AdaGeo: Adaptive Geometric Learning for Optimization and Sampling

Gabriele Abbati[1], Alessandra Tosi[2], Michael A Osborne[1], Seth Flaxman[3]

[1]University of Oxford, [2]Mind Foundry Ltd, [3]Imperial College London

Abstract

In high-dimensional optimization and sampling, well-known issues such as slow-mixing, non-convexity and correlations can arise.

We propose **AdaGeo**, a preconditioning framework for **ada**ptively learning the **geo**metry of parameter space.

We use the **Gaussian Process latent variable model** (GP-LVM) to represent a lower-dimensional embedding of the parameters, identifying the underlying **Riemannian manifold** on which the optimization or sampling is taking place. Samples or optimization steps are then proposed based on the geometry of the manifold.

We apply our framework to **stochastic gradient descent** (SGD) and **stochastic gradient Langevin dynamics** (SGLD) and show performance improvements for both optimization and sampling.

GP-Latent Variable Models

Latent variable models relate a set of observed variables $\Theta \subset \mathbb{R}^D$ to a set of lower-dimensional unobserved (or **latent**) variables $\Omega \subset \mathbb{R}^Q$ through the mapping f:

$$\boldsymbol{\theta} = f(\boldsymbol{\omega}) + \boldsymbol{\eta} \tag{6}$$

where $\boldsymbol{\theta} \in \Theta$, $\boldsymbol{\omega} \in \Omega$ and $\boldsymbol{\eta}$ is a noise term. Gaussian Process Latent Variable models assume a **Gaussian Process prior** for f.

For differentiable kernels the **Jacobian J** of f is normally distributed (if its columns are assumed to be independent):

$$p(\mathbf{J} \mid \boldsymbol{\Theta}, \boldsymbol{\Omega}, \beta) = \prod_{i=1}^{D} \mathcal{N}(\mathbf{J}_{i,:} \mid \boldsymbol{\mu}_{\mathbf{J}_{i,:}}, \boldsymbol{\Sigma}_{\mathbf{J}}), \tag{7}$$

where $\boldsymbol{\mu}_{\mathbf{J}_{i,:}}$ and $\boldsymbol{\Sigma}_{\mathbf{J}}$ are respectively the posterior mean and covariance of the Jacobian given $\boldsymbol{\theta}$ and the mapping f (derivation in [1]).
Gradients of a function $g(\boldsymbol{\theta}) : \mathbb{R}^D \to \mathbb{R}$ can then be transformed as:

$$\nabla_{\omega} g(\mathbf{f}(\boldsymbol{\omega})) = \boldsymbol{\mu}_{\mathbf{J}} \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}). \tag{8}$$

AdaGeo Optimization

The **minimization** of an objective function $g(\boldsymbol{\theta})$:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} g(\boldsymbol{\theta}) \tag{1}$$

can be solved with **gradient-based** schemes of the form $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \Delta\boldsymbol{\theta}_t$ where $\Delta\boldsymbol{\theta}_t = \Delta\boldsymbol{\theta}_t(\nabla g(\boldsymbol{\theta}))$.
After t steps the GP-LVM is trained on the set $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_t\}$ and the updates are computed into the **latent space**.
E.g., a vanilla stochastic gradient descent:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\epsilon_t}{N_b} \sum_{i=1}^{N_b} \nabla_{\boldsymbol{\theta}} g(\boldsymbol{\theta}_{ti}) \tag{2}$$

becomes, using eq. 8 for the gradients:

$$\begin{aligned} \boldsymbol{\omega}_{t+1} &= \boldsymbol{\omega}_t - \frac{\epsilon_t}{N_b} \sum_{i=1}^{N_b} \nabla_{\omega} g(\boldsymbol{\omega}_{ti}) \\ \boldsymbol{\theta}_{t+1} &= f(\boldsymbol{\omega}_{t+1}). \end{aligned} \tag{3}$$

Latent and classic updates are **alternated** to acquire and exploit geometrical information.

Results

Sampling AdaGeo-SGLD was used to sample from a 50-dimensional **banana** distribution: **Metropolis-Hastings** yields the first 100 samples, after 10^4 burn-in iterations and with a thinning factor of 250. **AdaGeo-SGLD**, using a 5-dimensional latent space, returned the next 250 samples, with 10^3 burn-in steps and a thinning factor of 100. As seen in figure 1 the main features of the distribution are preserved. A relevant speed up and better autocorrelation plots are obtained as well.
Optimization First, the training of a one-layer **neural network** implementing **logistic regression** on the **MNIST** dataset was speeded up: the 7850 weights were modeled through a 9-dimensional latent space and **AdaGeo** was used to accelerate an **SGD** scheme, where 20 classic updates were alternated with 30 latent ones until convergence. Second, a **Gaussian Process** with a total of 9 hyper-parameters is trained using **gradient descent** with Nesterov momentum: here we alternate 15 classic updates with 15 latent ones, using a 3-dimensional latent space.

