Intrinsic Geometry of Stochastic Gradient Descent Algorithms

Robert E. Mahony
Department of Engineering
Australian National University
Canberra, ACT 0200, Australia
Robert.Mahony@anu.edu.au

Kim Blackmore
Department of Engineering
Australian National University
Canberra, ACT 0200, Australia
Kim.Blackmore@anu.edu.au

Krzysztof A. Krakowski
Department of Engineering
Australian National University
Canberra, ACT 0200, Australia
Krzysztof.Krakowski@anu.edu.au

Robert C. Williamson
National ICT Australia (NICTA), and
Australian National University
Canberra, ACT 2601, Australia
Bob.Williamson@anu.edu.au

Abstract

We consider the intrinsic geometry of stochastic gradient descent (SG) algorithms. We show how to derive SG algorithms that fully respect an underlying geometry which can be induced by either prior knowledge in the form of a preferential structure or a generative model via the Fisher information metric. We show that using the geometrically motivated update and the “correct” loss function, the implicit and explicit discrete time updates are, under certain conditions, identical. This new loss function reduces to least square loss for linear regression with Gaussian measurement noise. We also show that the seemingly obvious requirement that the loss function is convex is not appropriate in non-flat geometries. We illustrate the power of the new framework by deriving an algorithm for a regression problem over a multinomial distribution.

1 Introduction

The stochastic gradient (SG) descent or Widrow-Hoff algorithm [1] is the best known approach used to derive on-line learning algorithms. The SG algorithm displays excellent performance for linear regression problems subject to Gaussian measurement noise. Each step of the SG algorithm can be motivated as maximum likelihood estimate of a density derived from the Gaussian observation of the linear regression conditioned by a Gaussian prior around the present parameter estimate. The conditioning process can be viewed as regularisation. In the last decade a number of algorithms for on-line learning have been proposed that improve on the performance of the SG descent algorithm for particular situations. The exponentiated gradient (EG) descent algorithm [2] out-performs the SG algorithm when the weight vector of the generative noise model is sparse. The EG algorithm uses a classical least squares cost but introduces a regularisation based on a Bregman divergence. Following early work in this area, a range of gradient descent algorithms inspired
by the EG algorithm have been analyzed for both classification and regression problems \cite{3,4,5,6,7}. State of the art techniques rely on the existence of a well defined statistical divergence function that can be used for the regularisation process. These techniques are not applicable to a range of important problems with structure such as curved exponential families \cite{8} or preferential structures \cite{9}.

We propose a new class of iterative update algorithms, termed geometric gradient (GG) algorithms, that provide an efficient on-line learning algorithm for any parametric regression problem with a well defined geometry. The framework of a regression problem involves estimating a parameter $\omega$ from a sequence of instances of data $x_k \in \mathbb{R}^n$ associated with labels $y_k \in \mathbb{R}$ related by a generative noise model

$$y_k = f(\omega, x_k) + \eta_k.$$  \hfill (1)

The noise $\eta_k$ is zero mean i.i.d. noise drawn from a set of probability density functions parameterised by $\omega \in M$. We assume that the set $M$ is equipped with a Riemannian metric which can come from either the “information geometry” of the underlying generative noise model \cite{10,11} or an a priori preferential structure \cite{9} on $M$. The update is chosen to optimise a tradeoff between a loss function $L_k(\omega_k) = L(\omega_k, (y_k, x_k))$ that measures the loss associated with a given parameter estimate $\omega_k$ for the given data instance $(y_k, x_k)$, and a regularisation function that penalises deviation from the existing parameter estimate. We use the Riemannian geometry of the problem to define both functions. The most natural geometric definition of the regularisation function at the point $\omega_k$ is the squared Riemannian distance $\text{dist}(\omega, \omega_k)^2$. The squared distance is not equivalent to the common divergence functions considered in statistical inference; however, it has the advantage that it is well defined in all the cases of interest and we show that it leads to well conditioned, efficient algorithms.

