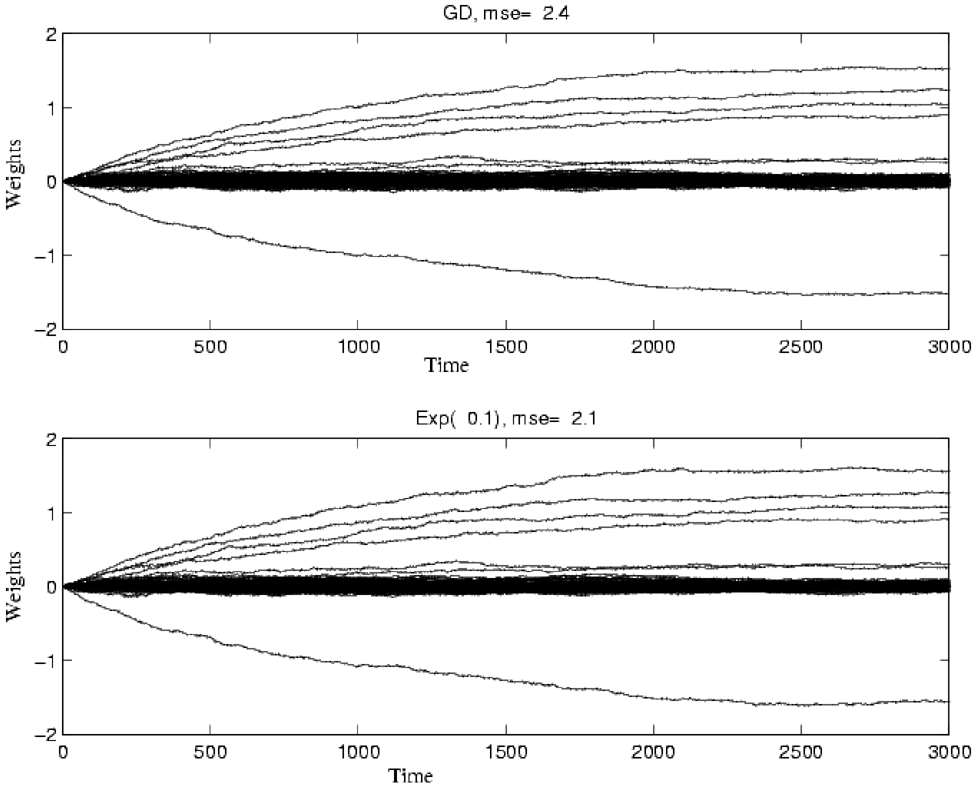

Figure 2: Same as Figure 1 except with $\alpha = 0.1$

In the experiments reported, we used a fixed step size s_{GD} for the standard Gradient Descent algorithm and then used (55) to determine s_k for the comparison algorithm. The value of s_{GD} was somewhat arbitrary, but chosen to ensure a clear stability margin for the GD algorithm.

For the exp(α) algorithm, $\phi(w) = \prod_{i=1}^{N} e^{-\alpha|w^i|}$. Thus from (55)

$$\frac{\left(s^k_{\exp(\alpha)}\right)^N}{\prod_{i=1}^{N} e^{-\alpha|w^i|}} = \frac{(s_{\mathrm{GD}})^N}{1} \tag{64}$$

$$\Rightarrow \quad s^k_{\exp(\alpha)} = s_{\mathrm{GD}} e^{-\frac{\alpha}{N}\sum_{i=1}^{N}|w_k^i|} = s_{\mathrm{GD}} e^{-\alpha\|w_k\|_1/N} \tag{65}$$

Similarly for the EGclipped(c) algorithm, with $\phi_c(w) = \prod_{i=1}^{N} \min(c, \frac{1}{|w^i|})$ we choose

$$s^k_{\mathrm{EGclipped}(c)} = s_{\mathrm{GD}} \phi_c(w_k) \tag{66}$$

With reference to Figure 1, it can be seen that by 3000 iterations both algorithms have reached a "steady-state" where each component is being jiggled around the true value by the added noise. The key difference between the GD algorithm and the Exp(0.9) is that

