Journal of Machine Learning Research 1 (3000) 1–∞ Submitted 0/00; Published 0/00

A Geometric View of Non-Linear On-Line Stochastic Gradient Descent

Krzysztof A. Krakowski KRZYSZTOF.KRAKOWSKI@UNE.EDU.AU
School of Mathematics, Statistics and Computer Science,
The University of New England, Armidale, NSW 2351, Australia
(work performed whilst PostDoc at Department of Engineering, ANU)

Robert E. Mahony ROBERT.MAHONY@ANU.EDU.AU
Department of Engineering,
Australian National University, Canberra, ACT 0200, Australia

Robert C. Williamson BOB.WILLIAMSON@ANU.EDU.AU
National ICT Australia Ltd (NICTA), and
Research School of Information Sciences and Engineering,
Australian National University, Canberra, ACT 2601, Australia

Manfred K. Warmuth MANFRED@CSE.UCSC.EDU
Department of Computer Science, University of California,
Santa Cruz, CA 95064, USA (work performed whilst at NICTA)

Editor: The Editor

Abstract

We investigate a relationship between geometric structure of a parameter space and on-line learning algorithms. The geometric structure is often implied by a nature of a problem to be solved. For example, the parameters one wishes to learn are a subject to constrains, or they are intrinsically non-linear objects or they are distributed in a non uniform, but known way. We develop a mistake bound analysis for a two new geometric algorithms and show that link functions based algorithms are essentially the transformed geometric algorithms.

Keywords: machine learning, on-line algorithms, differential geometry, link functions

1. Introduction

We are interested in the *Widrow-Hoff perceptron* algorithm (*cf.* Clarkson, 1993) given by

$$w_{t+1} = w_t - \eta \, \boldsymbol{\partial}_w \mathcal{L}(w_t, (x_t, y_t)). \tag{1}$$

Here $\eta > 0$ is a finite positive *learning rate*, w_t represents the current estimate of a vector that parameterizes the family of models considered, $\mathcal{L}$ is a convex *loss function* that quantifies the measurement error at each sample (x_t, y_t) and $\boldsymbol{\partial}_w$ is the vector of partial derivatives $\partial/\partial w^i$.

In the nineties some interesting variants of this algorithm, including the *Exponentiated Gradient* (EG) algorithm have been studied. Kivinen and Warmuth (1998) developed a mistake bounded framework that showed that the EG algorithm has better performance in situations, where the target weight vector is sparse. The intuition here is that, given prior knowledge about the problem considered, it is possible to tune a learning algorithm to improve performance. Kivinen and Warmuth (1997) introduced the concept of "link functions" in order to generalise the derivation of the

EG algorithm. This structure has lead naturally to the use of Bregman divergence in analysing the performance of these algorithms (Jagota and Warmuth, 1998). If the prior assumption is incorrect, it is expected that the performance of a "tuned algorithm" is inferior to an unmodified algorithm. Following early work in this area, a range of gradient descent algorithms inspired by the EG algorithm have been analysed for both, classification and regression problems (Grove et al., 1997; Jagota and Warmuth, 1998; Kivinen and Warmuth, 1997, 2001; Gentile and Littlestone, 1999; Gordon, 1999a,b). In Mahony and Williamson (2001) a geometric analysis of the EG algorithm has provided independent of the link function framework of Kivinen and Warmuth (1997) and Jagota and Warmuth (1998). The key aspect of this work was the introduction of a preferential structure, a Riemannian metric introduced to the space of weight vectors that allowed one to incorporate prior information in the derivation of an algorithm. It is of interest to note that this structure is of a different origin to the Information Geometry introduced by Amari (1998) for natural gradient learning. The preferential structure introduced by Mahony and Williamson (2001) does not depend on the generative noise model and is equally valid for deterministic learning problems.

In this paper we are concerned with geometry of on-line learning algorithms and continue to develop the geometric foundations of Mahony and Williamson (2001). We recognise that the space of the parameter w often has geometrical properties. For example, if w is a subject to the condition $\|w\| = 1$ then it is more natural to consider *the unit sphere* as the parameter space instead of the whole $\mathbb{R}^n$. If we have a prior knowledge that certain regions in parameter space are more likely to provide the optimal parameter vector $w_\star$ than other regions, then it is reasonable to incorporate this information into an on-line learning algorithm to improve its performance. Thus we enrich the parameter space and endow it with a geometric structure that reflects the nature of that space, parameter constraints, the prior knowledge.

In this paper we derive two geometric on-line algorithms: the *implicit* and *explicit* update rules, that naturally extend the stochastic gradient descent to geometric spaces. We analyse efficiencies of these two algorithms through the framework of relative mistake bound. A number of examples presented in this paper exhibit various forms of the two geometric algorithms in spaces like Euclidean, the unit sphere and the probability simplex. We show that the explicit rule in Euclidean space is identical to the perceptron algorithm (1).

The geometric approach undertaken by the authors offers a systematic study and classification of some known on-line learning algorithms. We show that the implicit and explicit algorithms are indeed geometrically intrinsic, i.e., they are isometrically invariant. If there exists an isometry, a distance preserving diffeomorphism between two parameter spaces, then the isometry transforms geometric algorithm in one space to essentially the same algorithm in the other space. Furthermore we convey the concept of equivalent algorithms, where a diffeomorphism maps steps of an algorithm in one parameter space to steps of the equivalent algorithm in another space. The significance of the equivalence of algorithms under isometries is that the mistake bound analysis of the two geometric algorithms derived in this paper is also isometrically invariant.

This paper is organised as follows. Section 2 introduces the notation and concepts of Riemannian geometry used throughout this paper. Here we review definitions of the Riemannian structure, the distance, geodesic and the exponential map. In Section 3 we derive the two geometric online algorithms, the explicit and implicit update rules. A number of examples in this section illustrates how these algorithms can be applied to different spaces. Here we note on equivalence of geometric algorithms. The mistake bounds framework is than developed and applied to the explicit and im-

plicit algorithm in Sections 4 and 5, respectively. Finally Section 6 establishes relationship between the explicit update and link functions based algorithms.

2. Riemannian Manifolds

This section introduces geometric terminology used throughout this paper. We give here informal definitions and fundamental facts of differential geometry. For readers not familiar with this subject, the authors suggest the two excellent modern expositions to Riemannian geometry, notably Lee (1997) and Petersen (1998).

Manifold $\mathbf{M}$ is a set of points such that for every point $p \in \mathbf{M}$, there exists an open neighbourhood $\mathcal{V} \subset \mathbf{M}$ of p diffeomorphic to some subset of $\mathbb{R}^n$. For example, the circle or any plain closed curve without self-intersections are 1-manifolds, although none of them is diffeomorphic to $\mathbb{R}$. To every point $p \in \mathbf{M}$ there is attached an n-vector space $\mathcal{T}_p\mathbf{M}$ called a *tangent space*. The disjoin union of the tangent spaces, $\mathcal{T}\mathbf{M} = \coprod_{p\in\mathbf{M}} \mathcal{T}_p\mathbf{M}$ is called a *tangent bundle*. Every tangent space $\mathcal{T}_p\mathbf{M}$ has assigned a bilinear positive definite form $g(p)$, a *metric*, defining an inner product $\langle U, V\rangle_g = g(p)(U, V)$, where $U, V \in \mathcal{T}_p\mathbf{M}$ are vectors in the same tangent space. The norm of a tangent vector V is defined as usual by $\|V\|_g = \sqrt{\langle V, V\rangle_g}$. If the metric g varies smoothly with respect to points in $\mathbf{M}$ then the pair $(\mathbf{M}, g)$ is called a *Riemannian structure*.

Let $I \subset \mathbb{R}$ be an interval of the real line then a continuous mapping $\gamma\colon I \to \mathbf{M}$ is called a *curve*. If γ is differentiable then the velocity vector, i.e., its derivative $\dot\gamma(s) = \frac{d}{dt}\Big|_{t=s} \gamma(t)$, is a tangent vector to γ at $\gamma(s)$. The set of such velocity vector forms the tangent space $\mathcal{T}_{\gamma(s)}\mathbf{M}$, justifying the name *tangent*. The norm of the velocity vector $\|\dot\gamma(s)\|_g$ is often called a *speed*. The length of a (piecewise) differentiable curve γ is given by the integral $\int_I \|\dot\gamma(s)\|_g \, ds$. Suppose that manifold $\mathbf{M}$ is connected. Then the distance $\operatorname{dist}(p, q)$ between any two points $p, q \in \mathbf{M}$ is the infimum of the lengths of all curves from p to q. In Riemannian geometry there are special curves of particular interest, *geodesics*. The Riemannian geodesics are characterised as the curves whose accelaration is zero and so they have constant speed. Geodesics have the following important property, they are, locally, the shortest curves. We have to add *locally* because, as in the case of a circle, it may happen that there is more than one geodesic joining a pair of points. A geodesic whose length is equal to the distance of its end points is called (distance) *minimising*. Geodesics are uniquely defined by an initial point $p = \gamma(0) \in \mathbf{M}$ and initial velocity $V = \dot\gamma(0) \in \mathcal{T}_p\mathbf{M}$. One defines the *exponential map* $\operatorname{Exp}_p\colon \mathcal{T}_p\mathbf{M} \to \mathbf{M}$ that assigns the end point of the geodesic $\gamma\colon [0, 1] \to \mathbf{M}$ starting at p to the vector $\dot\gamma(0)$.

In this paper we keep calculations of geometric obects to minimum and we often use properties of these objects in order to avoid tedious calculations in any paricular coordnate system. However, when it is necessary to refer to coordinates we follow the convention: vector components, function components and coordinates are denoted with superscripts, matrix entries are denoted with subscripts. Other indices appear as subscripts. For example w_t^i is the ith component of w_t and g_{ij} is the ijth entry of matrix (or tensor) g. To indicate that we refer to a tensor in a fixed basis we write $(g_{ij}(w))$. Vectors are denoted with uppercase letters and points with lower case.

3. The Geometrically Intrinsic On-Line Algorithms

This section derives two geometric on-line algorithms, the *implicit* and *explicit* updates. The two algorithms arise naturally with an introduction of a distance based regularisation term to the loss function. Examples of these two algorithms in $\mathbb{R}^n$, the unit sphere and probability simplex conclude this section.

3.1 On-line learning algorithms for parametric inference

In studying learning problems we consider the three components. Firstly, the *sampling process* $(x_t, y_t) = \Sigma(t)$ generates data points $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$. Typically, the sampling process is derived from a generative noise model

$$y_t = F(x_t) + \mu, \tag{2}$$

where μ is a given noise process and $F: \mathcal{X} \to \mathcal{Y}$ is an unknown function. The input x_t may itself be a realisation of a stochastic process on $\mathcal{X}$ or a deterministic process provided by the user. In this paper we do not make any assumptions about the sampling process. An objective of the parametric inference is to find the best fitted function from a *parameterised class* of functions $\widehat{F}_w: \mathcal{X} \to \mathcal{Y}$. The parameter $w \in \mathbf{M}$ is called the *weight vector* and we assume here that the set $\mathbf{M}$ is a manifold. Thus the parametrised model class provides the relation (identification) $\mathbf{M} \leftrightarrow \mathcal{C}(\mathcal{X}, \mathcal{Y})$ given by

$$w \leftrightarrow \widehat{F}_w: \mathcal{X} \to \mathcal{Y} \quad \text{and} \quad \hat{y} = \widehat{F}_w(x).$$

The goodness of fitting a model function to data is measured by a *loss function*, a continuously differentiable function $\mathcal{L}: \mathbf{M} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. Often, the loss function is required to be convex with respect to its first argument.

