Stochastic Gradient Descent as Approximate Bayesian Inference

Fan Cheng

Central University of Finance and Economics

chengfan9477@163.com

June 8, 2018

Outline

1. Stochastic Gradient Descent
2. Main questions
3. Continuous-Time Limit Revisited
4. SGD as Approximate Inference
5. Analyzing Stochastic Gradient MCMC Algorithms
6. A Bayesian View on Iterate Averaging
7. Experiments
8. Conclusions

Stochastic Gradient Descent

↪ Introduction

- Stochastic gradient descent (SGD) has become crucial to modern machine learning.
- SGD optimizes a function by following noisy gradients with a decreasing step size.
- The classical result of Robbins and Monro (1951) is that this procedure provably reaches the optimum of the function (or local optimum, when it is nonconvex).
- Recent studies investigate the merits of adaptive step sizes, gradient or iterate averaging, and constant step-sizes.
- SGD has enabled efficient optimization with massive data, since one can often obtain noisy-but-unbiased gradients very cheaply by randomly subsampling a large dataset.

Stochastic Gradient Descent
↪ Introduction

- Stochastic gradients (SG) have also been used in the service of scalable Bayesian Markov Chain Monte Carlo (MCMC) methods.
- The goal is to generate samples from a conditional distribution of latent variables given a data set.
- In Bayesian inference, our goal is to approximate the posterior

$$p(\theta|x) = \exp(\log p(\theta, x) - \log p(x))$$

  where we assume a probabilistic model $p(\theta, x)$ with data x and hidden variables θ;
- Stochastic gradient MCMC algorithms employ stochastic gradients of log $p(\theta, x)$ to improve convergence and computation of existing sampling algorithms.
- There are some SG MCMC algorithms, such as SG Langevin dynamics (Welling and Teh, 2011), SG Hamiltonian Monte Carlo (Chen et al., 2014), SG thermostats (Ding et al., 2014), and SG Fisher scoring (Ahn et al., 2012)

Main questions

- What is the simplest modification to SGD that yields an efficient approximate Bayesian sampling algorithm?
- How can we construct other sampling algorithms based on variations of SGD such as preconditioning (Duchi et al., 2011; Tieleman and Hinton, 2012), momentum (Polyak, 1964) or Polyak averaging (Polyak and Juditsky, 1992)?

Constant SGD

- SGD with a constant learning rate (constant SGD) first marches toward an optimum of the objective function and then bounces around its vicinity. In contrast, traditional SGD converges to the optimum by decreasing the learning rate.
- Constant SGD is a stochastic process with a stationary distribution, one that is centered on the optimum and that has a certain covariance structure.
- The main idea is that we can use this stationary distribution to approximate a posterior.
- We apply constant SGD as though we were trying to minimize the negative log-joint probability $-\log p(\theta, x)$ over the model parameters θ.
- Constant SGD has several tunable parameters: the constant learning rate, the minibatch size, and the preconditioning matrix (if any) that we apply to the gradient updates.
- If we set these parameters appropriately, we can perform approximate Bayesian inference by simply running constant SGD.

Main contributions

- First, we develop a variational Bayesian view of stochastic gradient descent.
- We show that constant SGD gives rise to a new variational EM algorithm (Bishop, 2006) which allows us to use SGD to optimize hyperparameters while performing approximate inference in a Bayesian model.
- We use our formalism to derive the stationary distribution for SGD with momentum (Polyak, 1964).
- Then, we analyze scalable MCMC algorithms. Specifically, we use the stochastic-process perspective to compute the stationary distribution of Stochastic-Gradient Langevin Dynamics (SGLD) byWelling and Teh (2011) when using constant learning rates, and analyze stochastic gradient Fisher scoring (SGFS) by Ahn et al. (2012).
- Finally, we analyze iterate averaging (Polyak and Juditsky, 1992), where one successively averages the iterates of SGD to obtain a lower-variance estimator of the optimum.

