

A geometric perspective of on-line machine learning for regression problems.

Robert Mahony

Department of Engineering,
Australian National University,

email: Robert.Mahony@anu.edu.au

url: <http://engnet.anu.edu.au/DEpeople/Robert.Mahony/>

Joint work with:

Kris Krakowski, Kim Blackmore, Bob Williamson, Manfred Warmuth

Presentation for ANU Systems and Control Reading Group,
RSISE, Australian National University, Canberra

A geometric perspective of on-line machine learning for regression problems

- A brief introduction to regression problems.
- A geometric perspective on loss functions.
- Regularisation and iterates for on-line learning algorithms.
- Geometric gradient (GG) descent algorithm.
- Results and comparisons

On-line parametric regression problems

Model class: A model class is a parameterised set of maps that generate regression estimates

$$\hat{y} = f_p(x), \quad f_p : \Omega_x \rightarrow \Omega_y, \quad p \in \mathbf{M}$$

Geometric setting: \mathbf{M} is a Riemannian manifold

Generative noise model: Data is generated by

$$y_k = f_{p_\star}(x_k) + \nu_k, \quad p_\star \in \mathbf{M}$$

where ν_k is some ‘noise’ process - possibly zero.

Goal: The goal is to determine the parameter p_\star that best ‘explains’ the observed data $\{(y_k, x_k)\}$.

On-line requirement: A continuously updated parameter estimate p_k is required at each time step.

Characteristics of on-line machine learning regression problems

1. Very high dimensionality of the unknown parameter p_* .
2. The measurement noise ν_k can be extremely large.
3. The noise is often highly non-Gaussian due to the underlying structure of the problem.

Machine learning problems are rarely straightforward linear regression problems subject to Gaussian measurement noise.

Where does the geometry come from

Expected Information Geometry:

$$y_k = f_p(x_k) + \nu_k, \quad \nu_k \sim \phi(y, x|p)$$

where $\phi(y, x|p) \in \Phi_{\mathbf{M}}$ is a parameterised set of probability distributions.

Let ∂_{p^i} denote variation in the i th parameter p on \mathbf{M} .

The Fisher information metric (in the frame $\{\partial_{p^i}\}$) is

$$g_{ij}(p_0) = \mathbb{E}_{\phi(y, x|p_0)} \left[(\partial_{p^i} \log \phi(y, x|p)) (\partial_{p^j} \log \phi(y, x|p)) \right]$$

Prior Knowledge: A preferential structure $g(p)$ is given as prior information. The prior distribution on parameter space is

$$\psi(p) = \sqrt{\det(g(p))}$$

Loss functions

In regression algorithms the loss function is a crucial part of the formulation

$$\mathcal{L}_k(p) := \mathcal{L}((y_k, x_k), p_k), \quad \mathcal{L}: \Omega \times \mathbf{M} \rightarrow \mathbb{R}_+.$$

Least squares error:

$$\mathcal{L}_k^{\text{LS}}(p) = \frac{1}{2} |y_k - f_p(x_k)|^2$$

Normalised least squares error for linear regression:

$$\mathcal{L}_k^{\text{NLS}}(p) = \frac{1}{2} \frac{|y_k - \langle x_k, p \rangle|^2}{|x_k|^2}$$

Log likelihood:

$$\mathcal{L}_k^{\text{LL}}(p) = -\log(\phi(y_k, x_k | p))$$

What properties are important in a loss function

- A loss function should penalise estimation error.

