A Variational Analysis of Stochastic Gradient Algorithms

Stephan Mandt[1], Matthew D. Hoffman[2], David M. Blei[1,3]

[1] Data Science Institute, Columbia University, USA
[2] Adobe Research, San Francisco, USA
[3] Departments of Computer Science and Statistics, Columbia University, USA

Introduction

- Stochastic Gradient Descent is an important algorithm. It minimizes an objective function $\mathcal{L}(\theta) = \sum_{i=1}^{N} \ell_i(\theta)$ based on the update

$$\theta_{t+1} = \theta_t - \epsilon \nabla_\theta \hat{\mathcal{L}}_S(\theta_t), \quad \hat{\mathcal{L}}_S(\theta) = \frac{N}{S} \sum_{i \in \mathcal{S}_t} \ell_i(\theta).$$

- Above, $\mathcal{S}_t \subset \{1, \dots, N\}$ is a random subset of indices of size S, drawn at time t which constitutes the *mini-batch*. We assume $N \gg S \gg 1$.
- When ϵ is constant, SGD does not converge to a point. Instead, the iterates of SGD converge *in distribution* to a stationary density.
- **Goal:** Analyze SGD for constant learning rates ϵ.
- **Intuition:** We interpret SGD as approximate inference and the sampling distribution as an approximation to a posterior (next column).
- **Method:** We use the formalism of stochastic differential equations.

Continuous-time limit of SGD revisited

A1 Assume that the gradient noise $\nabla \mathcal{L}_S(\theta) - \nabla \mathcal{L}(\theta)$ is Gaussian distributed.

A2 Assume that the iterates $\theta(t)$ are constrained to a small region s.th. the sampling noise covariance of the stochastic gradients is constant.

A3 Assume that the step size is small enough that we can approximate the discrete-time SGD algorithm with a continuous-time Markov process.

A4 Assume that the stationary distribution of the iterates is constrained to a region where the objective is approximately quadratic, $\mathcal{L}(\theta) = \frac{1}{2}\theta^\top A \theta$.

Comments on assumptions A1–A4.

- Assumption A1 can be justified by the central limit theorem. In formulas,

$$\nabla \hat{\mathcal{L}}_S(\theta) \approx \nabla \mathcal{L}(\theta) + \hat{\xi}_S(\theta), \; \hat{\xi}_S(\theta) \sim \mathcal{N}(0, C(\theta)/S),$$

$$\frac{C(\theta)}{S} \equiv \mathbb{E}\left[(\nabla \hat{\mathcal{L}}_S(\theta) - \nabla \mathcal{L}(\theta))(\nabla \hat{\mathcal{L}}_S(\theta) - \nabla \mathcal{L}(\theta))^\top \right].$$

- Based on A2, $C(\theta) \equiv C$ is constant.
  Write $C = BB^\top$ and define $B_{\epsilon/S} = \sqrt{\epsilon/S}\, B$.

$$\theta(t+1) - \theta(t) = -\epsilon \nabla \mathcal{L}(\theta(t)) + \sqrt{\epsilon}\, B_{\epsilon/S} W(t), \quad W(t) \sim \mathcal{N}(0, \mathbf{I}).$$

- Based on A3, this equation becomes a stochastic differential equation,

$$d\theta(t) = -\nabla_\theta \mathcal{L}(\theta) dt + B_{\epsilon/S}\, dW(t)$$

- Based on A4, we derive the **multivariate Ornstein-Uhlenbeck process**,

$$\boxed{d\theta(t) = -A\theta(t)dt + B_{\epsilon/S} dW(t).}$$

- This process approximates SGD under assumptions A1–A4.

Benefits of the Ornstein-Uhlenbeck Approximation

- Our approximation of SGD allows us to compute stationary distributions.
- Explicit formula for stationary distribution:

$$q(\theta) \propto \exp\left\{ -\tfrac{1}{2}\theta^\top \Sigma^{-1} \theta \right\}, \quad \Sigma A^\top + A\Sigma = \tfrac{\epsilon}{S} B B^\top.$$

- We can read-off how various parameters of SGD affect this distribution.