SGD with a continuous-time stochastic process

$\looparrowright$ Problem Setup

- Consider loss functions of the following form:

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{n=1}^{N}\ell_n(\theta), \quad g(\theta) \equiv \nabla_\theta \mathcal{L}(\theta). \tag{2}$$

  where

  - $L(\theta) = L(\theta; x)$ is a loss function that depends on data x and parameters θ,
  - Each $l_n(\theta)$ is the contribution to the overall loss from a single observation x_n.

- When finding a maximum-a-posteriori estimate of a model, the contributions to the loss may be

$$\ell_n(\theta) = -\log p(x_n \,|\, \theta) - \frac{1}{N}\log p(\theta), \tag{3}$$

  where $p(\theta, x)$ is the likelihood and $p(\theta)$ is the prior.

- Let S be a set of S random indices drawn uniformly at random from the set $[1, ..., N]$, and we call S a "minibatch" of size S. We used the indexed functions to form a stochastic estimate of the loss and a stochastic gradient,

$$\hat{\mathcal{L}}_S(\theta) = \frac{1}{S}\sum_{n\in\mathcal{S}}\ell_n(\theta), \quad \hat{g}_S(\theta) = \nabla_\theta \hat{\mathcal{L}}_S(\theta). \tag{4}$$

SGD with a continuous-time stochastic process
↪ Problem Setup

- We use this stochastic gradient in the SGD update

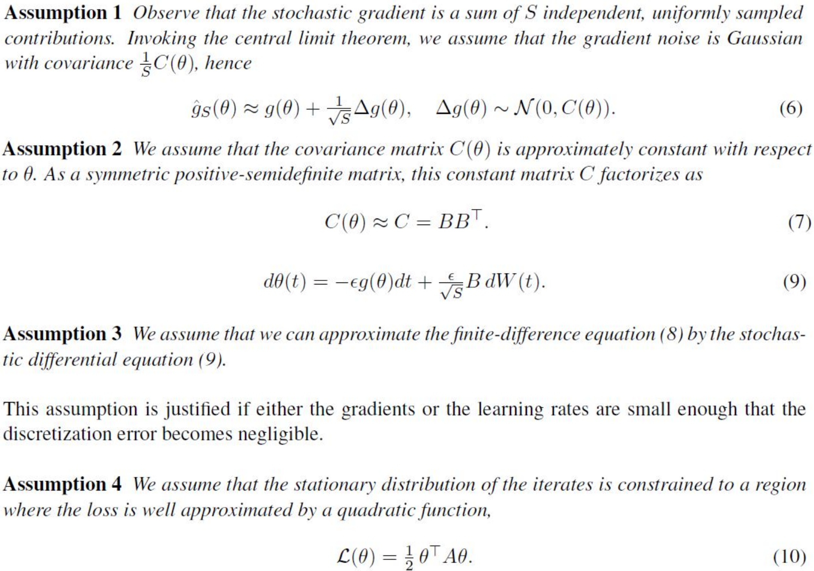
$$\theta(t+1) = \theta(t) - \epsilon\, \hat{g}_S(\theta(t)). \tag{5}$$

- These equations define the discrete-time process that SGD simulates from. We will approximate it with a continuous-time process that is easier to analyze.

SGD with a continuous-time stochastic process

↬ SGD as an Ornstein-Uhlenbeck Process

- To justify the approximation, we make four assumptions.