The most common learning problem involves a Gaussian generative noise model, the class of linear predictors as a parametrised model class

$$\widehat{F}_w(x) := \langle w, x \rangle, \tag{3}$$

where $w \in \mathbb{R}^n$, $x \in \mathbb{R}^n = \mathcal{X}$, $y \in \mathbb{R} = \mathcal{Y}$ and $\langle w, x \rangle = \sum_{i=1}^n w^i x^i = w^{\mathrm{T}} x$ denotes the standard vector inner product, and the squared error loss function

$$\mathcal{L}(w, (x, y)) = (y - \hat{y})^2 = \left(y - \widehat{F}_w(x)\right)^2 = (y - \langle w, x \rangle)^2. \tag{4}$$

The goal of a (parametric) on-line learning algorithm is to progressively refine an estimate $\widehat{F}_{w_t}$, for $t = 0, 1, \ldots, k$, to minimize the expected loss $\mathcal{L}(w_t, (x_t, y_t))$. A new parameter w_{t+1} is derived from the last estimated parameter w_t and the current observation (x_t, y_t). The previous observations are not used. Formally, an on-line learning algorithm may be expressed as a mapping $\mathfrak{A}: \mathbf{M} \times \mathcal{X} \times \mathcal{Y} \to \mathbf{M}$, with the update rule of the form

$$w_{t+1} = \mathfrak{A}(w_t, (x_t, y_t)).$$

Remark 1 *In principle, it is possible to formulate the on-line learning problem without requiring that the parameter set* $\mathbf{M}$ *is a manifold or indeed that the loss function is differentiable. Since on-line learning algorithms are concerned with small updates of the weight parameters it is natural to wish to apply an infinitesimal analysis.*

An example of an on-line learning algorithm is the Widrow-Hoff algorithm (1). However, the existing framework of the Widrow-Hoff algorithm does not explicitly take into account any prior knowledge. To improve performance, prior knowledge should be incorporated into an on-line learning algorithm. There are two important methods in the literature that address this issue.

Link function: The framework of link functions introduced to machine learning by Kivinen and Warmuth (1997) provides a reparametrization $\theta = f(w)$ of the original parameters w. The reparametrization $f: \mathbb{R}^n \to \mathbb{R}^n$ is a smooth invertible mapping of $\mathbb{R}^n$ called a *link function*. Thus, one considers a new class of approximating functions

$$\widehat{F}_\theta(x) := \left\langle f^{-1}(\theta), x \right\rangle .$$

By working with the new parametrization θ the underlying distribution of parameters across functional models $\widehat{F}_w(x)$ is altered. The Widrow-Hoff algorithm with respect to the new parametrization may be written (Jagota and Warmuth, 1998) as

$$f(w_{t+1}) = f(w_t) - \eta\, Df(w_t)^{-\mathrm{T}} \boldsymbol{\partial} \mathcal{L}_t(w_t), \tag{5}$$

where Df is the Jacobian of f, $Df(w_t)^{-\mathrm{T}}$ is its inverse transposed and $\mathcal{L}_t(w)$ denotes the loss function, given sample (x_t, y_t), i.e., $\mathcal{L}_t(w) := \mathcal{L}(w, (x_t, y_t))$.

Preferential structure: The framework of a preferential structure was introduced by the authors in Mahony and Williamson (2001). In this approach an adjustment of the parameter space is accomplished by introducing a Riemannian metric g called a *preferential structure* on the parameter space $w \in \mathbb{R}^n$. The metric $(g_{ij}(w))$ is an $n \times n$ positive definite matrix chosen such that

$$\sqrt{\det(g_{ij}(w))} = \phi(w),$$

where $\phi(w)$ is a Bayesian prior on parameter space $w \in \mathbb{R}^n$. This relationship is chosen in order that volume with respect to the new metric structure is equivalent to prior density. The stochastic gradient descent algorithm expressed with respect to the preferential structure is

$$w_{t+1} = \mathrm{Exp}_{w_t}(-\eta\, \mathrm{grad} \mathcal{L}_t(w_t)), \tag{6}$$

where Exp is the exponential map.

Remark 2 *Neither of the approaches mentioned above are directly related to the information geometry underlying the natural gradient algorithms introduced by Amari (1998). However, the underlying motivation for considering a reparametrization or alteration of the model parametrization is analogous to the motivation in considering the information geometric interpretation of on-linear learning algorithms. Note that in the natural gradient algorithm proposed by Amari (1998) the underlying assumption made is that the prior distribution is maximally non-informative with respect to the geometric structure induced by the noise model.*

A key concept in the design and analysis of on-line learning algorithms is that only a small change in parameter estimate is made at any one time step. Typically, there is a small positive constant at each step of the algorithm (η), called the *learning rate* or *step-size*, that limits the change in w_t. In noisy environments the step-size is chosen small to limit the effects of noise in disturbing the averaged convergence properties of the algorithm. In less noisy environments the step-size can be chosen larger.

3.2 Derivation of the geometric algorithms

This section derives geometrically intrinsic stochastic gradient descent algorithm for a parametrised model class $\mathbf{M}$, that is endowed with Riemannian metric g. The metric g determines geometric distance between parameters and in effect a comparison of parametrised functions $\widehat{F}_w$ within the same class. To see this consider a class of linear functions $\widehat{F}_w(x) = \langle w, x\rangle + c$ then the expectation

$$\mathbb{E}_x[(\widehat{F}_{w_1} - \widehat{F}_{w_2})^2] = \mathbb{E}_x[(\langle w_1, x\rangle - \langle w_2, x\rangle)^2] = (w_1 - w_2)^{\mathrm{T}}\, \mathbb{E}_x[x\, x^{\mathrm{T}}]\, (w_1 - w_2)$$

can be considered to be the squared norm $\left\|\widehat{F}_{w_1} - \widehat{F}_{w_2}\right\|^2$. For a normally distributed $x \sim \mathcal{N}(0, \Sigma)$, the expectation $\mathbb{E}_x[(\widehat{F}_{w_1} - \widehat{F}_{w_2})^2]$ becomes $\|w_1 - w_2\|^2_\Sigma$, if the covariance matrix Σ is positive-definite. When Σ is the identity matrix $\mathbf{I}$, then the expectation becomes the square of the distance between w_1 and w_2 in $\mathbb{R}^n$.

The most natural choice of an on-line learning algorithm is the stochastic gradient descent learning algorithm (1). To simplify the notation, for given a sample (x_t, y_t) indexed by t, let the loss function $\mathcal{L}(w, (x_t, y_t))$ be denoted by $\mathcal{L}_t(w)$. Then $\mathcal{L}_t \colon \mathbf{M} \to \mathbb{R}$ becomes a function on $\mathbf{M}$ and (1) can be written as

$$w_{t+1} = w_t - \eta\, \boldsymbol{\partial}\mathcal{L}_t(w_t).$$

Derivation of the geometric algorithms is based on the assumption that the new weight w_{t+1} produced by an on-line algorithm from the sample (x_t, y_t) and the previous weight w_t should minimize the current *instantaneous loss* $\mathcal{L}_t$, and at the same time be close to w_t. Let $\mathcal{D} \colon \mathbf{M} \times \mathbf{M} \to \mathbb{R}$ denote a measure of closeness of points in $\mathbf{M}$ then we set w_{t+1} to be a critical point to the new cost function $U_t \colon \mathbf{M} \to \mathbb{R}$ given by

$$U_t(w) := \mathcal{D}(w, w_t) + \eta\, \mathcal{L}_t(w). \tag{7}$$

Parameter $\eta > 0$, the *learning rate*, controls relative importance of the terms in U_t. In machine learning applications (Jagota and Warmuth, 1998; Lafferty et al., 1997) the *regularisation term* is the divergence. However, divergence is not a geometrically intrinsic object because it depends on a choice of coordinates in $\mathbf{M}$ (Kass and Vos, 1997). We note here that the "Preferred Point Geometry" (Critchley et al., 1994) makes it possible to introduce the Kullback-Leibler divergence into Riemannian manifolds of varying metric. Such approach however, goes beyond the scope of this exposition. For our purposes, we take $\mathcal{D}$ to be a half of the square of the geometric distance, namely

$$\mathcal{D}(w_1, w_2) := \frac{1}{2}\, \mathrm{dist}(w_1, w_2)^2, \quad \text{for any} \quad w_1, w_2 \in \mathbf{M}, \tag{8}$$

where $\mathrm{dist}(w_1, w_2)$ is the Riemannian distance between w_1 and w_2. $\mathcal{D}$ is symmetric and differentiable but in general *it is not* convex. Convexity of $\mathcal{D}$ depends on geometry of $\mathbf{M}$, both the curvature of the metric g and diameter of $\mathbf{M}$. In spaces of non-positive curvature, for example Euclidean space $\mathbb{R}^n$ and hyperbolic spaces $\mathbb{H}^n$, $\mathcal{D}$ is convex everywhere. In spaces of positive curvature, say, the unit sphere $\mathbf{S}^n$, $\mathcal{D}$ is convex only when points are close (less than $\pi/2$ apart for the unit sphere).

The modified cost U_t given by (7) is differentiable and if it is strictly convex then it has a unique critical point $w \in \mathbf{M}$, where U_t attains its minimum. Suppose that w is a critical point of U_t then the differential dU_t at w is zero, that is $dU_t(w)(V) = 0$, for any tangent vector $V \in \mathcal{T}_w\mathbf{M}$, where

$$\begin{aligned} dU_t(w)(V) &= d\mathcal{D}(w, w_t)(V) + \eta\, d\mathcal{L}_t(w)(V) \\ &= \left\langle -\operatorname{Exp}_w^{-1} w_t, V \right\rangle_g + \eta \left\langle \operatorname{grad}\mathcal{L}_t(w), V \right\rangle_g, \end{aligned}$$

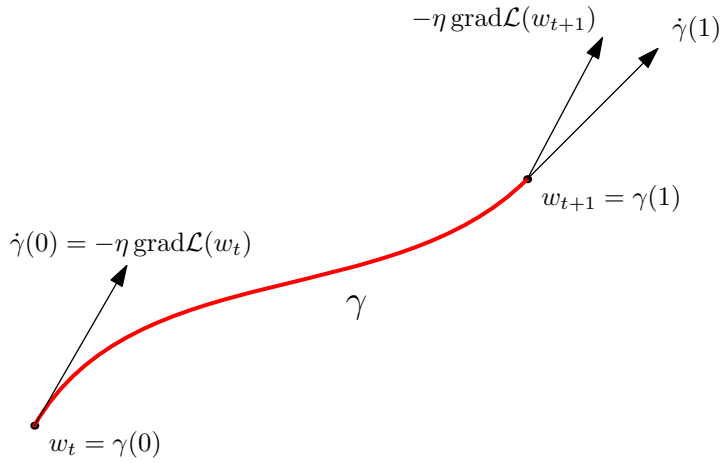

Figure 1: Step update: w_{t+1} is the endpoint of the geodesic $\gamma : [0,1] \to \mathbf{M}$, starting at w_t with initial velocity $-\eta\,\mathrm{grad}\mathcal{L}_t(w_t)$. The end velocity $\dot\gamma(1)$ is generally different from $-\eta\,\mathrm{grad}\mathcal{L}_t(w_{t+1})$.

because $d\mathcal{D}(w,w_t) = -\operatorname{Exp}_w^{-1} w_t$, where $\operatorname{Exp}_w^{-1} w_t$ denotes the velocity vector of the geodesic from w to w_t evaluated at w, (*cf.* Karcher, 1977). Therefore the necessary condition for w_{t+1} to be a point of local minimum of U_t is given by

$$\operatorname{Exp}_{w_{t+1}}^{-1} w_t = \eta\,\mathrm{grad}\mathcal{L}_t(w_{t+1}) \tag{9}$$

This equality however, has the unknown w_{t+1} on both sides of the equation. More importantly, condition (9) does not necessarily define w_{t+1} uniquely. It is often more convenient to work with an approximation to the point of minimum of U_t, where one takes

$$-\operatorname{Exp}_{w_t}^{-1} w_{t+1} = \eta\,\mathrm{grad}\mathcal{L}_t(w_t), \tag{10}$$

which is precisely the stochastic gradient descent algorithm (6). Formula (10), the equality of two vectors in the tangent space $\mathcal{T}_{w_t}\mathbf{M}$, states that w_{t+1} is the end point of a geodesic segment starting at w_t whose initial velocity is equal to $-\eta\,\mathrm{grad}\mathcal{L}_t(w_t)$. The two update rules (9) and (10) are generally different because the velocity vector of the geodesic at the end point w_{t+1} is different than the gradient of the loss at this point, Figure 1. The length of the geodesic segment and therefore the distance $\mathrm{dist}(w_t, w_{t+1})$ is equal to $\eta\,\|\mathrm{grad}\mathcal{L}_t(w_{t+1})\|_g$, for the *implicit rule* (9), and equal to $\eta\,\|\mathrm{grad}\mathcal{L}_t(w_t)\|_g$, for the *explicit rule* (10), where the norm $\|V\|_g = \langle V, V\rangle_g^{1/2}$ is taken with respect to the Riemannian metric g on $\mathbf{M}$.