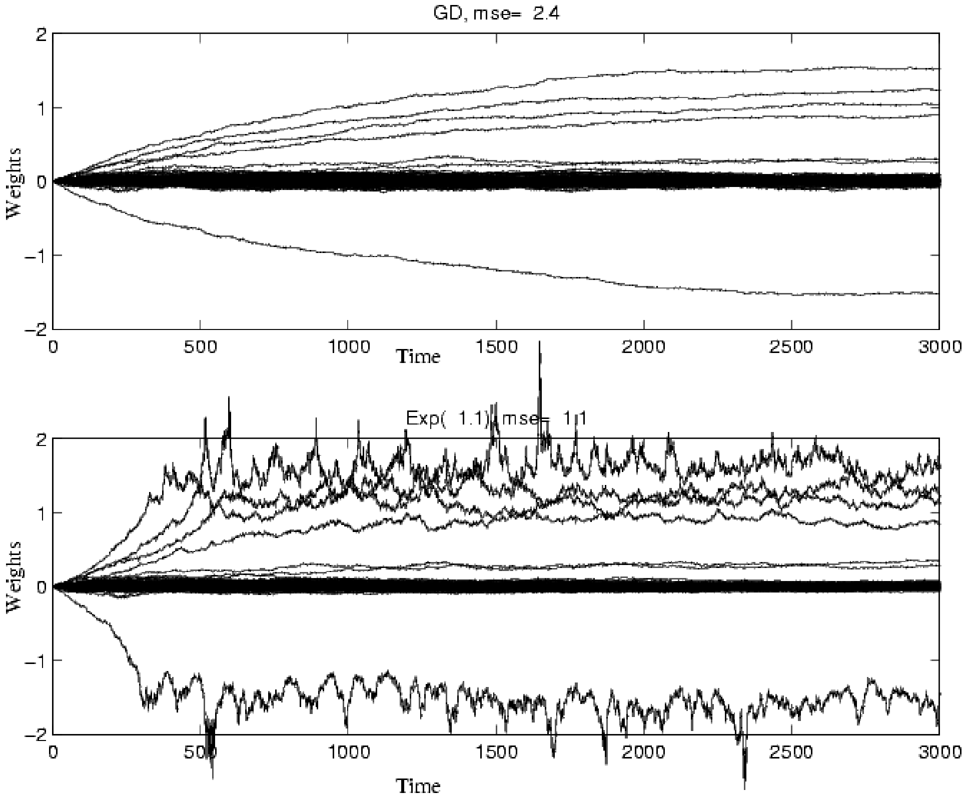

Figure 3: Same as Figure 1 except with $\alpha = 1.1$

the effective step size for the non-zero components is larger; or equivalently, the effective step size for the zero components is smaller, which is what leads to the smaller steady-state MSE even though the GD algorithm converges slower (taking around 2500 steps to reach a steady state, whereas Exp(0.9) reached steady state in around 1000 steps).

The $\exp(\alpha)$ and EGclipped(c) algorithms outperform the standard GD algorithm on the problems considered. This is not surprising given the choice of the prior. One can see that the convergence speed is qualitatively similar, but that the steady state MSE of the $\exp(\alpha)$ and EGclipped(c) algorithms is rather smaller than that for the GD algorithm. It can also be seen that increasing c or α, leading to a more extreme prior distribution, leads to algorithms whose behaviour is noticeably different from the GD algorithm. Letting $c \to \infty$ in EGclipped(c) leads to the (natural gradient) EG algorithm. We have observed that the singularity at the origin in $\phi_\infty(w)$ causes numerical difficulties.

We also observed it was necessary to replace $\Phi_\alpha^{-1}(x)$ given by (49) by

$$\tilde{\Phi}_\alpha^{-1}(x) = -\mathrm{sgn}(x)\ln(-\alpha|x| + \epsilon)/\alpha \tag{67}$$

where ϵ was chosen as 10^{-9} in the experiments reported in order to avoid numerical problems.