We argue below that there is a natural choice of loss function $L_k(\omega)$ (cf. Equation (4)) compatible with the regularisation function $D(\omega, \omega_k)$ derived from Riemannian distance. The geometric gradient iterate is given by solving the optimisation problem

$$\omega_{k+1} = \arg \min_{\omega} (D(\omega, \omega_k) + \lambda L_k(\omega)),$$ \hfill (2)

for $\lambda > 0$ a positive constant scalar termed the learning weight. A key property of choosing a compatible loss and regularisation functions is the existence of an explicit solution to this optimisation problem. The resulting algorithm can be thought of as a generalisation of the natural gradient algorithm \cite{12}. We show that proposed algorithm specialises to the normalised stochastic gradient (NSG) algorithm for linear regression problems subject to Gaussian noise. We also provide a concrete example of the algorithm for linear regression over the multinomial distribution \cite{13,14}.

2 Two examples

The following two examples will be used to demonstrate the concepts in the paper.

Example 1(a): Linear regression subject to Gaussian noise. We consider a linear regression problem $y_k = \langle \omega_*, x_k \rangle + \eta_k$, where the data at each instance is $(y_k, x_k) \in \mathbb{R} \times \mathbb{R}^n$ and the noise $\eta_k \sim N(0, 1)$. The Riemannian metric induced by the Fisher information metric is the identity matrix $g_{ij} = \delta_{ij}$ and $M \equiv \mathbb{R}^n$ is Euclidean space. \hfill $\ddagger$

Example 2(a): Linear regression over multinomial samples. Again consider a linear regression problem $y_k = \langle z_k, x_k \rangle$, where the data at each instance is $(y_k, x_k) \in \mathbb{R} \times \mathbb{R}^n$, but now $z_k \in \mathbb{R}^n$ is a normalised vector generated from a multinomial distribution with $m$ samples:

$$p(z \mid \omega_*) = \frac{m!}{\prod_{i=1}^n m_i!} \prod_{i=1}^n (\omega^*_i)^{m_i},$$ \hfill (3)
where $\omega_*$ is the true parameter to be learned. (Here, and throughout the paper, superscripts are used to indicate index components of vectors.) It can be seen that $mz_k^i \in \mathbb{N}$ and $z_k$ lies on the simplex $\Delta^{n-1} = \{ z \in \mathbb{R}^n : \sum_{i=1}^n z_i = 1, z_i \geq 0 \}$. For the purposes of the development it is possible to think of the regression model as a classical measurement noise regression problem $y_k = \langle \omega_*, x_k \rangle + \eta_k$, where $\eta_k = \langle z_k - \omega_*, x_k \rangle$ is no longer Gaussian and $\omega_* \in \Delta^{n-1}$.

It can be shown [8] that the Fisher information metric for this class of models is equivalent to the natural embedded metric of $\Delta^{n-1} \hookrightarrow \mathbb{R}^n$ via the injection

$$
\varphi : \Delta^{n-1} \hookrightarrow S^{n-1}_{2\sqrt{m}}, \quad \varphi(\omega^1, \ldots, \omega^n) := 2\sqrt{m}(\sqrt{\omega^1}, \ldots, \sqrt{\omega^n}),
$$

where $S^{n-1}_{2\sqrt{m}}$ is the $(n-1)$-sphere of radius $2\sqrt{m}$. In these coordinates the Riemannian metric has the form $g(\omega) = \sum_{i=1}^n \frac{m}{\omega^i} (d\omega^i)^2$.

### 3 Derivation of the geometric gradient algorithm

Let $(\mathbb{M}, g)$ denote a Riemannian structure, where $\mathbb{M}$ is a complete Riemannian manifold endowed with metric $g$. We choose the update iterate $\omega_{k+1}$ on $\mathbb{M}$ of an on-line algorithm as the solution to the local optimisation problem (2), where $D(\omega, \omega_k)$ is a locally convex, smooth, non-negative function with $D(\omega_k, \omega_k) = 0$, $L_\lambda(\omega) = L(\omega, (y_k, x_k))$ is a loss function and $\lambda$ is a positive constant. We term the constant $\lambda > 0$ the learning weight. It can be thought of as trading off between trust in the present estimate ($\lambda$ small) or trust in the new data ($\lambda$ large) when determining the new update.