We begin our discussion of the two geometric algorithms (9) and (10) in Euclidean space. The following example illustrates a relationship between the two learning algorithms and the Widrow-Hoff perceptron.

Example 1 (Euclidean space $\mathbf{M} = \mathbb{R}^n$) *The geometric gradient* grad *in* $\mathbb{R}^n$ *is equal to the standard gradient* ∂*. Considering the standard Euclidean connection, we find that geodesics*

in $\mathbb{R}^n$ are line segments and the velocity vector $\operatorname{Exp}_{w_{t+1}}^{-1} w_t = w_t - w_{t+1}$. Hence the implicit update (9) becomes

$$w_{t+1} = w_t - \eta\, \partial \mathcal{L}_t(w_{t+1}).$$

Similarly $\operatorname{Exp}_{w_t}^{-1} w_{t+1} = w_{t+1} - w_t$ and the explicit update (10) becomes

$$w_{t+1} = w_t - \eta\, \partial \mathcal{L}_t(w_t),$$

which is precisely the Widrow-Hoff learning algorithm (1). The two algorithm (9) and (10) are not the same as they differ in a way they calculate gradients of the loss. The former algorithm takes the "future gradient" evaluated at the new point w_{t+1} while the latter one takes the gradient at the initial point w_t. ▷

Next example derives the geometric explicit algorithms (10) in the case when $\mathbf{M}$ is the unit sphere $\mathbf{S}^n$. When $n = 2$ it is a standard sphere that serves as an intuitive example of a non-flat space. The sphere is a highly symmetric space that has constant sectional curvature equal to one. It will become apparent, and we will state it more precisely at the end of this section, that the two spaces, $\mathbb{R}^n$ and $\mathbf{S}^n$, induce fundamentally different algorithms.

Example 2 (The unit sphere $\mathbf{M} = \mathbf{S}^n$) *Consider the explicit update (10) in the unit n-sphere $\mathbf{S}^n$ equipped with its standard metric induced by the Euclidean metric $\bar{g}$ on $\mathbb{R}^{n+1}$. The exponential map on the unit sphere is given by*

$$\operatorname{Exp}_w(V) = w\, \cos \|V\| + \frac{V}{\|V\|} \sin \|V\|\,,$$

for any $w \in \mathbf{S}^n$ and $V \in \mathcal{T}_w\mathbf{S}^n$, where the norm is taken with respect to the Euclidean metric $\bar{g}$ of the ambient space $\mathbb{R}^{n+1}$. Given the exponential map, the learning algorithm (10) on the unit sphere $\mathbf{S}^n$ can be written as follows

$$w_{t+1} = \operatorname{Exp}_{w_t}\left(-\eta \operatorname{grad}\mathcal{L}_t(w_t)\right) = w_t\, \cos\left(\eta \|V\|\right) - \frac{V}{\|V\|} \sin\left(\eta \|V\|\right),$$

where $V = \operatorname{grad}\mathcal{L}_t(w_t)$ is the projection of the standard gradient $\partial \mathcal{L}_t$ onto the tangent plane $\mathcal{T}_{w_t}\mathbf{S}^n$, i.e., $\operatorname{grad}\mathcal{L}_t(w_t) = \partial \mathcal{L}_t(w_t) - w_t \langle \partial \mathcal{L}_t(w_t), w_t \rangle$. ▷

Examples 1 and 2 illustrate two forms of the geometric explicit algorithm (10) in two different geometric spaces. We shall now derive the formula for the same algorithm in yet another space. Namely, the probability simplex Δ^n. Here, the simplex Δ^n is endowed with the spherical metric arising from the Fisher Information metric of the multinomial distribution (Kass and Vos, 1997). Since geometry of the simplex is not that common in the literature, we include the calculations in detail. For another application of the simplex geometry in machine learning the authors refer the reader to Lebanon (2005).

Example 3 (The probability simplex $\mathbf{M} = \Delta^n$) *Let Δ^n denote the (open) n-simplex defined by*

$$\Delta^n = \left\{ p \in \mathbb{R}^{n+1} \;:\; \sum_{i=1}^{n+1} p^i = 1 \quad \text{and} \quad p^i > 0 \right\}.$$

If $\gamma \colon I \to \Delta^n$ is a smooth curve in the simplex then by differentiating the above constraint

$$\frac{d}{dt}\left(\sum_{i=1}^{n+1} \gamma^i(t)\right) = \sum_{i=1}^{n+1} \dot{\gamma}^i(t) = 0,$$

hence one sees that vectors in the tangent bundle $\mathcal{T}\Delta^n$ share the following property

$$\sum_{i=1}^{n+1} V^i = 0, \quad \textit{for any} \quad V \in \mathcal{T}\Delta^n.$$

Set $f\colon \Delta^n \to \mathbf{S}^n_2$ to be a map from the simplex to the positive orthant of the n-sphere of radius 2, whose components in standard coordinates on $\mathbb{R}^{n+1}$ are given by

$$f^i(p) = 2\sqrt{p^i}, \quad \textit{for} \quad i = 1, 2, \ldots, n+1.$$

The map f is a diffeomorphism inducing spherical geometry on the simplex Δ^n with the metric $g = f^*\bar{g}$. Therefore f is an isometry and the following diagram commutes

$$\begin{array}{ccc} \mathcal{T}_p\Delta^n & \xrightarrow{f_*} & \mathcal{T}_{f(p)}\mathbf{S}^n_2 \\ {\scriptstyle \mathrm{Exp}_p}\downarrow & & \downarrow{\scriptstyle \mathrm{Exp}_{f(p)}} \\ \Delta^n & \xrightarrow[f]{} & \mathbf{S}^n_2 \end{array}$$

which allows us to re-use calculations for the sphere in Example 2. Denote $w = f(p)$ then the explicit algorithm in Δ^n is transformed to $\mathbf{S}^n_2$, the n-sphere of radius 2, according to the above diagram, as follows

$$w_{t+1} = f(p_{t+1}) = f(\mathrm{Exp}_{p_t} V) = \mathrm{Exp}_{f(p_t)} f_* V = \mathrm{Exp}_{w_t} f_* V, \tag{11}$$

where vector $V \in \mathcal{T}_p\Delta^n$ is equal to $-\eta\,\mathrm{grad}\mathcal{L}_t(p_t)$ and the gradient $\mathrm{grad}\mathcal{L}_t$ is evaluated at $p_t \in \Delta^n$. To derive an explicit form of (11) we need to find the push-forward $f_*\mathrm{grad}\mathcal{L}_t$ of the gradient of the loss function in local coordinates. Here we use another property of isometries, namely that isometries preserve gradient. This is expected, since gradients are vector fields. To see that this indeed is the case can be reasoned as follows. By the definition of the gradient, for any vector $V \in \mathcal{T}_p\Delta^n$ there is $\langle \mathrm{grad}\mathcal{L}_t(p), V\rangle_g = d\mathcal{L}_t(p)(V)$, where $d\mathcal{L}_t$ is a 1-form. By the properties of pullback maps on functions (cf. Guillemin and Pollack, 1974) one formally writes

$$f^* d\widetilde{\mathcal{L}}_t = d(f^*\widetilde{\mathcal{L}}_t) = d(\widetilde{\mathcal{L}}_t \circ f) = d\mathcal{L}_t, \quad \textit{where} \quad \widetilde{\mathcal{L}}_t = \mathcal{L}_t \circ f^{-1}.$$

Let $\widetilde{V} = f_* V$ then since $g = f^*\bar{g}$ there is

$$\begin{aligned} \bar{g}(w)(\mathrm{grad}\widetilde{\mathcal{L}}_t(w), \widetilde{V}) &= f^* d\widetilde{\mathcal{L}}_t(\widetilde{V}) = d\mathcal{L}_t(V) = g(p)(\mathrm{grad}\mathcal{L}_t(p), V) \\ &= \bar{g}(f(p))(f_*\mathrm{grad}\mathcal{L}_t(p), f_* V) = \bar{g}(w)(f_*\mathrm{grad}\mathcal{L}_t(p), \widetilde{V}). \end{aligned}$$

Hence $f_*\mathrm{grad}\mathcal{L}_t = \mathrm{grad}\widetilde{\mathcal{L}}_t$, as expected. The spherical gradient $\mathrm{grad}\widetilde{\mathcal{L}}_t$ is equal to the projection of the gradient in $\mathbb{R}^{n+1}$ onto the tangent plane $\mathcal{T}_w\mathbf{S}^n_2$, namely

$$\left(\mathrm{grad}\widetilde{\mathcal{L}}_t(w)\right)^i = \partial_i \widetilde{\mathcal{L}}_t(w) - \frac{w^i}{4}\left\langle \boldsymbol{\partial}\widetilde{\mathcal{L}}_t(w), w\right\rangle.$$

It is now straightforward to derive the push-forward $f_*\mathrm{grad}\mathcal{L}_t$ in standard coordinates

$$\left(f_*\mathrm{grad}\mathcal{L}_t(f(p))\right)^i = \left(\mathrm{grad}\widetilde{\mathcal{L}}_t(w)\right)^i = \sqrt{p^i}\left(\partial_i\mathcal{L}_t - \sum_{j=1}^{n+1} p^j \partial_j \mathcal{L}_t\right).$$

We are now ready to write the explicit update in the probability simplex. By (11) the explicit algorithm (10) in Δ^n is equal to the transformed algorithm in $\mathbf{S}^n_2$ (cf. Example 2) and therefore it is given by

$$\sqrt{p^i_{t+1}} = \sqrt{p^i_t}\cos\left(\frac{\eta}{2}\|V\|\right) - \frac{V^i}{\|V\|}\sin\left(\frac{\eta}{2}\|V\|\right), \quad \textit{where} \quad V = f_*\mathrm{grad}\mathcal{L}_t(f(p)).$$

Further calculation shows that

$$\|V\|^2 = \sum_{i=1}^{n+1} p^i \left(\partial_i \mathcal{L}_t(p)\right)^2 - \left(\sum_{i=1}^{n+1} p^i \partial_i \mathcal{L}_t(p)\right)^2 .$$

▷

3.3 Equivalence of learning algorithms

In this paper we are interested in studying and comparing different classes of on-line learning algorithms. In order that we can speak of learning algorithms as essentially the same it is necessary that we can define their equivalence.

Definition 3 *Let* $\mathfrak{A}_1 \colon \mathbf{M}_1 \times \mathcal{X} \times \mathcal{Y} \to \mathbf{M}_1$ *and* $\mathfrak{A}_2 \colon \mathbf{M}_2 \times \mathcal{X} \times \mathcal{Y} \to \mathbf{M}_2$ *be two learning algorithms in two parameter spaces* $\mathbf{M}_1$ *and* $\mathbf{M}_2$*. Algorithms* $\mathfrak{A}_1$ *and* $\mathfrak{A}_2$ *are said to be* equivalent *if*

i) there exists a diffeomorphism $\varphi \colon \mathbf{M}_1 \to \mathbf{M}_2$*; and*

ii) for all $t = 0, \ldots, k$

$$\varphi(\mathfrak{A}_1(w_t, (x_t, y_t))) = \mathfrak{A}_2(\varphi(w_t), (x_t, y_t)).$$

The equivalence relation $\mathfrak{A}_1 \sim \mathfrak{A}_2$ indicates that $\mathfrak{A}_1$ and $\mathfrak{A}_2$ yield essentially the same iterates. We have the following property of the geometric algorithms.

Lemma 4 *Let* $\varphi \colon \mathbf{M}_1 \to \mathbf{M}_2$ *be an isometry between the two parameter spaces* $\mathbf{M}_1$ *and* $\mathbf{M}_2$*, and* $\mathcal{L}$ *and* $\widetilde{\mathcal{L}}$ *be two loss functions such that*

$$\widetilde{\mathcal{L}}(\varphi(w), (x_t, y_t)) = \mathcal{L}(w, (x_t, y_t)), \quad \textit{for all} \quad t = 0, \ldots, k, \quad \textit{and} \quad w \in \mathbf{M}_1.$$

Then the geometric implicit (explicit) rule derives equivalent algorithms.