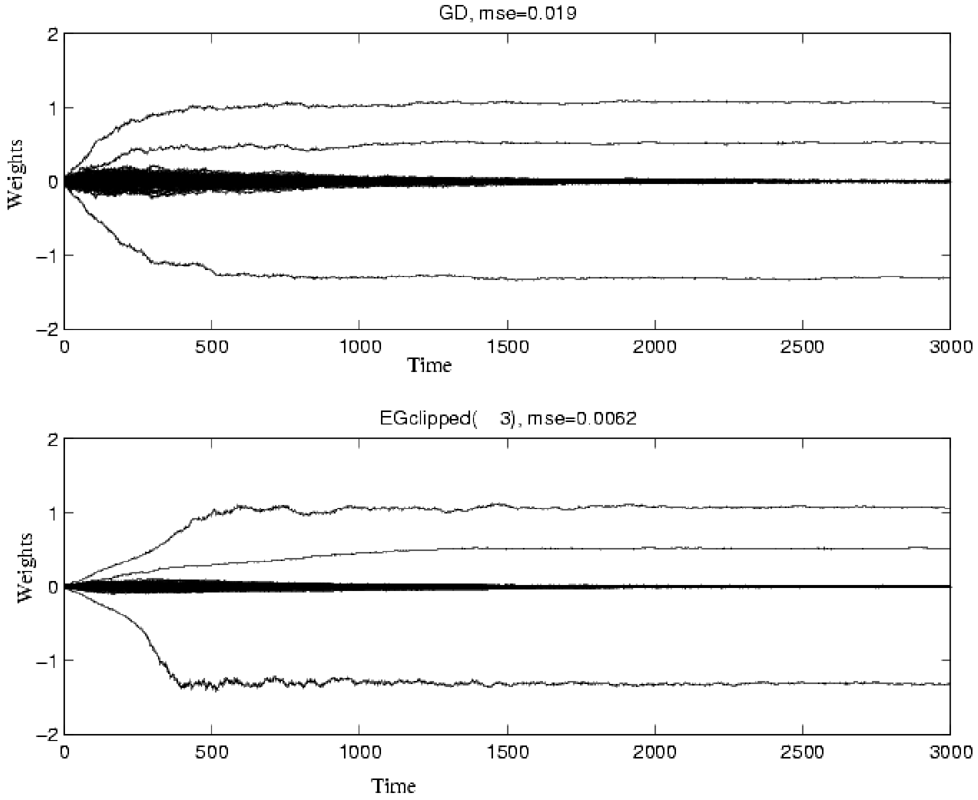

Figure 4: Comparison of the EGclipped(c) algorithm ($c = 3$), x_k drawn iid uniformly on $[-\frac{1}{2}, \frac{1}{2}]^M$, ($M = 300$), $s_{\mathrm{GD}} = 0.005$. Step size for EGclipped(c) chosen according to (55) (see text). The target had the non-zero values inferrable from the diagram, with all the remaining values zero (a fraction of 0.01 of the dimensions of w_* were non-zero). Gaussian noise of standard deviation 0.06 was added to the y_k sequence. Both algorithms were started from the initial condition $(\frac{1}{3000}, \ldots, \frac{1}{3000})'$. In the graphs, only the first 100 dimensions of w_k have been plotted for the sake of clarity. (The remaining dimensions all had target values of zero.) The indicated Mean Square Errors (MSE) were estimated from the final fifth of the run.

These numerical difficulties would be absent for any $\phi(w)$ satisfying

$$c_1 \leq \phi(w) \leq c_2$$

for all w and some $0 < c_1 < c_2 < \infty$.

11. 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

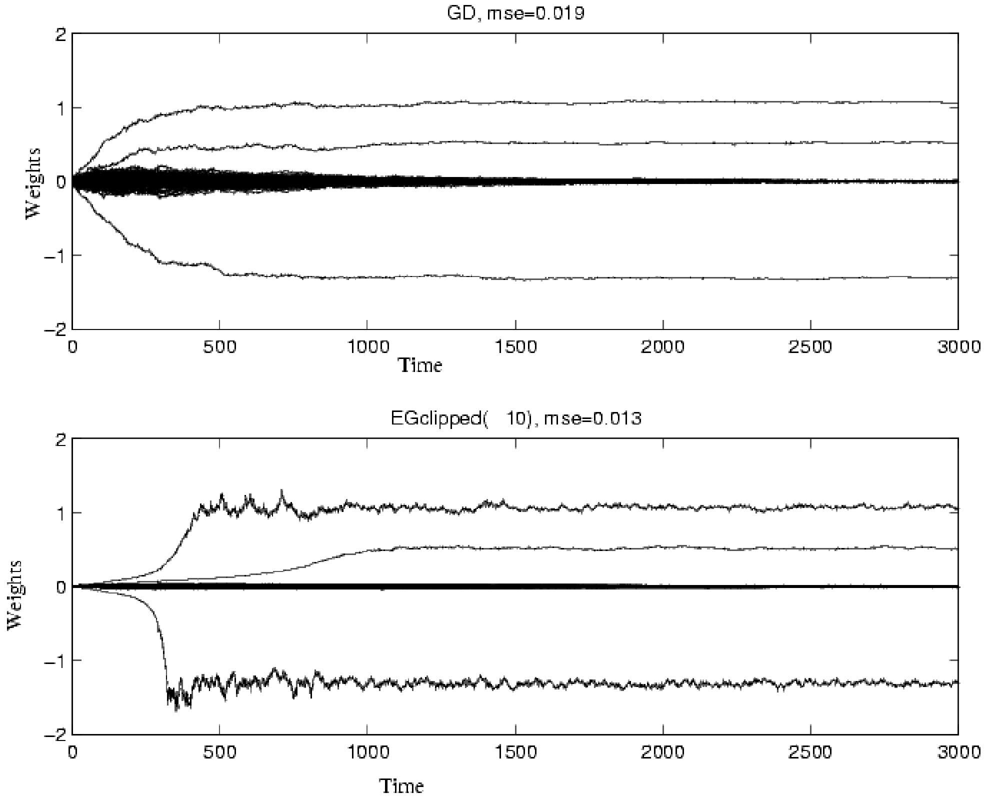

Figure 5: Same as Figure 1 except with $c = 10$.

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 (Kivinen and Warmuth, 1997, Hill and Williamson, 2001) has shown how the EG algorithm can perform well in real situations where the target weight 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. An advantage of the proposed framework is that, in the case of product prior distributions, it is a simple matter to generate algorithms optimized to the prior information available. An example of such an application as well as a simplified approximation to the algorithms developed in the present paper is given by Martin et al. (2001).