In recent work [3, 4, 5, 2, 6, 7] the regularisation functions used were based on divergence measures, often derived from the cumulant function associated with an exponential family. In the Riemannian framework considered it is natural to use the Riemannian distance as a regularisation function

$$
D(\omega, \omega_k) = \frac{1}{2} \text{dist}(\omega, \omega_k)^2.
$$

The distance squared is locally convex with respect to the Riemannian geometry, locally smooth, non-negative and $D(\omega_k, \omega_k) = 0$.

A common choice of loss function is the squared error between observation and prediction called the least squares loss. In practice, the least squares loss is specifically adapted to the case of Gaussian measurement noise (cf. Example 1). For general geometric structures, optimisation of the least squares loss may not lead to a good global performance of an estimator.

**Theorem 1** For any generative noise model $y_k = f(\omega, x_k) + \eta_k, \omega \in \mathbb{M} \subset \mathbb{R}^n$, there exists a Riemannian structure $g$ on $\mathbb{M}$ such that the least squares loss is non-convex with respect to $(\mathbb{M}, g)$.

The above theorem is easily verified. It is indicative of the failure of convex optimisation concepts in dealing with the complexities of learning algorithms in non-Euclidean spaces. Indeed, the authors are unaware of a single example of a situation where the least squares loss is convex and there is a (non-trivial) non-linear model class or non-Euclidean geometry. As a consequence, the SG algorithm for arbitrary geometry is prone to poor convergence properties and may very well get “stuck” just as the SG algorithm gets stuck in a flat space with non-convex losses. The least squares loss does have a local validity and is useful in asymptotic analysis of the estimator performance. However, to obtain good global performance of an on-line algorithm it is necessary to consider loss functions intrinsically adapted to the geometry of a given problem.
We propose a loss function termed the geometric loss function

\[
\mathcal{L}_k(\omega) = \mathcal{L}(\omega, (y_k, x_k)) := \frac{1}{2} \min_{\bar{\omega} \mid y_k = f(\bar{\omega}, x_k)} \text{dist}(\omega, \bar{\omega})^2. \quad (4)
\]

To understand the loss function consider Figure 1. Define \( S_k(\alpha) = \{ \omega \in \mathbb{M} \mid |y_k - f(\omega, x_k)| = \alpha \} \subset \mathbb{M} \) for each instance of data \((y_k, x_k)\). The proposed loss function measures the squared geodesic distance from the set \( S_k(0) \). The geometric loss depends only on the geometric structure of the problem and the model class. In this manner it is well adapted to deal with highly non-Euclidean geometric constraints encountered in non-linear generative noise models and non-convex parameter constraints. Moreover, the geometric loss function is well partnered by the regularisation function \( \mathcal{D} \) chosen in the optimisation problem (2). This leads to a range of advantageous properties of the update iterates.

Lemma 2 Let \( \mathbb{M} \) be a convex Riemannian manifold and \((y_k, x_k)\) be an instance of data. If \( \omega \in \mathbb{M} \) is a regular point of the loss function \( \mathcal{L}_k : \mathbb{M} \to \mathbb{R} \) then there exists a unique point \( \bar{\omega}_k \in S_k(0) \) that minimises the distance \( \text{dist}(\omega, \bar{\omega}_k) \). Furthermore, the gradient of \( \mathcal{L}_k \) at \( \omega \) is given by \( \nabla \mathcal{L}_k(\omega) = -2 \text{Exp}_\omega^{-1}(\bar{\omega}_k) \).

The proof of this result is obtained by deriving a variation of distance minimising geodesics to the set \( S_k(0) \), cf.\cite{17, 16}. Lemma 2 confirms that the loss function is minimised along geodesic curves from the set \( S_k(0) \) to the point \( \omega_k \). It follows directly that the minimising geodesic is orthogonal to the set \( S_k(0) \). Furthermore, it is straightforward to show that the minimising geodesics are orthogonal to the level sets \( S_k(\alpha) \) of \( \mathcal{L}_k \) (cf. Figure 1).