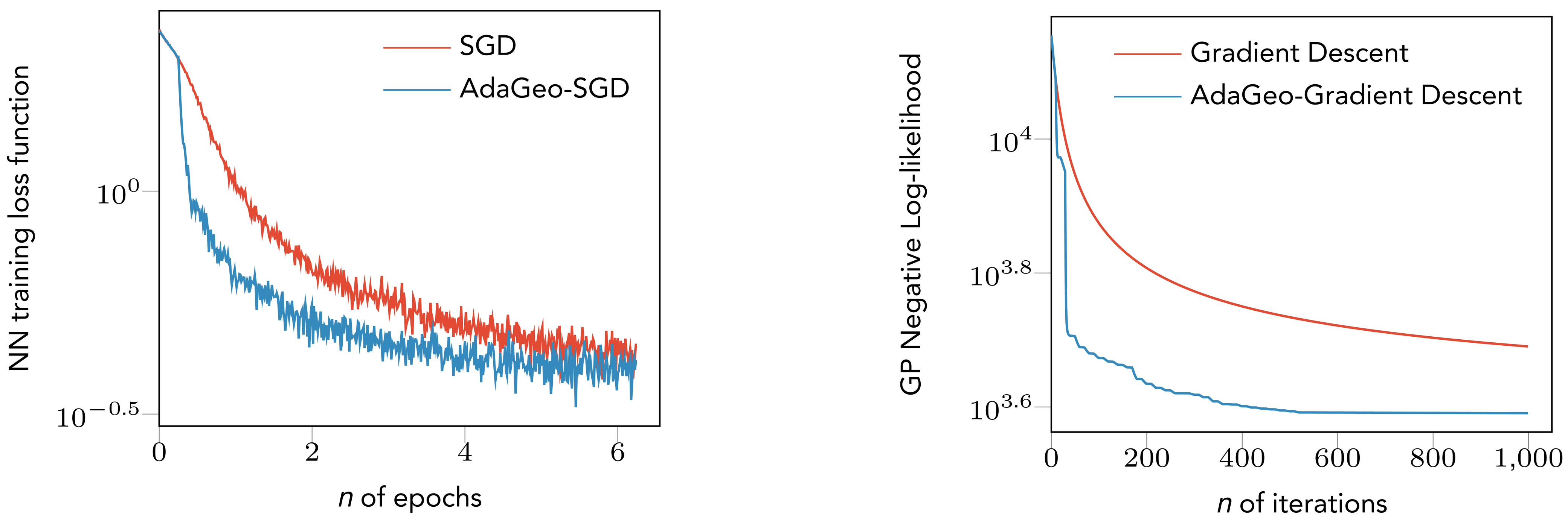

Fig. 2: Above: neural network loss function during training. Below: Negative log-likelihood during training of a Gaussian process.

AdaGeo-SGLD Sampling

Stochastic gradient Langevin dynamics (SGLD)[2] implements Bayesian **posterior sampling** of a distribution $p(x, \boldsymbol{\theta})$ by building an iterative method of the form:

$$\Delta\boldsymbol{\theta}_t = \frac{\epsilon_t}{2}\left(\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}_t) + \frac{N}{n}\sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}_i \mid \boldsymbol{\theta}_t)\right) + \boldsymbol{\eta}_t, \tag{4}$$

where $\boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \epsilon_t \mathbf{I})$. Analogously as the optimization case, the update can be brought onto the **lower-dimensional latent space**:

$$\Delta\boldsymbol{\omega}_t = \frac{\epsilon_t}{2}\left(\nabla_{\omega} \log p(\mathbf{f}(\boldsymbol{\omega}_t)) + \frac{N}{n}\sum_{i=1}^{n} \nabla_{\omega} \log p(\mathbf{x}_{ti} \mid \mathbf{f}(\boldsymbol{\omega}_t))\right) + \boldsymbol{\eta}_t, \tag{5}$$

where $\boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \epsilon_t \mathbf{I})$.