There are several directions for further research on this topic. The most obvious are to consider a wider range of loss functions and to see if there is a closer connection with the work on Bregman divergences for analyzing these algorithms. One can also envisage combining the framework proposed in the present paper with techniques from information geometry in order to draw stronger connections with recent work such as that of Amari (1998). Indeed, it is reasonable to ask if one can derive a suitable geometry and associated

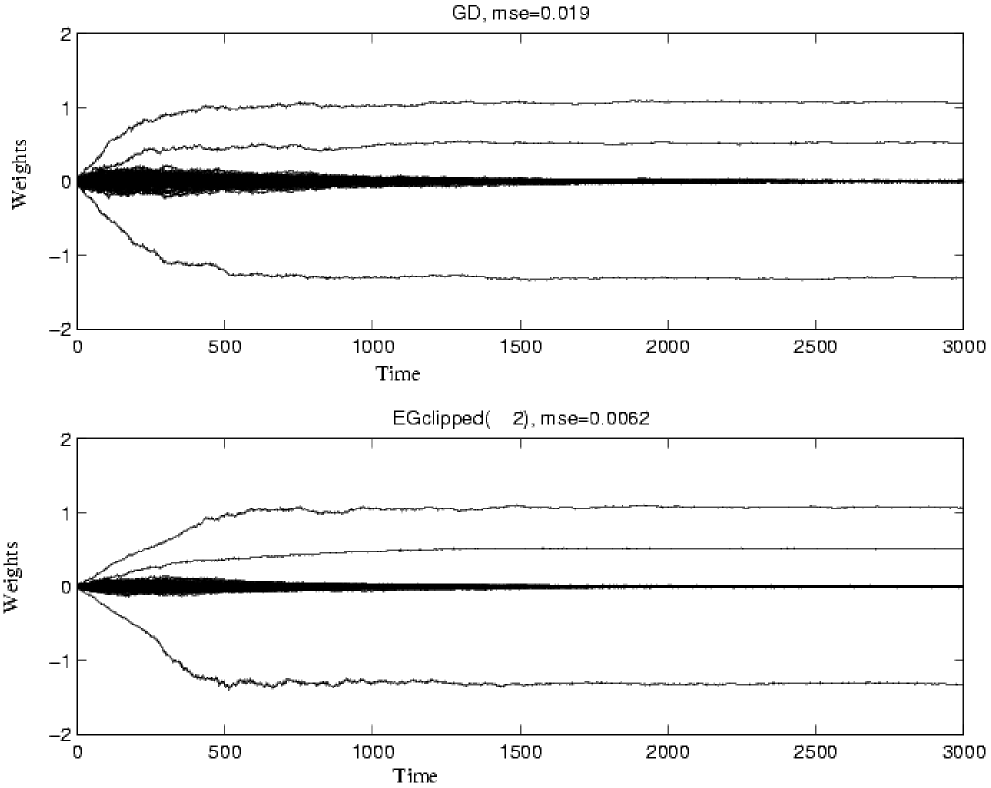

Figure 6: Same as Figure 1 except with $c = 2$.

natural gradient algorithms with respect to both a prior distribution on the weights and the likelihood function associated with measurement noise?

Another direction is to explore the connection (if any) with the prequential approach Dawid (1984), which studies sequential prediction from a viewpoint associated with Bayesian statistics.

Acknowledgments

This work was support by the Australian Research Council. RW would like to thank Manfred Warmuth for a discussion at COLT97 which provided a major impetus for this work. Thanks to the referees for some good suggestions which hopefully improved the readability of the paper.

Appendix A. Background on Riemannian Geometry

To obtain a better understanding of the subject it is necessary to enter into the details. An excellent quick overview of differential geometry and the basics of Riemannian metrics is given in (Helmke and Moore, 1994, appendix C). Good introductory texts have been written by Boothby (1986) and Lee (1997). A deeper knowledge can be obtain from Spivak (1979). Murray and Rice (1993) treat similar material and also discuss the geometry of statistical inference.