The equation for the update iterate is obtained by computing the critical points of the cost function

\[
U_k(\omega) := \mathcal{D}(\omega, \omega_k) + \lambda \mathcal{L}_k(\omega).
\]

Differentiating \( U_k \) and setting to zero to solve for a critical point yields (cf. \cite{17})

\[
dU_k(\omega)(V) = d\mathcal{D}(\omega, \omega_k)(V) + \lambda d\mathcal{L}_k(\omega)(V)
\]

\[
= \langle -\text{Exp}_\omega^{-1}(\omega_k), V \rangle_g + \lambda \langle \nabla \mathcal{L}_k(\omega), V \rangle_g = 0,
\]

for any tangent vector \( V \in T_\omega \mathbb{M} \). Let \( \omega_{k+1} \) be a point of local minimum of \( U_k \) then by the above argument

\[
\text{Exp}_{\omega_{k+1}}(\omega_k) = \lambda \nabla \mathcal{L}_k(\omega_{k+1}). \quad (\text{Implicit Update}) \quad (5)
\]

\footnote{Throughout this paper \( \langle \cdot, \cdot \rangle_g \) denotes the inner product with respect to the Riemannian metric \( g \), \( \nabla \) is the gradient operator extended to Riemannian manifolds, \( \text{Exp}_\omega^{-1}(\omega_k) \) denotes velocity vector of the unique minimising geodesic from \( \omega \) to \( \omega_k \) and \( T_\omega \mathbb{M} \) denotes the tangent space of \( \mathbb{M} \) at \( \omega \).}
This equation is termed the implicit update equation since its solution depends on the evaluation of $\nabla L_k$ at the point $\omega_{k+1}$. An implicit non-linear algebraic relation for the update iterate equation is a feature of other recent work in development of on-line learning algorithms [3, 4, 5, 2, 6, 7]. The accepted approach to overcoming this problem is to approximate the solution of (5) by the solution to the explicit update equation

$$-\text{Exp}_k^{-1}\omega_{k+1} = \mu \nabla L_k(\omega_k). \quad \text{(Explicit Update)} \quad (6)$$

The constant $\mu > 0$ is termed the learning rate and is not equal to the learning weight $\lambda$.

The explicit update is obtained by solving

$$\arg \min_{\omega} \left( D(\omega, \omega_{k+1}) - \mu L_k(\omega) \right) = \omega_k,$$

for $\omega_{k+1} \in M$. The trick is to approximate the local optimisation problem by one in which the regularisation function is defined around the new iterate $\omega_{k+1}$ and the critical value of the new cost is computed at the point $\omega_k$.

**Remark 3** It is interesting to compare the explicit update to the natural gradient algorithm ([12]) $\omega_{k+1} = \omega_k - \mu \nabla L_k(\omega_k)$. The update direction is chosen as the Riemannian gradient direction in both cases. The explicit geometric gradient algorithm uses a geodesic curve for the update calculation (cf. [9]).

A key result for the proposed algorithm is that for suitable choice of learning rate $\mu$ the explicit update is identical to the implicit update.

**Theorem 4** Let $(M, g)$ denote a Riemannian structure associated with a generative noise model $y_k = f(\omega, x_k) + \eta_k$. Let $\omega_k \in M$ be a regular point of the loss function $L_k$ and $\lambda > 0$ a positive constant.

The implicit [5] and explicit [6] updates are equal if and only if $\mu = \lambda/(1 + \lambda)$.

**Corollary 5** For the conditions of Theorem 4 and $\mu = \lambda/(1 + \lambda)$ then

$$L_k(\omega_{k+1}) = (1 - \mu)^2 L_k(\omega_k) = \left( \frac{1}{1 + \lambda} \right)^2 L_k(\omega_k). \quad (7)$$

Furthermore, the limiting cases satisfy

- **$\lambda$ small**: In the case where $\lambda \rightarrow 0$ the algorithm trusts the prior estimate to the exclusion of the new data received. $\lambda = 0$ corresponds to $\mu = 0$ and $\omega_{k+1} = \omega_k$.
- **$\lambda$ large**: In the case where $\lambda \rightarrow \infty$ the algorithm trusts the new data to the exclusion of the prior estimate. The limit $\lambda \rightarrow \infty$ corresponds to $\mu \rightarrow 1$ and $\omega_{k+1} \rightarrow \bar{\omega}_k$, where $\bar{\omega}_k \in S_k(0)$ is the closest point to $\omega_k$ in $S_k(0)$.