Main Result: Constant-rate SGD as approximate inference

- For many problems in machine learning (including neural networks), the objective has the interpretation of a negative log likelihood + log prior:

$$\mathcal{L}(\theta) = -\sum_{i=1}^{N} \log p(x_i|\theta) - \log p(\theta)$$

- The conventional goal of optimization is to find the minimum of $\mathcal{L}(\theta)$, but this may lead to wasted effort and overfitting. The exponentiated negative loss might capture just the right degree of parameter uncertainty:

$$f(\theta) \propto \exp\{-\mathcal{L}(\theta)\}$$

- **Idea:** Instead of minimizing the objective, let us aim to generate a single sample from this "posterior" (negative exponentiated objective).
- **Solution:** We run SGD with constant step size. With appropriate learning rates and minibatch sizes, the sampling distribution can be considered a proxy for the posterior! To this end, minimize

$$KL(q(\theta; \epsilon, S) || f(\theta)) \equiv \mathbb{E}_q[\log f(\theta)] - \mathbb{E}_q[\log q(\theta)].$$

Variational optimal learning parameters

- For sampling distibution $q(\theta) \propto \exp\left\{-\frac{1}{2}\theta^\top \Sigma^{-1} \theta\right\}$ and for posterior $f(\theta) \propto \exp\{-\frac{1}{2}\theta^\top A \theta\}$, we find

$$KL(q||f) = \tfrac{1}{2}\left(\mathrm{Tr}(A\Sigma) - \log A - \log|\Sigma| - d\right) \stackrel{c}{=} \tfrac{\epsilon}{2S}\mathrm{Tr}(BB^\top) - \log(\epsilon/S).$$

- Minimizing over ϵ yields $\epsilon^* = 2S/\mathrm{Tr}(BB^\top)$ for the optimal learning rate.
- We can derive a more complex result when allowing for a preconditioning matrix H, which gives the modified Ornstein-Uhlenbeck process

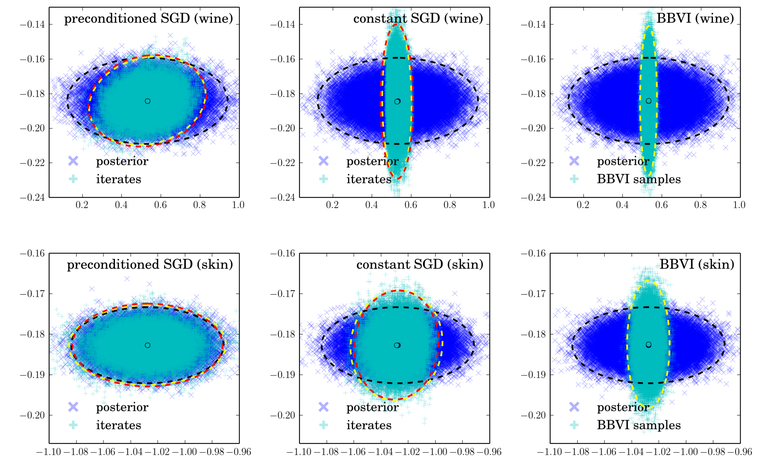
$$d\theta = -HA\theta(t)dt + HB_{\epsilon/S} dW(t).$$

- The KL divergence is for this more complex process is

$$KL = \tfrac{\epsilon}{2S}\mathrm{Tr}(BB^\top H) + \mathrm{Tr}\log(H) + \tfrac{1}{2}\log\tfrac{\epsilon}{S} - \log|\Sigma|.$$

- The optimal diagonal preconditioner is $H_k^* \propto 1/(2BB^\top)_{kk}$.
- This result relates to AdaGrad, but contains no square roots.

Experiments on real-world data



Secondary Result: Analyzing scalable MCMC algorithms

- Many modern MCMC algorithms are based on stochastic gradients
- We focus on Stochastic Gradient Fisher Scoring (Ahn et. al., 2012):

$$\theta_{t+1} = \theta_t - \epsilon H \nabla_\theta \hat{\mathcal{L}}(\theta_t) + \sqrt{\epsilon} H E W(t)$$

- Above, H is a preconditioner, $W(t)$ is a Gaussian noise, and E is a matrix-valued free parameter. Using assumptions A1–A4, this again becomes an Ornstein-Uhlenbeck process:

$$d\theta(t) = -HA\theta dt + H[B_\epsilon + E]dW(t).$$

- Minimizing KL justifies the optimal Fisher scoring preconditioner:

$$H^* = \tfrac{2}{N}(\epsilon BB^\top + EE^\top)^{-1}.$$

- This derivation is shorter and follows naturally from our formalism.
- We can furthermore quantify the bias due to a diagonal approximation.

Conclusion

- Stochastic differential equations are a powerful tool to analyze stochastic gradient-based algorithms.
- We can interpret SGD with constant learning rates as an approximate Bayesian sampling algorithm. Minimizing KL divergence to the true posterior leads to novel criteria for optimal parameters of SGD, where parameter uncertainty is taken into account.
- Using our formalism, we can analyze more complex algorithms. This will be presented elsewhere.