The modern form of Riemannian geometry provides a language in which the structure of curved spaces can be analysed. Riemann was motivated by the ongoing effort to understand the significance of Euclid's fifth postulate during the first half of the nineteenth century. This question must have been an important topic of discussion amongst all mathematicians of the time, however, Riemann's early work was in the field of geometric properties of analytic functions of complex variables, conformal mappings and connectivity of surfaces. In 1854 he submitted a dissertation for his *Habilitation* on representing complex functions using trigonometric series. He was also required to give a public dissertation and submitted three possible topics to his examining committee. The first two topics were subjects in which he was recognized as having made significant contributions. He clearly expected to be asked to speak on this work and, perhaps in wishing to impress Gauss, a member of his Habilitation committee, he chose *Über die Hypothesen welche der Geometrie zu Grunde liegen* (On the hypotheses that lie at the foundations of geometry)[4] as his third topic. Gauss called his bluff and Riemann had only a few months in which to prepare material that would impress the Habilitation committee. He had never been one to restrict his claims to those for which he had a rigorous development and his lecture was a sweeping introduction to a vague form of a differentiable manifold, the concept of local quadratic forms, a general connection to the concept of curvature, and finally a discussion of the implications of these concepts to our understanding of the world we live in. He provided a first vision of modern differential geometry, Riemannian manifolds and general relativity wrapped up into a single incoherent view of the world. Fortunately, Gauss was present to grasp the importance of the material presented. An indication of the visionary nature of Riemann's lecture is gained by noting that it took another sixty years before the concept of a differentiable manifold was properly developed in 1913 by Weyl (1923).

The aim of Riemannian geometry is to provide an intrinsic understanding of geometry based on observations made within the manifold considered. This should be contrasted to an extrinsic interpretation of geometry such as is possible, for example, for a sphere embedded in $\mathbb{R}^3$. The sphere is clearly curved, (from the perspective of an observer in the 'flat' Euclidean space $\mathbb{R}^3$), and it is simple enough to develop measures of this curvature such as Euler's principal curvatures[5] that involve extrinsic measurements. The problem is considerably less clear if one imagines a mythical being living within 2-dimensional space on the surface of the sphere. A valid intrinsic understanding of geometry would allow such a creature to make measurements to determine whether they were living on 'flat' Euclidean space compared to, for example, an extremely large sphere (eg. the polygonal creatures living in the world created by Abbot (1992)). Gauss certainly believed that this was possible and his work on the curvature of surfaces (Gauss, 1965) emphasizes that the

[4] A transcript of Riemann's lecture was published in (Riemann, 1868). An English translation was made by Spivak and appears in volume 2 of his comprehensive treatise on geometry (Spivak, 1979). Our knowledge of this work came from the recent delightful book by McCleary (1994).

[5] The principal curvatures of a surface in $\mathbb{R}^3$ are the maximal and minimal curvature of all curves obtained by intersecting the surface with a plane (Euler, 1760).

infinitesimal curvature of a surface (as measured by the product of the principal curvatures) is related to the difference between the sum of the angles of an infinitesimal triangle drawn at the point of interest from 180°. This is an intrinsic measure available to creatures living within the surface.

The key concept of Riemannian geometry is the concept of independence of the measure of length (and volume) from the local coordinates used to measure position. Underlying this concept is the understanding that there are no 'special' coordinates for a manifold and consequently that a geometric property is intrinsic to that manifold if and only if it is invariant under coordinate transformations. The fundamental axiom that Riemann worked from was that the infinitesimal length of a curve should be a physical quantity independent of the mathematical representation of the manifold but quite possibly dependent on the point at which the measurement is taken. In modern language, the measure of length on the manifold must be invariant under changes in local coordinates. In order for this to be true, the explicit expression of the infinitesimal measure of length must depend on the coordinates used; otherwise they would not transform with changes in the coordinates. From this principle it it is possible to derive the expression

$$ds = \sqrt{\sum_{i,j=1}^{n} g_{ij}(x) dx_i dx_j}$$

where s denotes the curve parameterization and the $g_{ij}(x)$ code the manner in which the length of the curve depends on the particular local coordinates $\{x_1, x_2, \dots, x_n\}$ used. The matrix $G(x) = [g_{ij}(x)]_{i,j}$ is called the Riemannian metric for the manifold considered. The properties of symmetry, non-degeneracy and positive definiteness of the matrix $G(x)$ correspond to physical properties of measuring length, albeit the infinitesimal limit of length. Of course, infinitesimal length may be integrated along curves to obtain a classical distance metric on the space. A curve tracing out the shortest distance between any two points according to this metric is called a geodesic. Such curves would appear as straight lines to a creature living within the manifold.

Using the above intuition is now possible to state the intrinsic difference between a curved Riemannian space and flat space. For Euclidean space there exists a 'special' coordinate system for which the infinitesimal length measure is expressed

$$ds = \sqrt{\sum_{i=1}^{n} dx_i dx_i}.$$

That is, the metric is $g_{ij}(x) = \delta_{ij}$ $(G(x) = I)$. Clearly, a manifold is intrinsically Euclidean (or 'flat') if there exists a set of coordinates for which the Riemannian metric is transformed to the identity matrix. Of course, this must hold equally at every point in the space. From symmetry there $n(n-1)/2$ linearly independent functions that define $g_{ij}(x)$, there are n degrees of freedom available by altering the coordinate function, and consequently there are $n(n-1)/2$ components of $g_{ij}(x)$ that cannot be arbitrarily fixed by variation in the coordinate chart. For example, for any two dimensional Riemannian manifold one may choose coordinates (locally) such that $g_{11} = g_{22} = 1$, however, the function $g_{12}(x) = g_{21}(x)$ will in general be non-zero and non-constant. The set of all 2-dimensional Riemannian manifolds