Theorem 4 and Corollary 5 provide a clear picture of the performance of the geometric gradient algorithm. The tradeoff between trust in the new data and trust in the previous iterate can be adjusted through choice of the learning weight $\lambda$. The corresponding range for the learning rate $\mu \in [0, 1)$ is confirmed by the limiting conditions given in Corollary 5.

The properties of the implicit and explicit loss function provide the strongest justification of the geometric loss function (7).

**Example 1(b): Linear regression subject to Gaussian noise.** The set $S_k(0)$ is a linear hyper-plane in $M$. For any $\omega_k^0 \in S_k(0)$, $y_k = \langle x_k, \omega_k^0 \rangle$. The shortest distance from $\omega_k \in M$ to $S_k(0)$ (in the Euclidean distance) is given by projecting the difference $(\omega_k - \omega_k^0)$ onto the direction $x_k$. Thus, the proposed geometric regression loss function is
\[ \mathcal{L}_k(\omega_k) = \left\| \frac{x_k x_k^T}{||x_k||^2} (\omega_k - \omega_k^0) \right\|^2 \] 

This is equal to the normalised least squares cost function

\[ \hat{\mathcal{L}}_k(\omega_k) = \left\| \frac{y_k - y_k^0}{||x_k||^2} \right\|^2 = \left\| \frac{x_k x_k^T}{||x_k||^2} (\omega_k - \omega_k^0) \right\|^2 = \mathcal{L}_k(\omega_k). \]

Thus, we see that the proposed geometric loss is equivalent to the normalised least squares cost for linear regression subject to Gaussian noise.

Take \( M = \mathbb{R}^n \) equipped with the standard metric. Then the gradient operator coincides with the differential. Considering the standard Euclidean connection, geodesics in \( \mathbb{R}^n \) are line segments and \( \exp_{\omega_k}^{-1} \omega_{k+1} = \omega_k + 1 - \omega_k \). The geometric gradient algorithm (5) therefore becomes

\[ - (\omega_{k+1} - \omega_k) = \mu \partial_{\omega} \mathcal{L}_k(\omega_k), \]

which is precisely the NSG or Widrow-Hoff learning algorithm.

**Example 2(b): Linear regression over multinomial samples.**

The distance on the simplex \( \Delta^{n-1} \subset \mathbb{R}^n \) is given by

\[ \text{dist}(\omega, \rho) = \frac{2 \sqrt{m} \arccos \left( \sum_{i=1}^{n} \sqrt{\omega_i \rho_i} \right)}{m} \]

minimizing the distance is equivalent to maximizing

\[ \sum_{i=1}^{n} \sqrt{\omega_i \rho_i}, \quad \text{for any} \quad \omega, \rho \in \Delta^{n-1}. \]  

(8)

By utilising Lagrange multipliers one can explicitly determine \( \mathcal{L} \) given by 4 by maximizing (8) subject to the conditions that \( y_k = \sum_{i=1}^{n} \rho^i x_k^i \) and \( \sum_{i=1}^{n} \rho^i = 1. \) It takes the form \( \mathcal{L}_k(\omega) = 4m \arccos^2 \left( 2 (\lambda_1 y_k + \lambda_2) \right) \), where \( \lambda_1 \) and \( \lambda_2 \) must satisfy

\[ \sum_{i=1}^{n} \frac{\omega_i^2}{(\lambda_1 x_k^i + \lambda_2)^2} = 4 \] 

and

\[ \sum_{i=1}^{n} \frac{\omega_i^2 x_k^i}{(\lambda_1 x_k^i + \lambda_2)^2} = 4 y_k, \]

so they depend on \( x_k, y_k \) and \( \omega \).

The gradient of the loss function (4) is determined by

\[ \frac{\partial}{\partial \omega^2} \mathcal{L}_k(\omega) = - \frac{2 \arccos \zeta}{(\lambda_1 y_k + \lambda_2) \sqrt{1 - \zeta^2}}, \]

where \( \zeta = 2(\lambda_1 y_k + \lambda_2) \).