Proof The proof follows the same reasoning as the one of Example (3). It is enough to show that the gradients transform properly. The equivalence will then follow from uniqueness of geodesics and the following commutative diagram

$$\begin{array}{ccc} \mathcal{T}_w \mathbf{M}_1 & \xrightarrow{\varphi_*} & \mathcal{T}_{\varphi(w)} \mathbf{M}_2 \\ {\scriptstyle \mathrm{Exp}_w}\downarrow & & \downarrow{\scriptstyle \mathrm{Exp}_{\varphi(w)}} \\ \mathbf{M}_1 & \xrightarrow[\varphi]{} & \mathbf{M}_2 \end{array}$$

Since φ is an isometry then $\varphi_* \mathrm{grad} \mathcal{L}_t = \mathrm{grad} \widetilde{\mathcal{L}}_t$, *cf.* Example 3. The result follows now directly by writing down the above diagram, *cf.* (11). Let $\widetilde{w}_t = \varphi(w_t)$ and $\widetilde{w}_{t+1} = \varphi(w_{t+1})$ then

$$\begin{aligned} \varphi(w_{t+1}) &= \varphi(\mathrm{Exp}_{w_t}(-\eta \,\mathrm{grad} \mathcal{L}_t(w_t))) = \mathrm{Exp}_{\widetilde{w}_t}(\varphi_*(-\eta \,\mathrm{grad} \mathcal{L}_t(w_t))) \\ &= \mathrm{Exp}_{\widetilde{w}_t}(-\eta \,\varphi_* \mathrm{grad} \mathcal{L}_t(w_t)) = \mathrm{Exp}_{\widetilde{w}_t}(-\eta \,\mathrm{grad} \widetilde{\mathcal{L}}_t(\widetilde{w}_t)) = \widetilde{w}_{t+1}. \end{aligned}$$

This proves equivalence of the explicit algorithm (10) in $\mathbf{M}_1$ and $\mathbf{M}_2$. The proof for the implicit algorithm (9) is exactly the same. ■

As a direct consequence of Lemma 4 one has, suspected then and obvious now, the result concerning the geometric explicit algorithms derived in Examples 2 and 3.

Corollary 5 *The two explicit algorithms, in the positive orthant of the sphere $\mathbf{S}^n_2$ of radius 2 (Example 2) and in the open simplex Δ^n (Example 3) are equivalent.*

The geometric framework allows identification of learning algorithms that are essentially the same. It has other implications as well. In the Information Geometry exponential families of parametric probability distributions induce manifolds equipped with Fisher metrics (*cf.* Kass and Vos, 1997). In fact there are two isometric spaces associated with every family: the space of *natural parameters* and the space of *expectation parameters*. Isometries between the natural and expectation spaces are provided by the *canonical link functions*. In the machine learning problems of density estimation one can employ the explicit (10) or implicit (9) algorithm in one or another space. The algorithms and their analysis will be exactly the same, independent of the choice of space, because they are equivalent.

4. Mistake Bound for the Explicit Update

Consider on-line learning methods for parameter inference. Here the model is repeatedly updated at every step of an algorithm. The update is a response of the algorithm to an error incurred by a mismatch between the current model and new data. The error is quantified by a loss function and the sum of these errors over a number of iterations is called the *total loss*. The total loss is an indication of how well an algorithm performs but because of the fluctuation of the data, it is not indicative by itself. Instead, one compares the total loss of an algorithm with a total loss for a fixed model using the same data. The difference between these two total losses is called the *relative loss*.

In this section we derive the upper bound for the relative loss of the explicit update (10) in Riemannian spaces of non-negative sectional curvature. The spaces considered so far: Euclidean space $\mathbb{R}^n$, the unit sphere $\mathbf{S}^n$ and the probability simplex Δ^n, are examples of such spaces.

Before we state the main result of this section recall the following definitions. A function f on a Riemannian manifold is *convex* if for any geodesic γ the composition $f \circ \gamma$ is convex. In situations when the assumption of convexity of a loss function is too restrictive, it is useful to consider λ-convex functions instead. A unit speed geodesic $\gamma\colon [0, l] \to \mathbf{M}$, where l is the length of γ, is said to be *parametrised by arc length*. A function f is called λ-*convex* if for any geodesic γ, parametrised by arc length, the function $(f \circ \gamma)(s) - \lambda s^2$ is convex.

Convexity of the loss function bounds its second derivative, the Hessian, from below, and consequently the total loss accumulated on a sequence of input data $\{(x_t, y_t)\}_{t=0}^{k-1} \subset (\mathcal{X} \times \mathcal{Y})^k$. This is the essence of the following result.

Theorem 6 *Let $(\mathbf{M}, g)$ be a complete manifold of non-negative sectional curvature and the measure $\mathcal{D}\colon \mathbf{M} \times \mathbf{M} \to \mathbb{R}$ be given by (8). For a convex loss function $\mathcal{L}\colon \mathbf{M} \to \mathbb{R}$ and any fixed $w_\star \in \mathbf{M}$ we have the following upper bound on the cumulative loss of the Widrow-Hoff algorithm (10)*

$$\sum_{t=0}^{k-1} \mathcal{L}_t(w_t) \leq \sum_{t=0}^{k-1} \mathcal{L}_t(w_\star) + \frac{1}{\eta}\left(\mathcal{D}(w_\star, w_0) - \mathcal{D}(w_\star, w_k)\right) + \frac{1}{\eta}\sum_{t=0}^{k-1} \mathcal{D}(w_t, w_{t+1}). \tag{12}$$

In particular, for all $0 \leq t \leq k-1$ and geodesic γ_t from w_t to $w_\star$, if the loss function $\mathcal{L}_t|_{\gamma_t}$ restricted to the image of γ_t is λ_t-convex, then

$$\sum_{t=0}^{k-1} \mathcal{L}_t(w_t) \leq \sum_{t=0}^{k-1} \mathcal{L}_t(w_\star) + \frac{1}{\eta}\left(\mathcal{D}(w_\star, w_0) - \mathcal{D}(w_\star, w_k)\right)$$

$$+\frac{1}{\eta}\sum_{t=0}^{k-1}\mathcal{D}(w_t,w_{t+1})-2\sum_{t=0}^{k-1}\lambda_t\,\mathcal{D}(w_t,w_\star). \tag{13}$$

Proof Let $\gamma\colon[0,1]\to\mathbf{M}$ be a geodesic from w_t to $w_\star$ and $\sigma\colon[0,1]\to\mathbf{M}$ be a geodesic from w_t to w_{t+1}. Consider a geodesic hinge with vertices $w_{t+1}, w_t, w_\star$ and an angle α at w_t. Denote lengths of γ and σ by $\ell(\gamma)$ and $\ell(\sigma)$, respectively. If $\mathbf{M}=\mathbb{R}^n$ is Euclidean space endowed with its standard metric then by the law of cosines

$$\operatorname{dist}(w_{t+1},w_\star)^2=\ell(\gamma)^2+\ell(\sigma)^2-2\,\ell(\gamma)\,\ell(\sigma)\cos\alpha.$$

In general, since sectional curvature of g on $\mathbf{M}$ is non-negative, then by the Toponogov hinge theorem (*cf.* Petersen, 1998)

$$\operatorname{dist}(w_{t+1},w_\star)^2\le\ell(\gamma)^2+\ell(\sigma)^2-2\left\langle\dot\sigma(0),\dot\gamma(0)\right\rangle.$$

Denote $l=\ell(\gamma)=\operatorname{dist}(w_t,w_\star)$ and let $\tilde\gamma(s)=\gamma(s/l)$ then $\tilde\gamma$ is a geodesic parametrised by arc length. Define a function $f\colon[0,l]\to\mathbb{R}$ by $f(s)=\mathcal{L}_t(\tilde\gamma(s))-\lambda_t s^2$. By the hypothesis $\mathcal{L}_t|_\gamma$ is λ_t-convex (take $\lambda_t=0$ if $\mathcal{L}$ is convex) hence f is convex and therefore $f(l)\ge f(0)+l\,f'(0)$. As a consequence one has the following.

$$\begin{aligned}\mathcal{L}_t(w_\star)-\lambda_t\,l^2 &\ge \mathcal{L}_t(\tilde\gamma(0))+l\,\left\langle\operatorname{grad}\mathcal{L}_t(\tilde\gamma(0)),\dot{\tilde\gamma}(0)\right\rangle\\ &= \mathcal{L}_t(w_t)-\frac{1}{\eta}\left\langle\dot\sigma(0),\dot\gamma(0)\right\rangle\\ &\ge \mathcal{L}_t(w_t)+\frac{1}{\eta}\left(\mathcal{D}(w_\star,w_{t+1})-\mathcal{D}(w_\star,w_t)-\mathcal{D}(w_t,w_{t+1})\right).\end{aligned}$$

Summing over $t=0,1,\dots,k-1$ brings

$$\begin{aligned}\sum_{t=0}^{k-1}\left(\mathcal{L}_t(w_\star)-\lambda_t\operatorname{dist}(w_t,w_\star)^2\right)&=\sum_{t=0}^{k-1}\mathcal{L}_t(w_\star)-2\sum_{t=0}^{k-1}\lambda_t\,\mathcal{D}(w_t,w_\star)\\ &\ge \sum_{t=0}^{k-1}\mathcal{L}_t(w_t)+\frac{1}{\eta}\sum_{t=0}^{k-1}\left(\mathcal{D}(w_\star,w_{t+1})-\mathcal{D}(w_\star,w_t)-\mathcal{D}(w_t,w_{t+1})\right)\\ &= \sum_{t=0}^{k-1}\mathcal{L}_t(w_t)+\frac{1}{\eta}\left(\mathcal{D}(w_\star,w_k)-\mathcal{D}(w_\star,w_0)\right)-\frac{1}{\eta}\sum_{t=0}^{k-1}\mathcal{D}(w_t,w_{t+1}).\end{aligned}$$

This completes the proof. ■

Inequalities (12) and (13) establish an upper bound for mistakes made by geometric explicit algorithm (10). This bound depends on $\mathcal{D}$ and therefore on the Riemannian distance. Hence it make sense to ask what geometric structure $(\mathbf{M},g)$ generates the most efficient algorithm. Careful look at the bounds of Theorem 6 gives a clue. The last term in (12) can be interpreted as follows. Let ℓ_t denote the Riemannian distance $\operatorname{dist}(w_t,w_{t+1})$, which is the length of the geodesic segment $\gamma\colon[t,t+1]\to\mathbf{M}$, then $\mathcal{D}(w_t,w_{t+1})=\frac{1}{2}\ell_t^2$ and the last term in (12) can be now written as

$$\sum_{t=0}^{k-1}\mathcal{D}(w_t,w_{t+1})=\frac{1}{2}\sum_{t=0}^{k-1}\ell_t^2=\frac{k}{2}\sum_{t=0}^{k-1}\frac{1}{k}\ell_t^2\ge\frac{k}{2}\left(\frac{1}{k}\sum_{t=0}^{k-1}\ell_t\right)^2=\frac{1}{2k}\,\ell^2(\gamma)\ge\frac{1}{k}\mathcal{D}(w_0,w_k),$$

where we used the Jensen's inequality and the triangle inequality for the Riemannian distance. In the above expression the equality holds if and only if

i) all ℓ_t are equal (the first inequality), and

ii) γ is the distance minimizing curve (the second inequality).

The above conditions *i)* and *ii)* give the best possible (lowest) upper bounds. This in turn, tells us that the best algorithm incurring the smallest total loss, is induced by such a Riemannian structure of the parameter space so that the two above conditions are not *far off*. That is, the steps sizes have similar lengths and the learning curve is close to a minimising geodesic, for example a straight line in $\mathbb{R}^n$.

The following example illustrates how Theorem 6 can be applied to the gradient descent algorithm in $\mathbb{R}^n$. To obtain better (more accurate) bound on the relative loss one has to derive the λs for a specific loss function. In the case of $\mathbb{R}^n$ and the square loss, the calculations are quite simple. In other spaces, however, a suitable candidate for a loss function is less obvious.