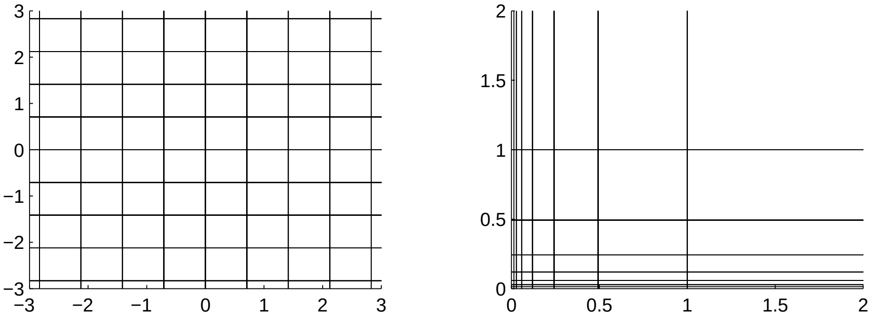

Figure 7: A grid of unit measure in the Euclidean coordinates associated with the EG-algorithm on the left maps to a grid of curvilinear coordinates in the weight space for the EG-algorithm on the right. Observe that the coordinate grid is compressed close to the axes corresponding to the property that a unit step of distance in this region only alters the actual weight vector w by a small amount.

are (locally) parameterised by the function $g_{12}(x)$. Euclidean space $\mathbb{R}^2$ corresponds to the case $g_{12}(x) \equiv 0$. The independent functions $g_{ij}(x)$ are linked to the curvature of the space.

An important observation is that the measure of length is dependent on the point at which the measurement is made in the manifold. An instructive example taken from the body of the paper is provided by considering the geometry used to analyze the EG-algorithm. The metric matrix is $G(w) = \mathrm{diag}(\frac{1}{(w^1)^2}, \ldots, \frac{1}{(w^n)^2})$ (cf. Subsection 6.2) defined on the weight space $w^i > 0$ for all $i = 1, \ldots, n$. Transforming the coordinates according to

$$u(w) := \left(\ln(w^1), \ldots, \ln(w^2)\right),$$

one obtains the constant Riemannian metric $G(u) = I_n$. Thus, the Riemannian structure introduced for the EG-algorithm is Euclidean of 'flat'. Nevertheless, the geodesics in the original weight space are not straight lines.

It is instructive to plot two examples of the mapping between the Euclidean coordinates $u(w)$ introduced above and original weight space in the local coordinates w (cf. Figures 7 and 8). In Figure 7 it is obvious that in the weight space the unit grid derived from the Euclidean coordinates gets compressed close to the i'th coordinate axis by a factor of $\sqrt{1/(w^i)^2} = \phi(w^i)$, the improper probability distribution that was introduced to model the prior knowledge assumed for the EG-algorithm. Thus, taking a unit step in this region results in very small changes in the weight vector coefficient w^i. The second figure (Figure 8) shows the curvilinear nature of the geodesics in the weight space compared to the linear geodesics in the Euclidean coordinates for the EG-algorithm.

The application of Riemannian geometry to statistical learning theory undertaken in this paper has a minor variant from classical Riemannian geometry. In statistical learning theory the weight vector w lives in a 'special' coordinate frame, namely, the coordinate frame that is linked to the linear model class of the learning problem. Thus, a preferential structure is given by a Riemannian metric defined with respect to 'special' coordinates corresponding to the weight vectors w used in the linear model class. In these coordinates the measure of

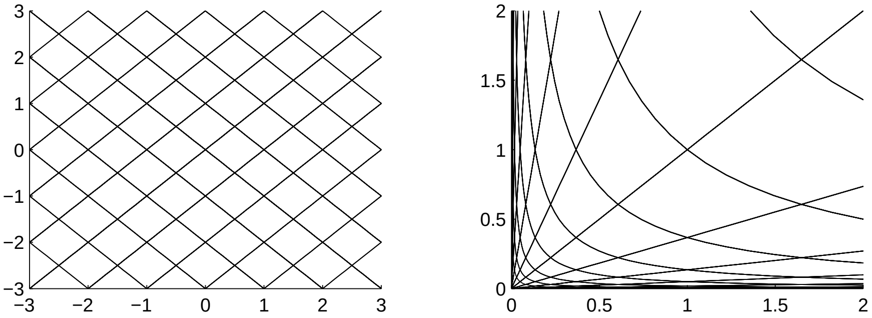

Figure 8: An analogous plot to that shown in Figure 7 except that the Euclidean coordinates have been rotated by 45°. This plot shows the curvilinear nature of the geodesic coordinates induced by the EG preferential structure in weight space. A creature living within the weight space equipped with the EG potential stucture would observe the curves on the right drawn as straight lines shown in the grid on the left.