In particular, a loss function should be zero on the set

$$S_k(0) = \{p \mid y_k = f_p(x_k)\}$$

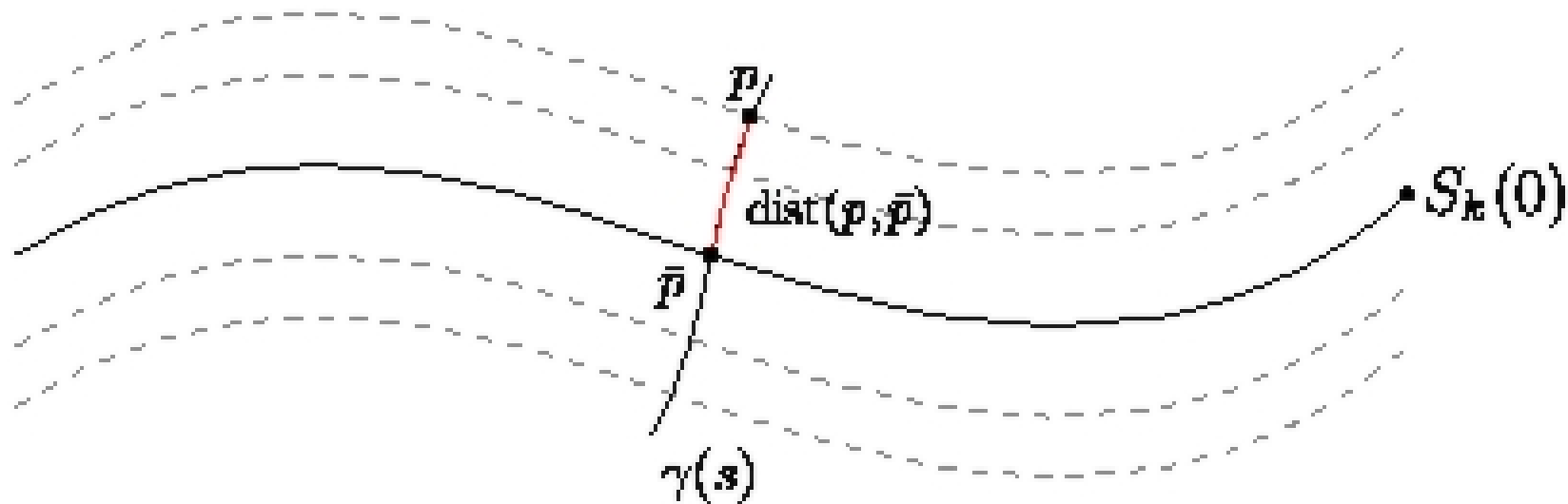
and positive off this set.

- A loss function should respect the **geometry** of the problem.

Proposed loss function

Loss function

$$\mathcal{L}_k(p) = \frac{1}{2} \text{dist}_M(p, S_k(0))^2 = \min_{\{\bar{p} \mid y_k = \langle x_k, \bar{p} \rangle\}} \frac{1}{2} \text{dist}_M(p, \bar{p})^2$$



Optimal distance is realised along the geodesic $\gamma(s)$ orthogonal to $S_k(0)$.

Example: Least squares error loss function

Problem type:

Linear regression with unit variance measurement noise.

Generative noise model

$$y_k = \langle x_k, p_\star \rangle + \nu_k, \quad \nu_k \sim \phi(y_k | x_k, p_\star) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2} |y_k - \langle x_k, p_\star \rangle|^2\right)$$

Parameterized Model Class

$$\mathbf{M} = \left\{ \phi \mid \phi(y_k | x_k, p) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2} |y_k - \langle x_k, p \rangle|^2\right) \right\}$$

Expected information metric

$$g_{ij}(p) = \mathbb{E}_y \left[(\partial_i \log \phi)(\partial_j \log \phi) \right] = \frac{\partial^2}{\partial p^i \partial p^j} \left(\frac{1}{2} p^T x_k x_k^T p \right) = x_k x_k^T$$

Geometric interpretation of LS loss

Compute the distance from p_k to the $S_k(0)$ with respect to the metric

$$\begin{aligned}d(p, S_k(0)) &= \min_{\{\bar{p} \mid y_k = \langle x_k, \bar{p} \rangle\}} \int_0^1 (g_{\gamma_\tau}(\dot{\gamma}_\tau, \dot{\gamma}_\tau))^{\frac{1}{2}} d\tau, & [\gamma_\tau = \tau p - (1 - \tau)\bar{p}] \\ &= \int_0^1 \sqrt{(p - \bar{p}_0)^T x_k x_k^T (p - \bar{p}_0)} d\tau, & \bar{p}_0 \in S_k(0) \\ &= |\langle (p - \bar{p}_0), x_k \rangle|.\end{aligned}$$

Thus, one has

$$\begin{aligned}\mathcal{L}_k^{\text{LS}}(p) &= \frac{1}{2} |\langle p, x_k \rangle - y_k|^2 = \frac{1}{2} \langle (p - \bar{p}_0), x_k \rangle^2 \\ &= \min_{\{\bar{p} \mid y_k = \langle x_k, \bar{p} \rangle\}} \frac{1}{2} \text{dist}_{\text{M}}(p, \bar{p})^2 = \frac{1}{2} \text{dist}_{\text{M}}(p, S_k(0))^2\end{aligned}$$

Example: Normalised least squares error

Problem type:

Linear regression with unit variance measurement noise and Gaussian i.i.d. sample distribution with unit variance.