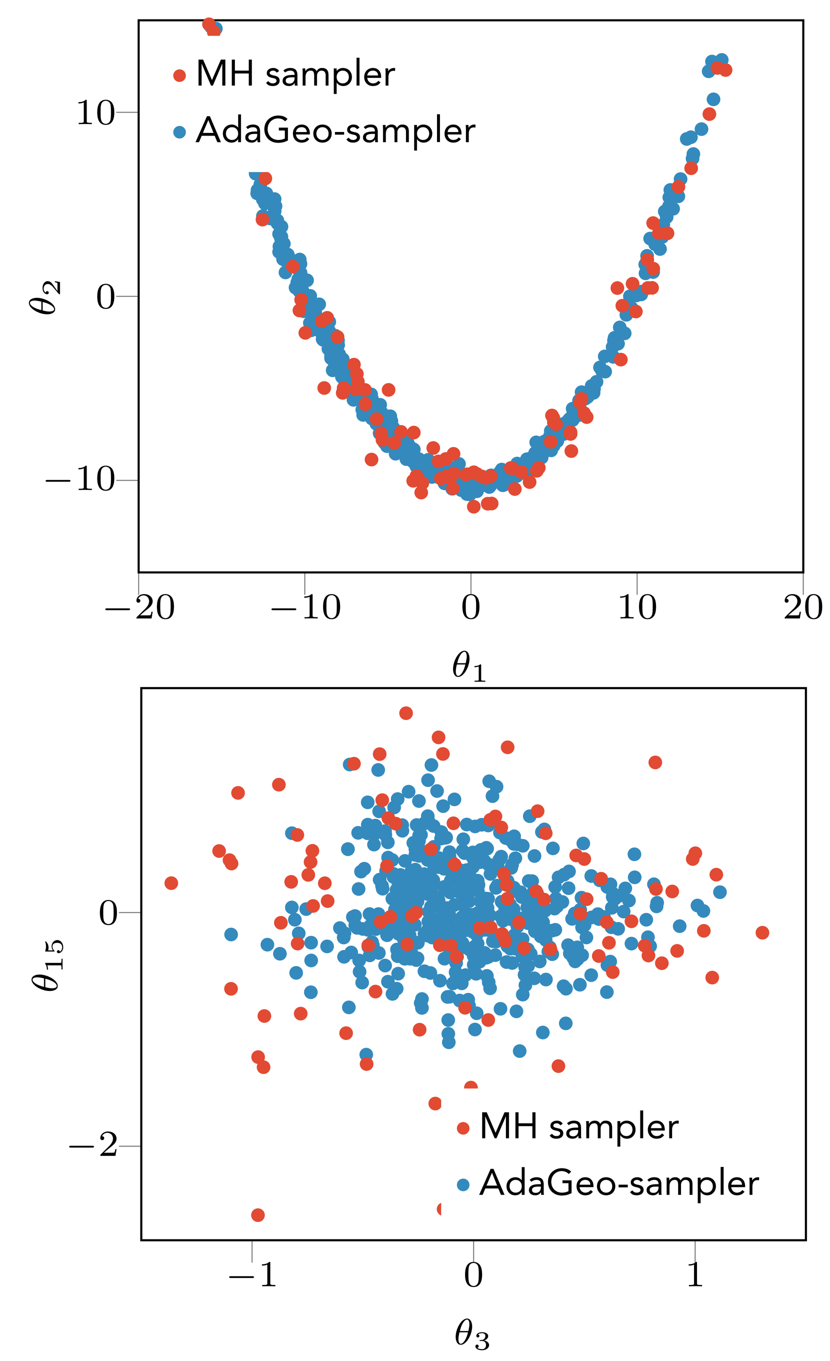

Fig. 1: Sampling with AdaGeo-SGLD from a 50-dimensional "banana" distribution.

Riemannian Extensions

On the more speculative side, with GP-LVM it is possible to access the distribution[1] of the latent Riemannian metric tensor $\mathbf{G} = \mathbf{J}^\top \mathbf{J}$:

$$\mathbf{G} \sim \mathcal{W}_Q\left(D, \boldsymbol{\Sigma}_{\mathbf{J}}, \mathbb{E}\left[\mathbf{J}^\top\right]\mathbb{E}[\mathbf{J}]\right)$$

where $\mathcal{W}$ denotes the non-central Wishart distribution, is a generalization to multiple dimensions of the χ^2-distribution.
Future works could include latent optimization and sampling on Riemannian manifolds (natural gradient descent, stochastic gradient Riemannian Langevin dynamics, etc.).

References

[1] Alessandra Tosi *et al.*, *Metrics for probabilistic geometries*, UAI, 2014

[2] Max Welling *et al.*, *Bayesian learning via stochastic gradient Langevin dynamics*, ICML, 2011

Contact

Gabriele Abbati
University of Oxford
Email: gabb@robots.ox.ac.uk
Website: http://www.robots.ox.ac.uk/~gabb/