length is certainly not constant. Thus, the geodesics appear curved when expressed in the coordinates w and the parameterization strongly contributes to the properties of the learning algorithm. The key observation is that the performance of the algorithm is measured in the 'special' weight vector coordinates. This is contrasted to classical Riemannian geometry where all local coordinate charts are equally valid. If one were to discard the weight vector coordinate representation and use, for example, the Euclidean parameterization $u := u(w)$ for the EG-algorithm it would be necessary to reparameterize the model class to obtain the same learning problem (cf. Section 2)

$$x \mapsto \langle \exp(u), x \rangle, \;\; x \in \mathbb{R}^N,$$

for u. Thus, in a learning problem it is both the preferential structure and the structure of the model class that combined define the geometry of the problem.[6]

It remains to comment on the issues involved in obtaining measurements of the Riemannian geometry of a manifold using only intrinsic measurements. Since an intrinsic measurement is made within the space a direct measure of distance is not sufficient to determine the important off diagonal metric structure. A differential measure, i.e. how the measure of length changes as a displacement is made, will provide information that can be used to infer the metric structure of the manifold. Riemann showed that this approach was sufficient to intrinsically compute the curvature of a manifold equipped with a infinitesimal quadratic distance measure (or Riemannian metric). As a consequence the curvature of a Riemannian manifold can be computed from first derivatives of the metric functions $g_{ij}(x)$. The actual expressions in coordinate form are intricate and difficult to work with. It was Christoffel (1869) who introduced a formal representation of this structure in terms of a set

[6] An interesting consequence of this observation is that any non-singular parameterization of the learning problem $x \mapsto \langle f(u), x \rangle$, (linear in the measurement x) can be represented as linear model class along with a preferential structure derived from the local coordinate chart $w := f(u)$, $u = f^{-1}(w)$.

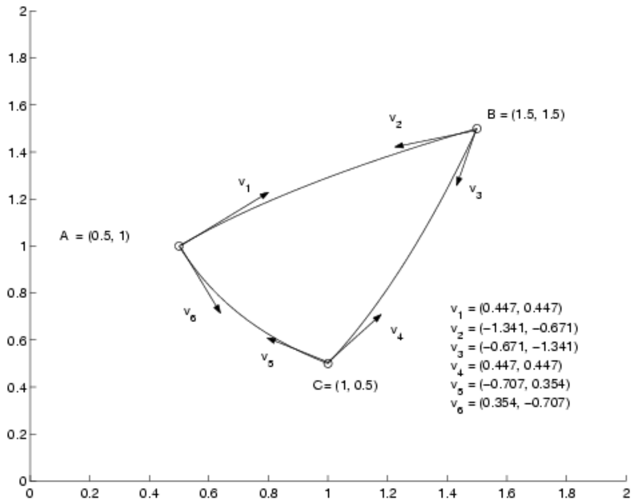

Figure 9: A triangle drawn in coordinated w of weight space.

	Correct	Incorrect
$\angle A$	71.5°	108°
$\angle B$	37°	37°
$\angle C$	71.5°	108°
Total	180°	253°

Figure 10: A table showing the angles between vertices of the triangle shown in Figure 9.

of symbols and transformation rules under change of coordinates. The Christoffel symbols can be used to compute intrinsic geometric properties of a Riemannian manifold such as curvature, parallel transport, geodesics, etc. In this paper, the Christoffel symbols are used simply to obtain an explicit ODE for the geodesic equations for the learning algorithms considered.

A final remark on the measurement of angles on Riemannian manifolds is of interest. It was well understood by workers in the early nineteenth century that for 2-dimensional surfaces the sum of the internal angles of a triangle is linked to the curvature of the surface. Indeed, for a triangle the internal angles sum to a value that differs from 180° by a factor that depends only on the curvature and the area of the triangle. Of course the angles must be measured relative the Riemannian metric that defines the geometry. Thus, the angle between two vectors at a point in the space is

$$\angle u, v = \arccos\left(\frac{u^T G(x) v}{(u^T G(x) u)^{\frac{1}{2}} (u^T G(x) u)^{\frac{1}{2}}}\right)$$

Consider the triangle drawn in Figure 9 constructed from geodesic curves for the geometry introduced for the EG-algorithm. Earlier it was shown that a smooth change of coordinates exists that transforms the metric into the Euclidean metric and hence that the space is flat. Computing the angles of the three points according to formulae above yields the correct values shown in Table A. As expected the correct calculation leads to total of 180° corresponding to the fact that the preferential structure for the EG-algorithm is flat. The incorrect values shown are the angles computed without using the Riemannian metric and correspond to the angles that one observes visually in Figure 9. Summing the incorrect angles yields something absurd. This example demonstrates the importance of the Riemannian metric in measuring angles as well as providing a measure of distance.