Generative noise model

$$y_k = \langle x_k, p_\star \rangle + \nu_k, \quad \phi(y_k | x_k, p_\star) = N(\langle x_k, p_\star \rangle, 1), \quad \psi(x_k) = N(0, 1)$$

Parameterized Model Class

$$\mathbf{M} = \left\{ \phi \mid \phi(y_k, x_k | p) = \frac{1}{\sqrt{2\pi}} \exp \left(-\frac{1}{2} |y_k - \langle x_k, p \rangle|^2 - \frac{1}{2} |x_k|^2 \right) \right\}$$

Expected information metric

$$g_{ij}(p) = \mathbb{E}_{y,x} \left[(\partial_i \log \phi)(\partial_j \log \phi) \right] = \mathbb{E}_x \left[\frac{\partial^2}{\partial p^i \partial p^j} \left(\frac{1}{2} p^T x x^T p \right) \right] = \delta_{ij}$$

Geometric interpretation of NLS loss

Compute the distance from p_k to the output set with respect to the metric

$$\begin{aligned}d(p, S_k(0)) &= \min_{\{\bar{p} \in S_k(0)\}} \int_0^1 g_{\gamma_\tau}(\dot{\gamma}_\tau, \dot{\gamma}_\tau)^{\frac{1}{2}} d\tau, & [\gamma_\tau = \tau p - (1 - \tau)\bar{p}] \\ &= \int_0^1 |(p - \bar{p}_0)| d\tau, & \bar{p}_0 = \arg \min_{\bar{p} \in S_k(0)} d(p, \bar{p}) \\ &= \left\langle (p - \bar{p}_0), \frac{x_k}{|x_k|} \right\rangle\end{aligned}$$

Thus, one has

$$\mathcal{L}_k^{\text{NLS}}(p) = \frac{1}{2} \frac{\langle (p - \bar{p}_0), x_k \rangle^2}{|x_k|^2} = \min_{\bar{p} \in S_k(0)} \frac{1}{2} \text{dist}_{\text{M}}(p, \bar{p})^2 = \frac{1}{2} \text{dist}_{\text{M}}(p, S_k(0))^2$$

Log likelihood loss functions

- The log likelihood loss function corresponds to the least squares and normalised least squares cost in the two cases considered above.

These cases correspond to Euclidean geometry - simple cases.

- For general families of distributions the log likelihood is some approximation of a distance function - similar in nature to a statistical divergence.

Log likelihood does not have a simple geometric interpretation.

Geometric loss function

Geometric loss function

$$\mathcal{L}_k(p) = \frac{1}{2} \text{dist}_{\mathbf{M}}(p, S_k(0))^2 = \min_{\{\bar{p} | y_k = \langle x_k, \bar{p} \rangle\}} \frac{1}{2} \text{dist}_{\mathbf{M}}(p, \bar{p})^2$$

- Depends only on geometry of the problem.
- Equally applicable to expected or observed information geometry or preferential structure.
- Can be applied to curved exponential families.
- Generalises to standard loss functions for classical regression problems.

Deriving an algorithm

1. Parameter update p_k must be available at each data iterate. Algorithm must be an iterative update.
2. Differentiable loss function available at each step .
3. No one instance of data should be trusted significantly due to noise. **Small step updates. No conjugate gradient updates.**
4. Dimensionality and degeneracy of problem prevents the use of second order optimisation methods.

Stochastic gradient (SG) descent algorithm is a small step update in the direction of the gradient of the loss function.

$$p_{k+1} = p_k - \eta \frac{\partial \mathcal{L}_k}{\partial p}(p_k)$$

Regularisation and MLE's

Consider a maximum likelihood estimate for a given data sample

$$\hat{p} = \arg \max_p \log \phi(y_k, x_k | p)$$

Example: Linear regression with Gaussian measurement noise

$$\hat{p} = \arg \max_p \left(-\frac{1}{2} |y_k - \langle x_k, p \rangle|^2 \right)$$