Example 4 (The gradient descent algorithm) *Taking parameter space* $\mathbf{M}$ *to be* $\mathbb{R}^n$ *the algorithm (10) becomes* $w_{t+1} = w_t - \eta\, \partial_w \mathcal{L}_t(w_t)$*, which is the standard gradient descent algorithm. We shall derive an upper bound for the total loss of this algorithm from Theorem 6, inequality (13).*

Let the loss function $\mathcal{L}$ *be the quadratic function* $\mathcal{L}_t(w) = (\langle w, x_t\rangle - y_t)^2$ *and let us fix* $w_\star \in \mathbf{M}$*. Let* $\gamma_t\colon [0, \ell_t] \to \mathbb{R}^n$ *be the line segment given by* $s \mapsto w_t + s\,(w_\star - w_t)\,/\ell_t$*, where* $\ell_t = \|w_\star - w_t\|$ *is the distance between* $w_\star$ *and* w_t *in* $\mathbb{R}^n$*. Then* $\mathcal{L}_t|_{\gamma_t}$*, the restriction to the image of* γ_t*, is* λ_t*-convex, where* $\lambda_t = \langle w_\star - w_t, x_t\rangle^2 / \ell_t^2$*. It is easy to check that* $\mathcal{L}_t|_l$ *is* λ_t*-convex for any line segment* l *contained in the image of* γ_t*. Since the distance* $\operatorname{dist}(w_t, w_{t+1})$ *is proportional to the gradient of the loss function* $\mathcal{L}_t$ *at* w_t *then*

$$\mathcal{D}(w_t, w_{t+1}) = \frac{1}{2}\operatorname{dist}(w_t, w_{t+1})^2 = \frac{1}{2}\left(2\eta\left(\langle w_t, x_t\rangle - y_t\right)\right)^2 \langle x_t, x_t\rangle = 2\eta^2 \mathcal{L}_t(w_t)\ \|x_t\|_2^2$$

and

$$\begin{aligned} 2\lambda_t\, \mathcal{D}(w_t, w_\star) &= \langle w_\star - w_t, x_t\rangle^2 = \left(\left(\langle w_\star, x_t\rangle - y_t\right) - \left(\langle w_t, x_t\rangle - y_t\right)\right)^2 \\ &= \mathcal{L}_t(w_\star) + \mathcal{L}_t(w_t) - 2\sqrt{\mathcal{L}_t(w_\star)\,\mathcal{L}_t(w_t)}. \end{aligned}$$

Dropping the negative term $-\mathcal{D}(w_\star, w_k)$ *from (13) and using the above calculations yields*

$$\begin{aligned} \sum_{t=0}^{k-1} \mathcal{L}_t(w_t) \quad \leq \quad & \sum_{t=0}^{k-1} \mathcal{L}_t(w_\star) + \frac{1}{\eta}\frac{1}{2}\ \|w_\star - w_0\|^2 + \frac{1}{\eta}\, 2\eta^2 \sum_{t=0}^{k-1} \mathcal{L}_t(w_t)\ \|x_t\|^2 \\ & - \left(\sum_{t=0}^{k-1} \mathcal{L}_t(w_\star) + \sum_{t=0}^{k-1} \mathcal{L}_t(w_t) - 2\sum_{t=0}^{k-1} \sqrt{\mathcal{L}_t(w_\star)\,\mathcal{L}_t(w_t)}\right). \end{aligned}$$

Let $X_2 \geq \|x_t\|$*, for all* $0 \leq t \leq k-1$*, then by the Cauchy-Schwarz inequality*

$$\sum_{t=0}^{k-1} \mathcal{L}_t(w_t) \leq \frac{1}{4\eta}\|w_\star - w_0\|^2 + \eta X_2^2 \sum_{t=0}^{k-1} \mathcal{L}_t(w_t) + \left(\sum_{t=0}^{k-1} \mathcal{L}_t(w_\star)\right)^{1/2} \left(\sum_{t=0}^{k-1} \mathcal{L}_t(w_t)\right)^{1/2}.$$

Let $\mathrm{Loss}(\mathfrak{A})$ *and* $\mathrm{Loss}(w_\star)$ *denote the cumulative loss of* $\mathcal{L}_t(w_t)$ *and* $\mathcal{L}_t(w_\star)$*, respectively, cf. (14). To be more precise, we define*

$$\mathrm{Loss}(\mathfrak{A}) := \sum_{t=0}^{k-1} \mathcal{L}_t(w_t) \quad \textit{and} \quad \mathrm{Loss}(w_\star) := \sum_{t=0}^{k-1} \mathcal{L}_t(w_\star), \tag{14}$$

for a given data sequence $\{(x_0, y_0), (x_1, y_1), \ldots, (x_{k-1}, y_{k-1})\}$*. Furthermore, let the starting weight be* $w_0 = 0$ *then the above inequality can be written as*

$$\left(1 - \eta X_2^2\right) \mathrm{Loss}(\mathfrak{A}) - \left(\mathrm{Loss}(w_\star)\right)^{1/2} \left(\mathrm{Loss}(\mathfrak{A})\right)^{1/2} \leq \frac{1}{4\eta} \|w_\star\|^2$$

which, provided $\eta X_2^2 < 1$*, leads in effect to quadratic inequality which implies*

$$\left(\mathrm{Loss}(\mathfrak{A})\right)^{1/2} \leq \frac{\left(\mathrm{Loss}(w_\star)\right)^{1/2} + \sqrt{\mathrm{Loss}(w_\star) + \|w_\star\|^2 \left(1 - \eta X_2^2\right)/\eta}}{2\left(1 - \eta X_2^2\right)}.$$

When $\mathrm{Loss}(w_\star) < \|w_\star\|^2 X_2^2$ *then the right hand side of the above inequality attains its only minimum*

$$\|w_\star\| X_2 + \left(\mathrm{Loss}(w_\star)\right)^{1/2} \quad \textit{at} \quad \eta = \frac{\|w_\star\|}{2X_2\left(\|w_\star\| X_2 + \sqrt{\mathrm{Loss}(w_\star)}\right)}. \tag{15}$$

From (15) it is seen that in an ideal situation when $\mathrm{Loss}(w_\star) = 0$ *the optimal learning rate is* $\eta = \left(2X_2^2\right)^{-1}$ *and the total loss of the algorithm satisfies* $\mathrm{Loss}(\mathfrak{A}) \leq \|w_\star\|^2 X_2^2$*.* ▷

The upper bound for an online algorithm gives an indirect way of investigating its behaviour. The natural question arises about convergence properties of the algorithm. In a generic situation, without making any assumptions for the input sequence, we can say very little about the convergence. What we can say is that, under certain assumptions, one can choose small enough fixed learning rate $\eta > 0$, such that there always is $\mathcal{L}_t(w_{t+1}) \leq \mathcal{L}_t(w_t)$, for all $0 \leq t \leq k-1$. This holds for any differentiable loss function $\mathcal{L}$, not necessarily convex.

Recall that the manifold $\mathbf{M}$ is *complete* if the exponential map $\mathrm{Exp}_p(V)$ is defined for all $p \in \mathbf{M}$ and all vectors $V \in \mathcal{T}_p\mathbf{M}$. By the Hopf-Rinow theorem (*cf.* Lee, 1997) a connected manifold is complete if and only if it is complete as a metric space. For example, a sphere $\mathbf{S}_r^n$ and the probability simplex Δ^n are complete manifolds, but any proper open subset of $\mathbb{R}^n$ with Euclidean metric is not.

Proposition 7 *Let* $\mathbf{M}$ *be a complete manifold and* $\mathcal{L}\colon \mathbf{M} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ *be a continuously differentiable function in respect to the first argument, with a finite number of isolated critical points. Then there exists* $\xi > 0$ *such that for any learning rate* $0 < \eta < \xi$

$$\mathcal{L}_t(w_{t+1}) \leq \mathcal{L}_t(w_t), \quad \textit{for any} \quad 0 \leq t \leq k-1,$$

where w_{t+1} *is given by (10). The equality holds if and only if* $w_{t+1} = w_t$ *is a critical point of* $\mathcal{L}$*.*

Proof Let $\gamma_V\colon [0, c] \to \mathbf{M}$ be a geodesic with initial point $w_t = \gamma_V(0)$ and initial velocity vector $V = \dot{\gamma}_V(0) = -\eta\, \mathrm{grad}\mathcal{L}_t(w_t)$, and let $f = \mathcal{L}_t \circ \gamma_V$. Then by the chain rule

$$\begin{aligned} f'(0) &= \langle \mathrm{grad}\mathcal{L}_t(\gamma_V(0)), \dot{\gamma}_V(0)\rangle \\ &= \langle \mathrm{grad}\mathcal{L}_t(w_t), -\eta\, \mathrm{grad}\mathcal{L}_t(w_t)\rangle = -\eta\, \|\mathrm{grad}\mathcal{L}_t(w_t)\|^2 \leq 0, \end{aligned}$$

where equality holds if and only if w_t is a critical point of $\mathcal{L}_t$, in which case (10) gives $w_{t+1} = w_t$. If w_t is not a critical point of $\mathcal{L}_t$ then, since f' is continuous, f is monotonically decreasing in some interval $[0, \xi_t)$. Since the number of isolated critical points of f is finite then $\xi_t > 0$. Henceforth, for any $0 < \eta < \xi_t$ there is $f(\eta) < f(0)$. By the rescaling lemma (*cf.* Lee, 1997) and by (10) it follows that $\gamma_V(\eta) = \gamma_{\eta V}(1) = w_{t+1}$. Hence $\mathcal{L}_t(w_{t+1}) < \mathcal{L}_t(w_t)$, for any $0 < \eta < \xi_t$. We choose $\xi \leq \xi_t$, for all $t = 0, 1, \ldots, k-1$. Since the number of steps k is finite then $\xi > 0$ and the result follows. ■

The above property of the geometric gradient descent method guarantees that, for a suitable small learning rate, the explicit algorithm will always step towards a minimum of a loss function. This fact is independent of the input samples (x_t, y_t). The convexity of the loss function is not necessary.

5. Mistake Bound for the Implicit Update

In the previous section we studied total loss bounds of the geometric explicit algorithm (10). This section completes the investigation of the relative mistake bounds, where we now consider the *implicit update* (9). The implicit algorithm (9) takes the updated parameter to be a critical point of the modified loss U_t whereas the explicit algorithm uses the approximation (10) of a critical point. This is the fundamental difference between the two rules and this difference is reflected in derivation of the total mistake bound. The following analog of Theorem 6 establishes bound for a total loss for spaces of non-positive sectional curvature. Examples of such spaces include Euclidean $\mathbb{R}^n$ and the hyperbolic space $\mathbb{H}^n$.

Theorem 8 *Let $(\mathbf{M}, g)$ be a complete manifold of non-positive sectional curvature and the measure $\mathcal{D}\colon \mathbf{M} \times \mathbf{M} \to \mathbb{R}$ be given by (8). For a convex loss function $\mathcal{L}\colon \mathbf{M} \to \mathbb{R}$ and any fixed $w_\star \in \mathbf{M}$ we have the following upper bound on the cumulative loss of the implicit algorithm (9)*

$$\sum_{t=0}^{k-1} \mathcal{L}_t(w_t) \leq \sum_{t=0}^{k-1} \mathcal{L}_t(w_\star) + \frac{1}{\eta} \left(\mathcal{D}(w_\star, w_0) - \mathcal{D}(w_\star, w_k)\right) - \frac{1}{\eta} \sum_{t=0}^{k-1} \mathcal{D}(w_t, w_{t+1}). \tag{16}$$

In particular, for all $0 \leq t \leq k$ and a geodesic γ_{t+1} from w_{t+1} to $w_\star$, if the loss function $\mathcal{L}_t|_{\gamma_{t+1}}$ restricted to the image of γ_{t+1} is λ_t-convex, then

$$\begin{aligned} \sum_{t=0}^{k-1} \mathcal{L}_t(w_t) \leq \sum_{t=0}^{k-1} \mathcal{L}_t(w_\star) + \frac{1}{\eta} \left(\mathcal{D}(w_\star, w_0) - \mathcal{D}(w_\star, w_k)\right) \\ - \frac{1}{\eta} \sum_{t=0}^{k-1} \mathcal{D}(w_t, w_{t+1}) - 2 \sum_{t=0}^{k-1} \lambda_t \, \mathcal{D}(w_t, w_\star). \end{aligned} \tag{17}$$

Proof The proof is almost identical to the proof of Theorem 6, where we consider the geodesic hinge with vertices $w_t, w_{t+1}, w_\star$ and an angle α at w_{t+1}, and the two geodesics: $\gamma\colon [0,1] \to \mathbf{M}$ from w_{t+1} to $w_\star$, and $\sigma\colon [0,1] \to \mathbf{M}$ from w_{t+1} to w_t.