References

Edwin Abbot. *Flatland: A Romance of Many Dimensions.* Thrift editions. Dover pubn, 1992. Unabridged version of the revised edition, 1884.

Hirotugu Akaike. The interpretation of improper prior distributions as limits of data dependent proper prior distributions. *Journal of the Royal Statistical Society, Series B*, 42 (1):46–52, 1980.

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

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

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

Ole E. Barndorff-Nielsen. *Parametric Statistical Models and Likelihood.* Springer, Berlin, 1988.

William M. Boothby. *An Introduction to Differentiable Manifolds.* Academic Press, London, 1986.

Joseph T. Chang and David Pollard. Conditioning as disintegration. *Statistica Neelandica*, 51(3):287–317, 1997.

Elwin Bruno Christoffel. Über die Transfmoration der homogenen Differentialausdrücke zweiten Grades. *Crelle*, 70:46–70, 1869.

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

A. Phil Dawid. Statistical theory: The prequential approach. *Journal of the Royal Statistical Society, Series A*, 147:278–292, 1984.

Scott C. Douglas and Shun-ichi Amari. Natural-gradient adaptation. In Simon Haykin, editor, *Unsupervised Adaptive Filtering. Volume 1: Blind Source Separation*, pages 13–61, New York, 2000. John Wiley.

Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification.* John Wiley, New York, 2nd edition, 2001.

Leonhard Euler. Recherches sur la courbure des surfaces. *Mémoires de l'academie es Sciences*, 16:119–143, 1760.

Terrence L. Fine. *Feedforward Neural Network Methodology.* Springer, New York, 1999.

Karl F. Gauss. *General Investigations of Curved Surfaces.* Raven Press, New York, USA, 1965. Tranlated from material published in 1827.

Claudio Gentile and Nick Littlestone. The robustness of the p-norm algorithms. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 1–11, 1999.

Geoffrey J. Gordon. Approximate solutions to markov decision processes. PhD thesis, Department of Computer Science, Carnegie-Mellon University, 1999a.

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

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.

John A. Hartigan. *Bayes Theory.* Springer, New York, 1983.

Mohamad H. Hassoun. *Fundamentals of Artificial Neural Networks.* MIT Press, Cambridge, MA, 1995.

Uwe Helmke and John B. Moore. *Optimization and Dynamical Systems.* Springer, London, 1994.

Simon I. Hill and Robert C. Williamson. An analysis of the exponentiated gradient descent algorithm. In *Proceedings of the International Symposium on Signal Processing and its Applications (ISSPA'99)*, pages 379–382, 1999.

Simon I. Hill and Robert C. Williamson. Convergence of exponentiated gradient algorithms. *IEEE Transactions on Signal Processing*, 49(6):1208–1215, 2001.

Jørgen Hoffmann-Jorgensen. *Probability with a View Toward Statistics (Volume II).* Chapman and Hall, New York, 1994.

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

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.

Jyrki Kivinen and Manfred K. Warmuth. Relative loss bounds for multidimensional regression problems. *Machine Learning*, 45:301–329, 2001.

John M. Lee. *Riemannian Manifolds : An Introduction to Curvature*, volume 176 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, 1997.

Richard K. Martin, William A. Sethares, Robert C. Williamson, and C. Richard Johnson Jr. Exploiting sparsity in adaptive filters. Proceedings of 2001 Conference on Information Sciences and Systems, The Johns Hopkins University, March 2001. Extended version submitted to *IEEE Transactions on Signal Processing*, 2001.

John McCleary. *Geometry from a differentiable point of view.* Cambridge University Press, Cambridge, UK, 1994.

Michael K. Murray and John W. Rice. *Differential Geometry and Statistics.* Chapman and Hall, London, 1993.

Bernhard Riemann. Über die Hypothesen welche der Geometrie zu Grunde liegen. *Abhandlungen der Königlichen Gesellschaft der Wissenschaften zu Göttingen*, 13, 1868.

Christian P. Robert. *The Bayesian Choice.* Springer, New York, 1994.

Albert N. Shiryaev. *Probability.* Springer, New York, 2nd edition, 1996.

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

Murray Spivak. *A comprehensive introduction to differential geometry.* Publish or Perish, Inc., Wilmington, Delaware, USA, 2nd edition, 1979. 5 volumes.

Manfred K. Warmuth and Arun K. Jagota. Continuous and discrete-time nonlinear gradient descent: Relative loss bounds and convergence. Preprint, University of California, Santa Cruz. http://www.cse.ucsc.edu/~manfred/pubs/differential.ps, September 1997.

Hermann Weyl. *Die Idee der Riemannschen Fläche.* Teubner, Liepzig, Germany, 1923. English translation: *The concept of a Riemann surface*, Addison-Wesley, Reading, Massachusetts, USA.