Regularisation is the process of introducing a prior distribution $\psi(p)$ into the maximum likelihood estimate

$$\hat{p} = \arg \max_p (\log \phi(y_k, x_k | p) \psi(p))$$

Example: Gaussian prior at p_k with variance $1/\sqrt{\eta}$

$$\hat{p} = \arg \max_p \left(-\frac{1}{2} |y_k - \langle x_k, p \rangle|^2 - \frac{1}{2\eta} |p - p_k|^2 \right), \quad \psi(p) = \exp \left(-\frac{1}{2\eta} |p - p_k|^2 \right).$$

SG algorithm is the regularised MLE

The regularised maximum likelihood estimator was

$$\hat{p} = \arg \max_p \left(-\frac{1}{2} |y_k - \langle x_k, p \rangle|^2 - \frac{1}{2\eta} |p - p_k|^2 \right) = \arg \min_p U_k(p)$$

Recall the loss function for normally distributed linear regression

$$\mathcal{L}_k(p) = \frac{1}{2} |y_k - \langle x_k, p \rangle|^2$$

The MLE \hat{p} occurs at the critical point of $U_k(p)$ (convexity)

$$p_{k+1} = \hat{p} = p_k - \eta \frac{\partial \mathcal{L}_k}{\partial p}(p_k)$$

This is the stochastic gradient descent algorithm

Geometric Gradient Descent Algorithm

Define

$$U_k(p) = \mathcal{L}_k(p) + \frac{1}{2\eta} \text{dist}_{\mathbf{M}}(p, p_k)^2$$

where $\mathcal{L}_k(p)$ is the geometric regression loss.

Think of the regularisation function $\frac{1}{\eta} \text{dist}_{\mathbf{M}}(p, \bar{p})^2$ as related to a Gaussian prior with variance $1/\sqrt{\eta}$ and mean p_k , distributed with respect to the given geometry.

The **geometric gradient descent (GD)** update is given by

$$p_{k+1} = \arg \min_{p \in \mathbf{M}} U_k(p)$$

Explicit and implicit update steps

Evaluate the critical point of $U_k(p)$

$$\begin{aligned} dU_k(p)[V] &= d\mathcal{L}_k(p)[V] + \frac{1}{2\eta} d\text{dist}_{\mathbf{M}}(p, p_k)^2[V] \\ &= \langle \text{grad}\mathcal{L}_k(p), V \rangle_g + \frac{1}{\eta} \langle -\text{Exp}_p^{-1} p_k, V \rangle_g = 0, \quad \forall V \in T_p\mathbf{M} \end{aligned}$$

Solving the above equation yields the **implicit** update equation

$$p_t = \text{Exp}_{p_{t+1}} \left(\eta \text{grad}\mathcal{L}_t(p_{t+1}) \right),$$

The **explicit** update is obtained by approximating the solution to the implicit update step

$$p_{t+1} = \text{Exp}_{p_t} (-\eta \text{grad}\mathcal{L}_t(p_t))$$

Matching geometry of loss and regularisation function

Lemma:

Integral curves of the gradient vector field $\text{grad}\mathcal{L}_k$ are reparameterised geodesics of the Riemannian geometry.

Lemma:

Let $p_{t+1}^I(\eta)$ denote the GG implicit update iterate and $p_{t+1}^E(\eta)$ denote the GG explicit update iterate, then

$$p_{t+1}^I(\eta) = p_{t+1}^E\left(\frac{\eta}{1 + \eta}\right)$$

The step-size η is chosen according to the underlying problem and analysis.

A geometric view of on-line algorithms for regression problems

Generative noise model

$$y_k = f_p(x_k) + \nu_k, \quad p \in \mathbf{M} \quad \text{Riemannian manifold}$$

Loss function

$$\mathcal{L}_k(p) = \min_{\{\bar{p} \mid y_k = \langle x_k, \bar{p} \rangle\}} \frac{1}{2} \text{dist}_{\mathbf{M}}(p, \bar{p})^2$$

The **geometric gradient (GG)** descent update is given by

$$p_{t+1} = \text{Exp}_{p_t}(-\eta \text{grad} \mathcal{L}_t(p_t))$$

that minimises the regularised loss function

$$U_k(p) = \mathcal{L}_k(p) + \frac{(1 - \eta)}{\eta} \text{dist}_{\mathbf{M}}(p, p_k)^2$$

Regression over a multinomial distribution

Multinomial distribution

$$\phi(\zeta|p) = \frac{m!}{\prod \zeta^i!} \prod (p^i)^{\zeta^i}$$

entries of ζ are the count of event i observed over m trials.