Denote $l = \ell(\gamma) = \operatorname{dist}(w_{t+1}, w_\star)$ and let $\tilde{\gamma}(s) = \gamma(s/l)$ then $\tilde{\gamma}$ is a geodesic parametrised by arc length. Define a function $f\colon [0, l] \to \mathbb{R}$ by $f(s) = \mathcal{L}_t(\tilde{\gamma}(s)) - \lambda_t s^2$. Since by the hypothesis

$\mathcal{L}_t|_\gamma$ is λ_t-convex (take $\lambda_t = 0$ if $\mathcal{L}$ is convex) it follows that f is convex and therefore $f(l) \geq f(0) + l\, f'(0)$. As a consequence one has

$$\begin{aligned}
\mathcal{L}_t(w_\star) - \lambda_t\, l^2 &\geq \mathcal{L}_t(\tilde{\gamma}(0)) + l\, \left\langle \mathrm{grad}\mathcal{L}_t(\tilde{\gamma}(0)), \dot{\tilde{\gamma}}(0) \right\rangle \\
&= \mathcal{L}_t(w_{t+1}) + \langle \mathrm{grad}\mathcal{L}_t(w_{t+1}), \dot{\gamma}(0) \rangle \\
&= \mathcal{L}_t(w_t) + \frac{1}{\eta} \langle \dot{\sigma}(0), \dot{\gamma}(0) \rangle \\
&\geq \mathcal{L}_t(w_t) + \frac{1}{\eta} \left(\mathcal{D}(w_\star, w_{t+1}) + \mathcal{D}(w_{t+1}, w_t) - \mathcal{D}(w_\star, w_t) \right).
\end{aligned}$$

Summing over $t = 0, 1, \ldots, k-1$ yields

$$\begin{aligned}
\sum_{t=0}^{k-1} \left(\mathcal{L}_t(w_\star) - \lambda_t\, \mathrm{dist}(w_{t+1}, w_\star)^2 \right) &= \sum_{t=0}^{k-1} \mathcal{L}_t(w_\star) - 2 \sum_{t=0}^{k-1} \lambda_t\, \mathcal{D}(w_{t+1}, w_\star) \\
&\geq \sum_{t=0}^{k-1} \mathcal{L}_t(w_t) + \frac{1}{\eta} \sum_{t=0}^{k-1} \left(\mathcal{D}(w_\star, w_{t+1}) - \mathcal{D}(w_\star, w_t) + \mathcal{D}(w_t, w_{t+1}) \right) \\
&= \sum_{t=0}^{k-1} \mathcal{L}_t(w_t) + \frac{1}{\eta} \left(\mathcal{D}(w_\star, w_k) - \mathcal{D}(w_\star, w_0) \right) + \frac{1}{\eta} \sum_{t=0}^{k-1} \mathcal{D}(w_t, w_{t+1}).
\end{aligned}$$

The proof is now complete. ■

We shall now derive a bound for total loss for the implicit version of the gradient descent algorithm in $\mathbb{R}^n$, *cf.* Example 4. However, before we do that the following property of the geometric implicit update will be needed.

Proposition 9 *If $\mathcal{L}_t$ is convex and w_{t+1} is derived from the* implicit update *(9) then the following inequality holds*

$$\mathcal{D}(w_t, w_{t+1}) \leq \frac{\eta}{2} \left(\mathcal{L}_t(w_t) - \mathcal{L}_t(w_{t+1}) \right). \tag{18}$$

Proof Let $\gamma\colon [0,1] \to \mathbf{M}$ be a geodesic from w_{t+1} to w_t and $f\colon [0,1] \to \mathbb{R}$ be the composition $f = \mathcal{L}_t \circ \gamma$. Since by the hypothesis f is convex then $f(1) \geq f(0) + f'(0)$ therefore

$$\begin{aligned}
\mathcal{L}_t(\gamma(1)) &= \mathcal{L}_t(w_t) \geq \mathcal{L}_t(\gamma(0)) + \langle \mathrm{grad}\mathcal{L}_t(\gamma(0)), \dot{\gamma}(0) \rangle \\
&= \mathcal{L}_t(w_{t+1}) + \langle \mathrm{grad}\mathcal{L}_t(w_{t+1}), \dot{\gamma}(0) \rangle = \mathcal{L}_t(w_{t+1}) + \frac{1}{\eta} \langle \dot{\gamma}(0), \dot{\gamma}(0) \rangle \\
&= \mathcal{L}_t(w_{t+1}) + \frac{1}{\eta}\, \mathrm{dist}(w_t, w_{t+1})^2 = \mathcal{L}_t(w_{t+1}) + \frac{2}{\eta}\, \mathcal{D}(w_t, w_{t+1}),
\end{aligned}$$

where we used the update rule (9). Rearranging the terms completes the proof. ■

As a consequence of Proposition 9 we see that the implicit update (9) ensures that the cost incurred by U_t in a derived point is not greater than at the original point.

$$\begin{aligned}
U_t(w_{t+1}) &= \mathcal{D}(w_{t+1}, w_t) + \eta \mathcal{L}_t(w_{t+1}) \\
&\leq \mathcal{D}(w_{t+1}, w_t) + \eta \mathcal{L}_t(w_t) - 2\mathcal{D}(w_t, w_{t+1}) \\
&= \eta \mathcal{L}_t(w_t) - \mathcal{D}(w_{t+1}, w_t) = U_t(w_t) - \mathcal{D}(w_{t+1}, w_t).
\end{aligned}$$

Example 5 (The implicit gradient descent algorithm in Euclidean space) *Take* $\mathbf{M} = \mathbb{R}^n$ *with the standard Euclidean metric. Choose the loss function* $\mathcal{L}$ *be the same as in Example 4, that is* $\mathcal{L}_t(w) = (\langle w, x_t\rangle - y_t)^2$. *For any fixed parameter* $w_\star \in \mathbb{R}^n$, *let* $\ell_t = \|w_\star - w_t\|$ *denote the distance between* $w_\star$ *and* w_t *and* $\gamma_t\colon [0, \ell_t] \to \mathbb{R}^n$ *be the line segment from* w_t *to* $w_\star$ *given by* $s \mapsto w_t + s\,(w_\star - w_t)\,/\ell_t$. *As seen in Example 4 the loss function* $\mathcal{L}_t|_{\gamma_{t+1}}$ *is* λ_t*-convex, where now* $\lambda_t = \langle w_\star - w_{t+1}, x_t\rangle^2 / \ell_{t+1}^2$. *Since the distance* $\operatorname{dist}(w_t, w_{t+1})$ *is proportional to the gradient of the loss function* $\mathcal{L}_t$ *a* w_{t+1} *then*

$$\mathcal{D}(w_t, w_{t+1}) = \frac{1}{2}\operatorname{dist}(w_t, w_{t+1})^2 = \frac{1}{2}\left(2\eta\left(\langle w_{t+1}, x_t\rangle - y_t\right)\right)^2 \langle x_t, x_t\rangle = 2\eta^2 \mathcal{L}_t(w_{t+1})\ \|x_t\|^2$$

and

$$\begin{aligned} 2\lambda_t\,\mathcal{D}(w_{t+1}, w_\star) &= \langle w_\star - w_{t+1}, x_t\rangle^2 = \left(\left(\langle w_\star, x_t\rangle - y_t\right) - \left(\langle w_{t+1}, x_t\rangle - y_t\right)\right)^2 \\ &= \mathcal{L}_t(w_\star) + \mathcal{L}_t(w_{t+1}) - 2\sqrt{\mathcal{L}_t(w_\star)\,\mathcal{L}_t(w_{t+1})}. \end{aligned}$$

Dropping the negative term $-\mathcal{D}(w_\star, w_k)$ *from (17) and using the above calculations yields*

$$\begin{aligned} \sum_{t=0}^{k-1}\mathcal{L}_t(w_t) &\le \sum_{t=0}^{k-1}\mathcal{L}_t(w_\star) - \sum_{t=0}^{k-1}\left(\mathcal{L}_t(w_\star) + \mathcal{L}_t(w_{t+1}) - 2\sqrt{\mathcal{L}_t(w_\star)\,\mathcal{L}_t(w_{t+1})}\right) \\ &\quad + \frac{1}{2\eta}\ \|w_\star - w_0\|^2 - 2\eta\sum_{t=0}^{k-1}\mathcal{L}_t(w_{t+1})\ \|x_t\|_2^2 \\ &= \frac{1}{2\eta}\ \|w_\star - w_0\|^2 - \sum_{t=0}^{k-1}\left(1 + 2\eta\,\|x_t\|_2^2\right)\mathcal{L}_t(w_{t+1}) + 2\sum_{t=0}^{k-1}\sqrt{\mathcal{L}_t(w_\star)\,\mathcal{L}_t(w_{t+1})}. \end{aligned}$$

Let $X_2 \le \|x_t\|$, *for all* $0 \le t \le k-1$, *and let* $\operatorname{Loss}(\mathfrak{A})$, $\operatorname{Loss}(\mathfrak{A}+)$ *and* $\operatorname{Loss}(w_\star)$ *denote cumulative losses of* $\mathcal{L}_t(w_t)$, $\mathcal{L}_t(w_{t+1})$ *and* $\mathcal{L}_t(w_\star)$, *respectively, cf. (14). Then by the Cauchy-Schwarz inequality*

$$\operatorname{Loss}(\mathfrak{A}) \le \frac{1}{2\eta}\ \|w_\star - w_0\|^2 - \left(1 + 2\eta X_2^2\right)\operatorname{Loss}(\mathfrak{A}+) + 2\left(\operatorname{Loss}(w_\star)\right)^{1/2}\left(\operatorname{Loss}(\mathfrak{A}+)\right)^{1/2}. \tag{19}$$

By (18) it follows that $0 \le \mathcal{L}_t(w_{t+1}) \le \mathcal{L}_t(w_t)/(1 + 4\eta\,\|x_t\|_2^2)$, *for all* $0 \le t \le k-1$, *hence*

$$\operatorname{Loss}(\mathfrak{A}+) = \sum_{t=0}^{k-1}\mathcal{L}_t(w_{t+1}) \le \sum_{t=0}^{k-1}\frac{\mathcal{L}_t(w_t)}{1 + 4\eta\,\|x_t\|^2} \le \frac{1}{1 + 4\eta X_2^2}\operatorname{Loss}(\mathfrak{A}).$$

Plugging the above inequality to (19) yields

$$\left(1 + 4\eta X_2^2\right)\operatorname{Loss}(\mathfrak{A}+) \le \frac{1}{2\eta}\ \|w_\star - w_0\|^2 - \left(1 + 2\eta X_2^2\right)\operatorname{Loss}(\mathfrak{A}+) + 2\left(\operatorname{Loss}(w_\star)\right)^{1/2}\left(\operatorname{Loss}(\mathfrak{A}+)\right)^{1/2}.$$

Take $w_0 = 0$, *then by solving the above inequality one derives the following bound for* $\operatorname{Loss}(\mathfrak{A}+)$

$$\left(\operatorname{Loss}(\mathfrak{A}+)\right)^{1/2} \le \frac{\|w_\star\|^2}{2\eta\left(\sqrt{\operatorname{Loss}(w_\star) + \|w_\star\|^2\ (1 + 3\eta X^2)\,/\eta} - \sqrt{\operatorname{Loss}(w_\star)}\right)}$$

If $\operatorname{Loss}(w_\star) = 0$ *then simply*

$$\left(\operatorname{Loss}(\mathfrak{A}+)\right)^{1/2} \le \frac{\|w_\star\|}{2\sqrt{\eta\,(1 + 3\eta X^2)}}.$$

▷

6. Link Functions

This section relates the geometric explicit algorithm (10) to the *general additive algorithm*, sometimes also called the *reparameterised gradient descent*, studied in Jagota and Warmuth (1998); Kivinen and Warmuth (1997). We compare the upper bound for the relative total loss result (Theorem 6) with the results of Kivinen and Warmuth (2001) and Cesa-Bianchi (1999) mistake bounds for the gradient descent (GD) algorithms in the case of a general (not necessarily quadratic) loss function.