The geometric updates are derived from the geometry of the sphere \( S^{n-1}_{R} \) of radius \( R = 2\sqrt{m} \). It can be shown that the geometric gradient algorithm on \( \Delta^{n-1} \) is given by

\[ \omega_{k+1}^i = \left( \sqrt{\omega_k^i \cos (\mu \frac{||U||}{2m})} - \frac{U^i}{||U||} \sin (\mu \frac{||U||}{2m}) \right)^2, \]

(9)

where the components of \( U \) are \( U^i = \sqrt{\omega_k^i} \left( \frac{\partial}{\partial \omega^j} \mathcal{L}_k(\omega) - \sum_{j=1}^{n} \omega_j \frac{\partial}{\partial \omega^j} \mathcal{L}_k(\omega) \right) \).

\[ \triangleright \]

### 4 Numerical results

We compare the effectiveness of the geometric gradient algorithm appropriate to Example 2, with a modification of the NSG algorithm, appropriate for Example 1 given that \( \omega^* \in \Delta^{n-1} \). The modification is to project the parameter \( \omega_{k+1} \) onto the simplex after each update. Our simulation results demonstrate significant improvement from proper consideration of the geometry, compared to the flat geometry that is appropriate to the Gaussian noise case.

Training data is created according to the stochastic structure of Example 2. The vector \( x_k \in \mathbb{R}^{100} \) is uniformly distributed and the random variable \( z_k \) is drawn according to the multinomial distribution (3), where \( m = 10 \) and \( \omega^* = (0.301, 0.301, 0.301, 0.001, 0.001, \ldots) \).

The calculations in Example 2 were derived for a loss function that did not have the factor of \( 1/2 \) in Equation 7. This minor discrepancy does not alter the results (Figure 2) apart from requiring a modified learning rate \( \bar{\mu} = 2\mu/(1 + \mu) \) where \( \mu \) is the learning rate given in Table (1). The derivation will be corrected in the final draft and the simulations re-tabulated.
Table 1: Stepsizes used and performance achieved for linear regression over multinomial samples example as illustrated in Figure 2

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Stepsize</th>
<th>$\mathcal{L}$</th>
<th>Steps to convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\omega_k = \omega^{*}$</td>
<td>-</td>
<td>0.062</td>
<td>-</td>
</tr>
<tr>
<td>(modified) NSG</td>
<td>$\mu_S = 0.01$</td>
<td>0.071</td>
<td>745</td>
</tr>
<tr>
<td>geometric gradient</td>
<td>$\mu_{G1} = 0.12$</td>
<td>0.064</td>
<td>376</td>
</tr>
<tr>
<td>geometric gradient</td>
<td>$\mu_{G2} = 0.70$</td>
<td>0.071</td>
<td>26</td>
</tr>
</tbody>
</table>

Figure 2: Comparison of NSG and geometric gradient algorithm for linear regression over multinomial samples

The label is then constructed as $y_k = \langle x_k, z_k \rangle$. The initial parameter $\omega_0$ is in the centre of the simplex.

Figure 2 shows the total loss, and $L_2$ parameter error, for one choice of stepsizes for the NSG algorithm and two values of stepsize for the geometric gradient algorithm. In Table 1 we compare the mean square error (MSE) after convergence, $\mathcal{L}$, for each algorithm with the MSE for the true parameter over the same data set. For matching MSE, the geometric gradient algorithm converges significantly faster than the NSG algorithm. Alternatively, the geometric gradient algorithm is able to achieve significantly improved MSE after convergence than the NSG algorithm while still converging faster.

5 Conclusions

We have presented a new general framework for developing and understanding stochastic gradient descent algorithms in a non-Euclidean environment. The framework is developed intrinsically; that is to say it fully respects the underlying geometry. An interesting feature is that the loss function proposed is matched to the geometry (in a manner analogous to
the so-called matching loss functions derived in the Bregman divergence framework). We showed that the explicit and implicit updates are identical, for an appropriate learning rate. We demonstrated the utility of the new framework on a multinomial regression model. The approach also allows the direct derivation of mistake bounds (although that is not reported in the present paper). Our hope is that the new framework will allow the exploitation of the power of Riemannian geometry to the design and analysis of SG algorithms.

Acknowledgement

This work was supported by the Australian Research Council. Thanks to Manfred Warmuth for helpful discussions.

References