Generative noise model

$$y_k = \left\langle \frac{1}{m} \zeta_k, x_k \right\rangle, \quad \zeta_k \sim \phi(\zeta|p)$$

Prediction estimates

$$\hat{y}_k = \langle p_k, x_k \rangle$$

the entries of p_k are estimates of the probability of event i .

$$\sum_{i=1}^{n+1} p_k^i = 1, \quad p_k \in \Delta^n \quad \text{Simplex}$$

LS Algorithm

Least squares error

$$\mathcal{L}_k^{LS}(p) = \frac{1}{2} |y_k - \langle x_k, p \rangle|^2$$

Stochastic gradient algorithm

$$p_{k+1} = \frac{p_k + \eta(y_k - \langle x_k, p_k \rangle)x_k}{|p_k + \eta(y_k - \langle x_k, p_k \rangle)x_k|}$$

The normalisation factor is necessary to preserve the probability constraint.

Geometry of the multinomial regression

Expectation parameters $p \in \Delta^n$ lie on the simplex

$$\Delta^n = \{p \in \mathbb{R}_+^{n+1} \mid \sum p^i = 1\}$$

The expected information metric is

$$g_p = \frac{1}{m} \begin{pmatrix} \frac{1}{p^1} & 0 & 0 \\ 0 & \cdots & 0 \\ 0 & 0 & \frac{1}{p^{n+1}} \end{pmatrix}$$

This metric is equivalent to that induced on the simplex via the isometry

$$\varphi(p^1, \dots, p^{n+1}) := 2\sqrt{m}(\sqrt{p^1}, \dots, \sqrt{p^{n+1}})$$

mapping to the sphere $2\sqrt{m}S^n \hookrightarrow \mathbb{R}^{n+1}$, a regular submanifold of Euclidean space.

Geometric loss function

Compute loss function

$$\mathcal{L}(p, (x_t, y_t)) = 4 \arccos^2 (2 (\lambda_1 y_t + \lambda_2)),$$

where

$$\underbrace{\sum_{i=1}^{n+1} \frac{p^i}{(\lambda_1 x_t^i + \lambda_2)^2}}_{\text{simplex constraint}} = 4,$$

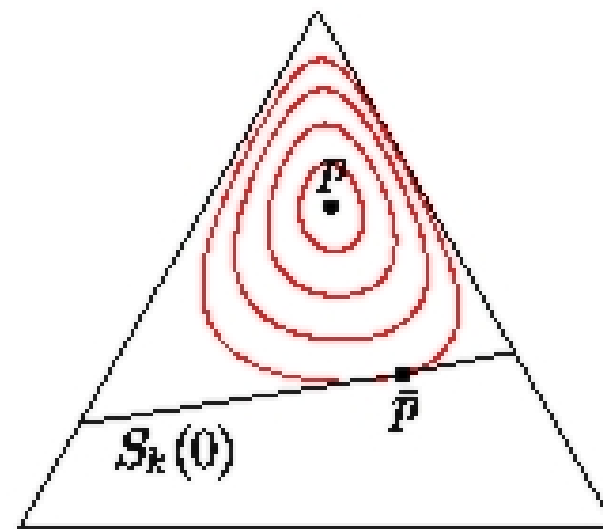
$$\underbrace{\sum_{i=1}^{n+1} \frac{p^i x_t^i}{(\lambda_1 x_t^i + \lambda_2)^2}}_{\text{data constraint}} = 4y_t.$$

Lagrange multiplier techniques used to compute loss function:

λ_1 : Simplex constraint.

λ_2 : Data constraint

$$\bar{p} \in S_k(0).$$



Geometric Gradient algorithm

Gradient of the Loss function

$$(\text{grad}\mathcal{L}_t(p))^i = \sqrt{p^i} \frac{\partial}{\partial p^i} \mathcal{L}_t(p) - \sqrt{p^i} \sum_{j=1}^{n+1} p^j \frac{\partial}{\partial p^j} \mathcal{L}_t(p)$$

where

$$\frac{\partial}{\partial p^i} \mathcal{L}_t(p) = -\frac{2 \arccos(2(\lambda_1 y_t + \lambda_2))}{(\lambda_1 x_t^i + \lambda_2) \sqrt{1 - 4(\lambda_1 y_t + \lambda_2)^2}}.$$