Let the space of parameters $\mathbf{M}$ be a complete Riemannian n-manifold endowed with a metric g and $f\colon \mathbf{M} \to \mathbb{R}^n$ be a diffeomorphism from $\mathbf{M}$ to $\mathbb{R}^n$. Mapping f provides a reparametrization $\theta = f(w)$ of the weights from $\mathbf{M}$ to Euclidean space and is called a *link function*. Diffeomorphism f is an isometry, i.e., it preserves distance, if and only if $g = f^*\bar{g}$ is the pullback metric induced by $\bar{g}$, where $\bar{g}$ is the standard Euclidean metric on $\mathbb{R}^n$. In terms of a local coordinate system, the matrix of the pullback metric g is given by $(g_{ij}) = Df^{\mathrm{T}} \cdot Df$, where Df is the Jacobian of f, Df^{T} denotes the transpose of Df and '$\cdot$' is the matrix multiplication.

6.1 Geometry of link functions

In this section we fully characterise the relationship between the link function and preferential structure reformulation of the explicit on-line learning algorithm (10). We have the following result.

Proposition 10 *Consider an on-line learning problem with generative noise model (2), data sequence $\{(x_t, y_t)\}$ and linear predictive model class (3). Let $f\colon \mathbf{M} \to \mathbb{R}^n$ be a smooth invertible link function and let g be a preferential structure on $\mathbf{M}$. Then the link transformed Widrow-Hoff learning algorithm (5) is equivalent to the preferential stochastic gradient descent algorithm (6), for all $\eta > 0$ step-sizes and all loss functions $\mathcal{L}(w_t, (x_t, y_t))$, if and only if the matrix of the preferential structure on $\mathbf{M}$, in any coordinates in a neighbourhood of w, is given by*

$$(g_{ij}) = Df^{\mathrm{T}} \cdot Df, \quad \textit{for all} \quad w \in \mathbf{M}.$$

In particular f is an isometry, i.e., metric preserving map.

Proof Assume firstly that $g_{ij} = \left(Df^{\mathrm{T}} \cdot Df\right)_{ij}$, then g is the pullback $g = f^*\bar{g}$ via the link function of the standard Euclidean metric $\bar{g}$. That is, for any pair of vectors $X, Y \in \mathcal{T}_w\mathbf{M}$

$$g(w)(X,Y) = f^*\bar{g}(w)(X,Y) = \bar{g}(f(w))(f_*X, f_*Y) = \bar{g}(f_*X, f_*Y),$$

where $f_*\colon \mathcal{T}_w\mathbf{M} \to \mathcal{T}_{f(w)}\mathbb{R}^n \cong \mathbb{R}^n$ is the *push-forward* map given by (*cf.* Spivak, 1965).

$$f_*X\big|_w = Df(w)(X).$$

Therefore, at any $w \in \mathbf{M}$ and local frame $\{\partial_i\}$ on $\mathcal{T}\mathbf{M}$, putting ∂_i and ∂_j in place of X in the above formula, yields

$$\begin{aligned} g_{ij} &= g(\partial_i, \partial_j) = \bar{g}\Big(\sum_l (Df)_{il}\, E_l, \sum_m (Df)_{jm}\, E_m\Big) = \sum_{lm} (Df)_{il}\, (Df)_{jm}\; \bar{g}(E_l, E_m) \\ &= \sum_{lm} (Df)_{il}\, (Df)_{jm}\; \bar{g}_{lm} = \sum_l (Df)_{il}\, (Df)_{jl} = \left(Df^{\mathrm{T}} \cdot Df\right)_{ij}, \end{aligned}$$

because $\bar{g}_{lm} = \delta_{lm}$, where δ_{lm} is the Kronecker delta.

As a consequence $f\colon \mathbf{M} \to \mathbb{R}^n$ is an isometry between Riemannian manifolds. Geodesics are isometry invariant therefore f maps geodesics in $\mathbf{M}$ to geodesics in $\mathbb{R}^n$. Since geodesics in Euclidean space are straight lines one has

$$f(w_{t+1}) = f(\mathrm{Exp}_{w_t}(-\eta\,\mathrm{grad}\mathcal{L}_t(w_t))) = f(w_t) - \eta\, f_*\mathrm{grad}\mathcal{L}_t(w_t)).$$

Finally,

$$f_*\mathrm{grad}\mathcal{L}_t(w_t)) = f_* g^{-1}\boldsymbol{\partial}\mathcal{L}_t(w_t) = Df(w_t)\cdot Df(w_t)^{-1}\cdot Df(w_t)^{-\mathrm{T}}\boldsymbol{\partial}\mathcal{L}_t(w_t),$$

and the first part of the result follows.

In the reverse direction, by (6) we have that

$$f(w_{t+1}) = f(\mathrm{Exp}_{w_t}(-\eta\,\mathrm{grad}\mathcal{L}_t(w_t))) = f(w_t) - \eta\, f_*\mathrm{grad}\mathcal{L}_t(w_t),$$

for all $\eta > 0$. Comparing with (5), by the hypothesis it must be

$$f(w_t) - \eta\, f_*\mathrm{grad}\mathcal{L}_t(w_t) = f(w_t) - \eta\, Df(w_t)^{-\mathrm{T}}\boldsymbol{\partial}\mathcal{L}_t(w_t).$$

Substituting $g(w_t)^{-1}\boldsymbol{\partial}\mathcal{L}_t(w_t) = \mathrm{grad}\mathcal{L}_t(w_t)$ one has

$$-f_*\mathrm{grad}\mathcal{L}_t(w_t)) = -Df(w_t)\cdot g(w_t)^{-1}\boldsymbol{\partial}\mathcal{L}_t(w_t) = -Df(w_t)^{-\mathrm{T}}\boldsymbol{\partial}\mathcal{L}_t(w_t)$$

and consequently

$$0 = \left(g(w_t)^{-1} - Df(w_t)^{-1}\cdot Df(w_t)^{-\mathrm{T}}\right)\boldsymbol{\partial}\mathcal{L}_t(w_t).$$

Since the loss function $\mathcal{L}_t$ is arbitrary the result follows. ■

It is interesting to note that to obtain the converse result in the above theorem we use the fact that the loss function is arbitrary. In fact, it is only necessary that the span $\boldsymbol{\partial}\mathcal{L}_t(w_t)$ over all possible loss functions is $\mathbb{R}^n$ at each point $w \in \mathbf{M}$. The necessity of the condition indicates the independence of the choice of loss function from the structural assumptions of the learning problem. That is, we may analyse the same learning problem, with the same prior knowledge using different loss functions. It is interesting to ask whether there is a best loss function for a given analysis. We tackle this problem in forthcoming paper.

Any link function defines pullback metric on $\mathbf{M}$ but this is not a one-to-one correspondence. For example, a composition of a link function with an isometry on $\mathbb{R}^n$ is again a link function. One would like to explore the question whether there exists a link function for any preferential structure g on $\mathbf{M}$. The standard results of the Riemannian geometry assert that complete and simply connected Riemannian n-manifold $(\mathbf{M}, g)$ is isometric to $\mathbb{R}^n$ if and only if sectional curvature of g is zero everywhere. As a consequence one has the following criteria for existence of a link function.

Corollary 11 *The preferential structure g on $\mathbf{M}$ admits a link function if and only if its sectional curvature is zero everywhere.*

Proposition 10 provides a stochastic interpretation of the prior knowledge encoded in the link transformed Widrow-Hoff algorithm. Using the Bayesian interpretation of volume under the preferential structure, the link function structure is equivalent to a prior distribution

$$\phi = \sqrt{\det(g_{ij})} = \det\left(\sqrt{(g_{ij})}\right) = \det(Df). \tag{20}$$

Another interpretation of the link function transformation is that it maps from a parametrization $w \in \mathbf{M}$ with Bayesian prior $\phi(w)$ to a parametrization $\theta = f(w)$ with uniform prior distribution. This can be seen by computing the pullback of the Euclidean volume element $dV_{\bar{g}} = dx^1 \wedge \cdots \wedge dx^n$ on $\mathbb{R}^n$ (*cf.* Guillemin and Pollack, 1974; Spivak, 1965)

$$f^* dV_{\bar{g}} = f^*(dx^1 \wedge \cdots \wedge dx^n) = \det(Df)\, dw^1 \wedge \cdots \wedge dw^n = \phi\, dw^1 \wedge \cdots \wedge dw^n = dV_g.$$

This interpretation of the link transformed Widrow-Hoff algorithm corresponds to the structure of equation (5) and the motivation of general stochastic gradient descent algorithm.

Proposition 10 shows that for any link transformed Widrow-Hoff algorithm there is an equivalent preferential stochastic gradient descent algorithm associated with the preferential structure $(g_{ij}) = Df^{\mathrm{T}} \cdot Df$. The converse is not true.

Once a preferential structure — the metric on $\mathbf{M}$ — is known then any isometry from $\mathbf{M}$ to $\mathbb{R}^n$ will produce an equivalent algorithm, *cf.* Definition 3. Therefore there is a correspondence between a class of link functions, i.e., class of isometries up to the isometry group of $\mathbb{R}^n$, and classes of learning algorithms. By the following commutative diagram

$$\begin{array}{ccc} \mathbf{M} & \xrightarrow{\ f\ } & \mathbb{R}^n \\ \| & & \downarrow \varphi \\ \mathbf{M} & \xrightarrow[\tilde{f}]{} & \mathbb{R}^n \end{array} \tag{21}$$

clearly, if $\varphi\colon \mathbb{R}^n \to \mathbb{R}^n$ is an isometry of $\mathbb{R}^n$, i.e., it is a composition of an orthogonal linear operator and a translation, then $\varphi \circ f$ is another link function leading to an equivalent algorithm. Conversely, if f and $\tilde{f}$ are two isometries then the Diagram (21) commutes, and $\varphi = f^{-1} \circ \tilde{f}$, as a composition of isometries, is again an isometry.

The framework of link functions makes it now easy to apply the mistake bounds results of Section 4 to known online algorithms. Both mistake bounds are applicable in this case because $\bar{g}$ and any pullback metric $g = f^* \bar{g}$ via the link function has curvature zero.

6.2 Learning algorithms in a framework of link functions

Section 6.1 showed that as far as reparameterised learning algorithms are concerned, link functions are isometries of a space of weights $(\mathbf{M}, g)$ and Euclidean space $(\mathbb{R}^n, \bar{g})$. Intuitively, a link function f maps isometrically a step of an algorithm in $\mathbf{M}$ to a step of equivalent algorithm in Euclidean space, *cf.* Figure 2. The initial point $w_t \in \mathbf{M}$ is mapped to $f(w_t) \in \mathbb{R}^n$ and the velocity vector $V \in \mathcal{T}_{w_t}\mathbf{M}$ is mapped to $f_* V \in \mathcal{T}_{f(w_t)}\mathbb{R}^n \cong \mathbb{R}^n$. Since the curvature is isometry invariant (*cf.* Corollary 11) and the curvature of $\bar{g}$ is zero, therefore the curvature of the induced metric $g = f^* \bar{g}$ on $\mathbf{M}$ is also zero and Theorem 6 holds. In this section we apply the geometric framework to the two online algorithms, the gradient descent algorithm and the natural exponentiated gradient algorithm.

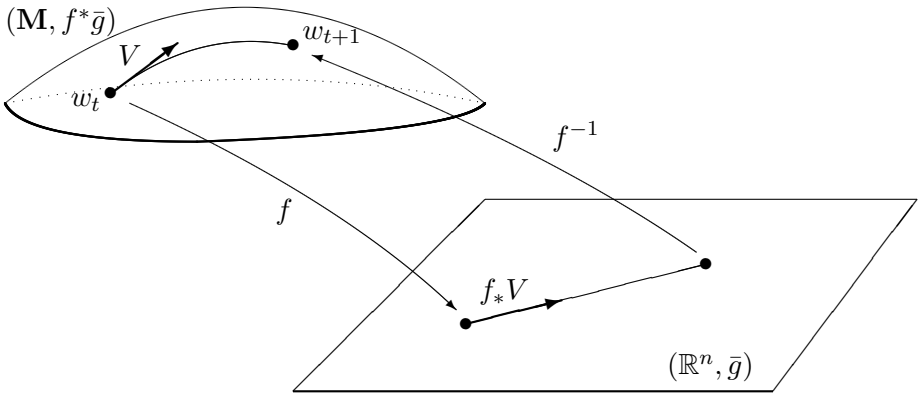

Figure 2: Link function f maps geodesic path given by the explicit update rule (10) to a line segment of the general additive algorithm in $\mathbb{R}^n$

We shall use the following standard notation for vector spaces. For any positive p, the p-norm is given by

$$\|X\|_p = \left(\left|X^1\right|^p + \cdots + \left|X^n\right|^p\right)^{1/p}, \quad \text{for any} \quad X \in \mathbb{R}^n.$$

We shall also use $\|X\|_\infty = \max_i \left|X^i\right|$. If $\langle X, Y\rangle$ denotes the inner product of two vectors $X, Y \in \mathbb{R}^n$ then the Euclidean norm $\|X\| = \langle X, X\rangle^{1/2} = \|X\|_2$. To shorten the expressions, let the sum of losses $\mathcal{L}_t$, for $t = 0, 1, \ldots, k-1$, be denoted by $\mathrm{Loss}(\mathfrak{A})$ and the losses $\mathcal{L}_t$ for a fixed $w_\star$ by $\mathrm{Loss}(w_\star)$, *cf.* (14) of Example 4.

We shall begin our discussion with the *gradient descent algorithm*, *cf.* Example 4.

Example 6 (The gradient descent algorithm) *This is the simplest case when the link function is the identity and the space of parameters* $\mathbf{M}$ *can be identified with* $\mathbb{R}^n$*. The geometric explicit learning algorithm (10) becomes* $w_{t+1} = w_t - \eta\, \partial\mathcal{L}_t(w_t)$*, which is the standard Widrow-Hoff perceptron algorithm (1). Let the loss function be convex and depend on the inner product* $\langle w, x_t\rangle$*, i.e., has the following form* $\mathcal{L}_t(w) = \mathcal{L}(w, (x_t, y_t)) = \mathcal{L}(y_t, \langle w, x_t\rangle)$*. We shall derive an upper bound for the total loss of this algorithm from Theorem 6. From the discussion in Section 3.2 it follows the the distance between two consecutive point of the algorithm is proportional to the gradient of* $\mathcal{L}$*. Hence*

$$\mathcal{D}(w_t, w_{t+1}) = \frac{1}{2}\|w_{t+1} - w_t\|^2 = \frac{1}{2}\eta^2 \left|\mathcal{L}_t'(w_t)\right|^2 \|x_t\|^2,$$

where $\mathcal{L}_t'$ *denotes derivative of* $\mathcal{L}\colon \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ *with respect to the second argument. Set the constants to satisfy the following conditions:* $R_2 \geq \max_t \|x_t\|_{\bar{g}}$*,* $Z \geq \max_t |\mathcal{L}_t'(w_t)|$ *and* $U \geq \|w_\star\|$*. Take the initial weight* w_0 *set to* 0 *then the terms in (12) satisfy*

$$\frac{1}{\eta}\mathcal{D}(w_\star, w_0) \leq \frac{1}{2\eta}U^2, \quad \textit{and} \quad \frac{1}{\eta}\sum_{t=0}^{k-1}\mathcal{D}(w_t, w_{t+1}) \leq \frac{1}{\eta}\, k\, \frac{1}{2}\eta^2 Z^2 R_2^2 = \frac{\eta}{2}\, k\, Z^2 R_2^2.$$

Applying these calculations to (12) yields

$$\mathrm{Loss}(\mathfrak{A}) - \mathrm{Loss}(w_\star) \leq \frac{1}{2\eta}U^2 + \frac{\eta}{2}\, k\, Z^2 R_2^2.$$

The right hand side of the above inequality assumes its minimum when the two terms are equal, i.e., the learning rate is $\eta = U/(R_2 Z\sqrt{k})$. *In this case*

$$\mathrm{Loss}(\mathfrak{A}) - \mathrm{Loss}(w_\star) = \frac{R_2 Z\sqrt{k}}{2U}\, U^2 + \frac{U}{2R_2 Z\sqrt{k}}\, k\, Z^2 R_2^2 = R_2\, Z\, U\, \sqrt{k}.$$

The upper relative bound on the cumulative loss is proportional to the square root of the length of a sequence k. *This result agrees with Theorem 3 in Cesa-Bianchi (1999).* ▷

We shall now review the natural exponentiated gradient algorithm introduced in Mahony and Williamson (2001). We note here that despite similarity of the algorithm to the explicit algorithm in the simplex Δ^n (Example 3) the two algorithms are fundamentally different and are *not equivalent* in the view of Definition 3.

Example 7 (The natural exponentiated gradient algorithm) *This is the* unconstrained version of the exponentiated gradient *(UGE) proposed by Mahony and Williamson (2001). The link function* $f\colon (\mathbb{R}_+)^n \to \mathbb{R}^n$ *is a natural logarithm* $\ln$ *acting on each component, i.e.,* $f^i(w) = \ln(w^i)$, *for* $i = 1, 2, \ldots, n$. *The reparameterised Widrow-Hoff algorithm (5) can be now written in coordinates*

$$\ln w_{t+1}^i = \ln w_t^i - \eta\, \mathcal{L}_t'(w_t)\, x_t^i\, w^i. \tag{22}$$

The induced metric g *on* $\mathbf{M} = (\mathbb{R}_+)^n$ *is a diagonal matrix given by*

$$(g_{ij}(w)) = \begin{pmatrix} \frac{1}{(w^1)^2} & 0 & \cdots & 0 \\ 0 & \frac{1}{(w^2)^2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{1}{(w^n)^2} \end{pmatrix}.$$

For a convex loss function we derive an upper bound of the total loss of this algorithm (Theorem 6) as follows. The distance $\mathrm{dist}(w_{t+1}, w_t)$ *is equal to the length of the geodesic from* w_t *to* w_{t+1}, *which in turn is equal to the norm of the velocity vector*

$$\mathrm{dist}(w_{t+1}, w_t) = \left\|\mathrm{Exp}_{w_t}^{-1} w_{t+1}\right\|_g = \left\|f(w_{t+1}) - f(w_t)\right\|.$$

Note the change of norm in the above expression. By (10) this norm is equal to

$$\left\|\eta\, \mathrm{grad}\mathcal{L}_t(w_t)\right\| = \eta\, \left\|f_* \mathrm{grad}\mathcal{L}_t(w_t)\right\| = \eta\, \left\|Df(w_t)^{-\mathrm{T}} \boldsymbol{\partial}\mathcal{L}_t(w_t)\right\|,$$

where $\left(Df(w_t)^{-\mathrm{T}} \boldsymbol{\partial}\mathcal{L}_t(w_t)\right)^i = \mathcal{L}_t'(w_t)\, x_t^i\, w_t^i$. *Therefore*

$$\mathrm{dist}(w_{t+1}, w_t) = \eta\, \left|\mathcal{L}_t'(w_t)\right|\, \left\|\left(x_t^i\, w_t^i\right)\right\|_2 \leq \eta\, \left|\mathcal{L}_t'(w_t)\right|\, \left\|x_t\right\|_\infty \left\|w_t\right\|_2 \leq \eta\, Z R_\infty U,$$

where we set $R_\infty \geq \max_t \left\|x_t\right\|_\infty$, $Z \geq \max_t \left|\mathcal{L}_t'(w_t)\right|$ *and* $U \geq \max_t \left\|w_t\right\|_2$. *Hence*

$$\frac{1}{\eta} \sum_{t=0}^{k-1} \mathcal{D}(w_t, w_{t+1}) \leq \frac{1}{\eta}\, k\, \frac{1}{2} \eta^2 Z^2 R_\infty^2 U^2 = \frac{\eta}{2}\, k\, Z^2 R_\infty^2 U^2.$$

Assigning the learning rate $\eta = 1/(Z\, R_\infty \sqrt{k})$ *that minimizes the sum of terms in Theorem 6 and choosing initial* $w_0 = (1, \ldots, 1)$ *yields*

$$\mathrm{Loss}(\mathfrak{A}) - \mathrm{Loss}(w_\star) = \frac{1}{2\eta}\, U^2 + \frac{\eta}{2}\, k\, Z^2 R_\infty^2 U^2 = R_\infty\, Z\, U^2\, \sqrt{k}.$$

▷

From the two above examples one can see that the mistake bound analysis for the two algorithms yields similar results. The bound in both cases is of order of $\sqrt{k}$. The only difference here are the constants that depend on the norms of the input x_t and parameter w_t. This is the effect of using different link functions. Depending on the predicted magnitudes of the input and the parameter one would prefer one algorithm over the other.

7. Conclusion

We have presented two new online learning algorithms and their analysis from a geometric view. The algorithms can be seen as a particular stochastic gradient descent rule in the presence of constraints, for example. Two such cases of the sphere and the probability simplex have been demonstrated. Their performance is at least as good as the exponentiated gradient descent algorithm. Our investigation helped us to understand online learning algorithms and their connection with geometry of the parameter space. We were able to derive the mistake bounds for a general loss function by the methods of Riemannian geometry. Many questions however, remain yet to be answered.

The current and future research is focused on the choice of a suitable loss function for the geometric online stochastic gradient descent algorithm. We also want to understand better the optimal geometric structure that would suit a particular learning problem. It seems that we only brushed the surface of a larger domain of geometry induced by probability distributions, both parametric and non-parametric. We hope that this paper will inspire further research in this fascinating combination of machine learning, probability and geometry.

8. Acknowledgments

Krzysztof Krakowski and Robert Mahony were supported by the Australian Research Council. National ICT Australia is funded by the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian Research Council.

References

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

Nicolò Cesa-Bianchi. Analysis of two gradient-based algorithms for on-line regression. *Journal of Computer and System Sciences*, 59(3):392–411, 1999. ISSN 0022-0000. doi: http://dx.doi.org/10.1006/jcss.1999.1635.

Peter M. Clarkson. *Optimal and Adaptive Signal Processing*. CRC Press, Boka Raton, 1993. ISBN 0849386098.

Frank Critchley, Paul Marriott, and Mark Salmon. Preferred point geometry and the local differential geometry of the Kullback-Leibler divergence. *The Annals of Statistics*, 22(3):1587–1602, 1994.

Claudio Gentile and Nick Littlestone. The robustness of the p-norm algorithms. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 1–11. ACM Press, 1999. ISBN 1-58113-167-4. doi: http://doi.acm.org/10.1145/307400.307405.

Geoffrey J. Gordon. *Approximate Solutions to Markov Decision Processes*. PhD thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, 1999a.

Geoffrey J. Gordon. Regret bounds for prediction problems. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 29–40. ACM Press, 1999b. ISBN 1-58113-167-4. doi: http://doi.acm.org/10.1145/307400.307410.

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*, pages 171–183. ACM Press, 1997. ISBN 0-89791-891-6. doi: http://doi.acm.org/10.1145/267460.267493.

Victor Guillemin and Alan Pollack. *Differential Topology*. Prentice-Hall, New Jersey, 1974.

A. Jagota and Manfred K. Warmuth. Continuous and discrete time nonlinear gradient descent: relative loss bounds and convergence. In *International Symposium on Artificial Intelligence and Mathematics*, 1998.

H. Karcher. Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics*, 30:509–541, 1977.

Robert E. Kass and Paul W. Vos. *Geometrical Foundations of Asymptotic Inference*. Wiley Series in Probability and Statistics. John Wiley and Sons, New York, 1997.

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

Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997. ISSN 0890-5401. doi: http://dx.doi.org/10.1006/inco.1996.2612.

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*, volume 10. The MIT Press, 1998.

J. Lafferty, S. Pietra, and V. Pietra. Statistical learning algorithms based on Bregman distances. In *Proceedings of the Canadian Workshop on Information Theory*, 1997.

Guy Lebanon. *Riemannian Geometry and Statistical Machine Learning*. PhD thesis, Carnegie Mellon University, 2005.

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

Robert E. Mahony and Robert C. Williamson. Prior knowledge and preferential structures in gradient descent learning algorithms. *Journal of Machine Learning Research*, 1:311–355, 2001.

Peter Petersen. *Riemannian Geometry*. Number 171 in Graduate Texts in Mathematics. Springer-Verlag, New York, 1998.

Michael Spivak. *Calculus on Manifolds*. Mathematics Monograph Series. Addison-Wesley, New York, 1965.