Update iterate

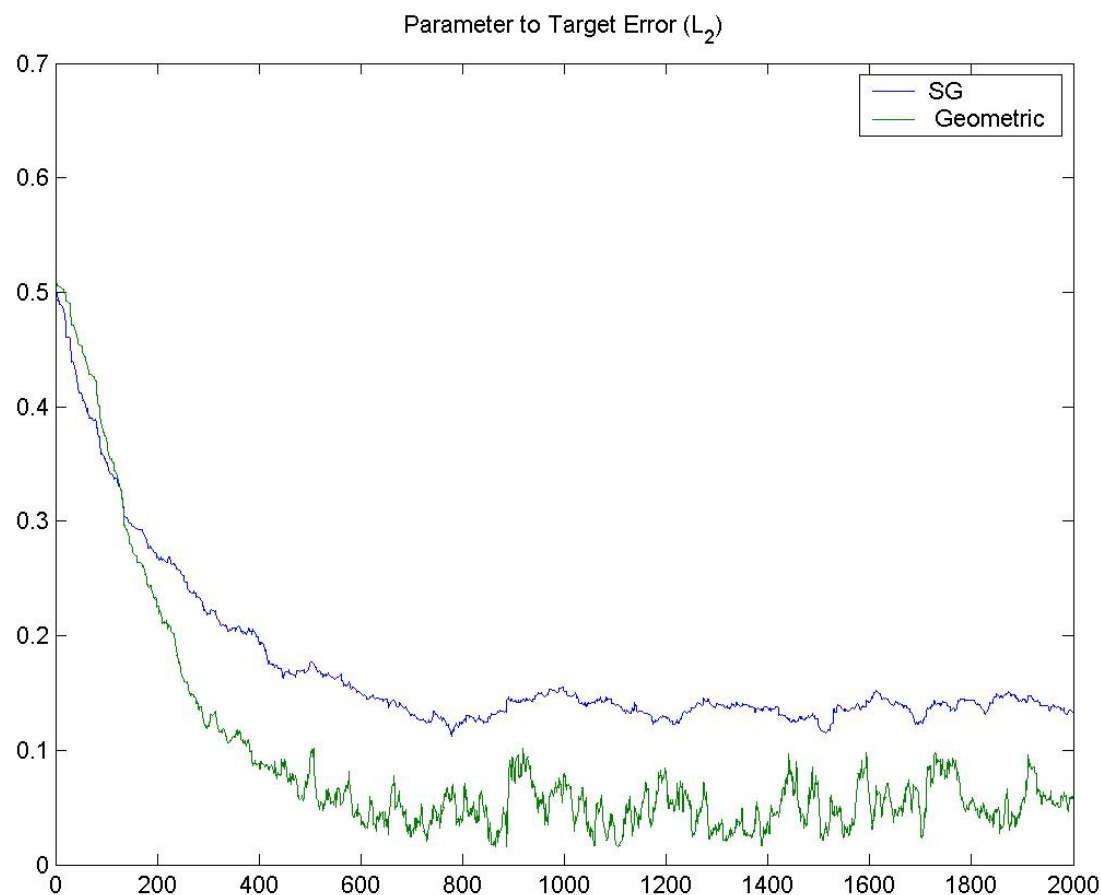
$$p_{t+1}^i = \left(\sqrt{p_t^i} \cos(\eta |\text{grad}\mathcal{L}_t(p)|/2) - \frac{(\text{grad}\mathcal{L}_t(p))^i}{|\text{grad}\mathcal{L}_t(p)|} \sin(\eta |\text{grad}\mathcal{L}_t(p)|/2) \right)^2,$$

Great circles on the Sphere $\sqrt{4m}S^n$.

Results

Comparison of performance of GG and SG algorithms

Step-sizes were separately calculated by equalising the mean square error of the output.



Best existing algorithm

When the parametric inference problem is linked to an exponential family then it is possible to use a regularisation process based on the Bregman divergence.

Using the least squares loss function with regularising function based on the Bregman divergence in the natural coordinates one obtains the exponentiated gradient (EG) algorithm

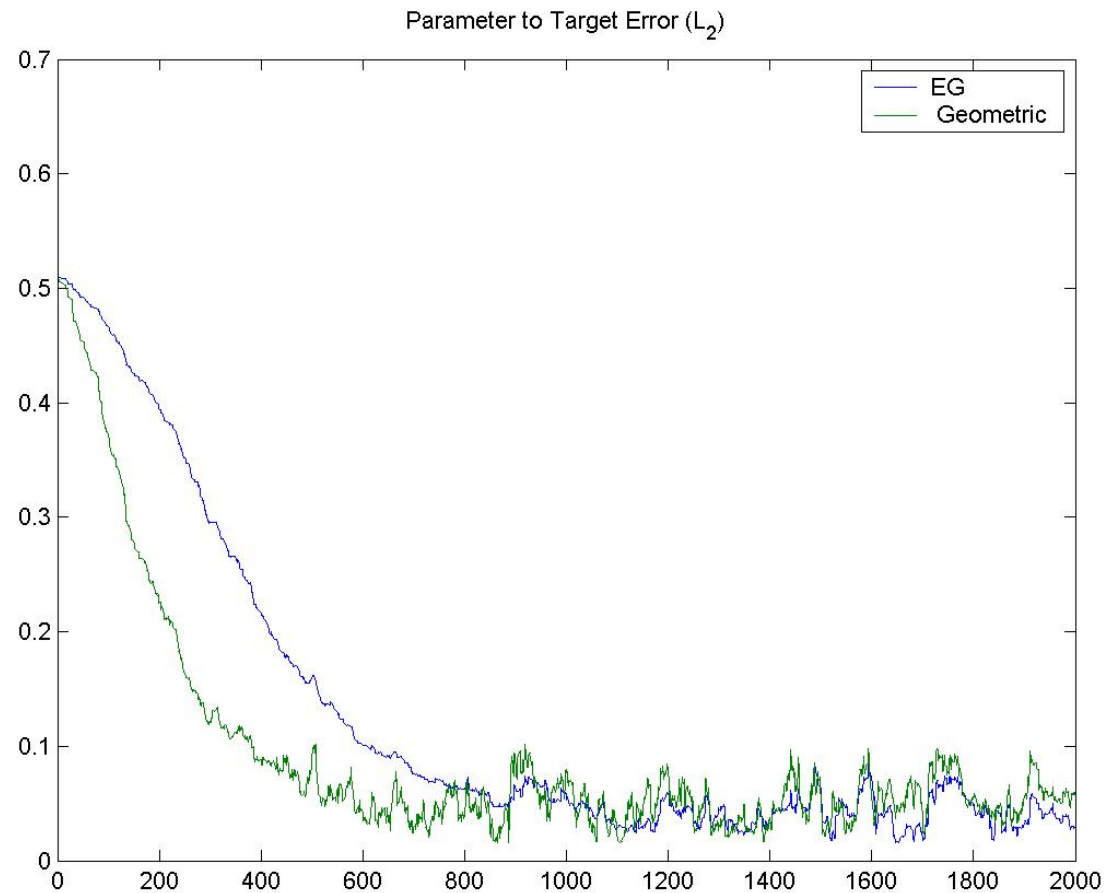
$$p_{k+1}^i = \frac{p_k^i \exp(\eta(y_k - \langle x_k, p_k \rangle) x_k^i)}{\sum_{j=1}^{n+1} p_k^j \exp(\eta(y_k - \langle x_k, p_k \rangle) x_k^j)}$$

Note that this algorithm is the explicit version of an implicit algorithm. Using the Bregman divergence and LS loss the explicit and implicit updates are not equivalent.

Results

Comparison of performance of EG and GG algorithms

Experiment shown is best performance. In general, performance of GG is comparable to that of the EG algorithm.



Conclusions

- Development requires only the geometric structure. Applicable to expected and observed information geometries, preferential structures, curved exponential families, etc.
- Specialises to stochastic gradient descent algorithm for the classical linear regression problem.
- Experimental results indicate excellent performance.
- The drawback is the computational effort involved in computing the geometric loss function and geodesic updates.

The rôle of this work is to provide a theoretically efficient standard approach to deriving on-line machine learning algorithms against which practical algorithms can be benchmarked.

All material presented is jointly undertaken with



Kris Krakowski

Department of Engineering,
Australian National University.



Kim Blackmore

Department of Engineering,
Australian National University



Bob Williamson

National ICT Australia,
Australian National University.



Manfred Warmuth

Department of Computer Science,
Univ. of Calif., Santa Cruz,