Assumption 1 *Observe that the stochastic gradient is a sum of S independent, uniformly sampled contributions. Invoking the central limit theorem, we assume that the gradient noise is Gaussian with covariance $\frac{1}{S}C(\theta)$, hence*

$$\hat{g}_S(\theta) \approx g(\theta) + \tfrac{1}{\sqrt{S}}\Delta g(\theta), \quad \Delta g(\theta) \sim \mathcal{N}(0, C(\theta)). \tag{6}$$

Assumption 2 *We assume that the covariance matrix $C(\theta)$ is approximately constant with respect to θ. As a symmetric positive-semidefinite matrix, this constant matrix C factorizes as*

$$C(\theta) \approx C = BB^\top. \tag{7}$$

$$d\theta(t) = -\epsilon g(\theta)dt + \tfrac{\epsilon}{\sqrt{S}}B\,dW(t). \tag{9}$$

Assumption 3 *We assume that we can approximate the finite-difference equation (8) by the stochastic differential equation (9).*

This assumption is justified if either the gradients or the learning rates are small enough that the discretization error becomes negligible.

Assumption 4 *We assume that the stationary distribution of the iterates is constrained to a region where the loss is well approximated by a quadratic function,*

$$\mathcal{L}(\theta) = \tfrac{1}{2}\,\theta^\top A\theta. \tag{10}$$

SGD with a continuous-time stochastic process
↪ SGD as an Ornstein-Uhlenbeck Process

- The four assumptions above result in a specific kind of stochastic process, the multivariate Ornstein-Uhlenbeck process (Uhlenbeck and Ornstein, 1930):

$$d\theta(t) = -\epsilon A\,\theta(t)dt + \frac{1}{\sqrt{S}}\epsilon B\,dW(t) \tag{11}$$

- This connection helps us analyze properties of SGD because the Ornstein-Uhlenbeck process has an analytic stationary distribution $q(\theta)$ that is Gaussian.

$$q(\theta) \propto \exp\left\{-\tfrac{1}{2}\theta^\top \Sigma^{-1}\theta\right\}. \tag{12}$$

where the covariance Σ satisfies

$$\Sigma A + A\Sigma = \tfrac{\epsilon}{S}BB^\top. \tag{13}$$

SGD with a continuous-time stochastic process

- We now discuss how to use constant SGD as an approximate inference algorithm.
- The classical goal of SGD is to minimize this loss, leading us to a maximum-a-posteriori point estimate of the parameters.
- In contrast, our goal here is to tune the parameters of SGD so that its stationary distribution approximates the posterior.

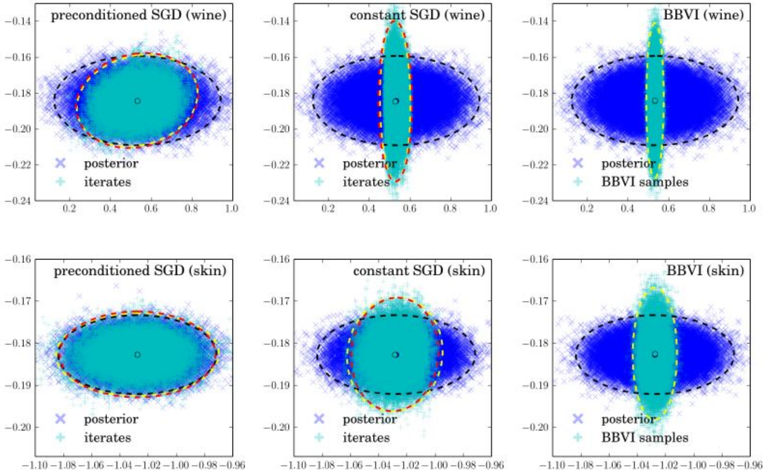


Figure 1: Posterior distribution $f(\theta) \propto \exp\{-N\mathcal{L}(\theta)\}$ (blue) and stationary sampling distributions $q(\theta)$ of the iterates of SGD (cyan) or black box variational inference (BBVI) based on reparameterization gradients. Rows: linear regression (top) and logistic regression (bottom) discussed in Section 7. Columns: full-rank preconditioned constant SGD (left), constant SGD (middle), and BBVI (Kucukelbir et al., 2015) (right). We show projections on the smallest and largest principal component of the posterior. The plot also shows the empirical covariances (3 standard deviations) of the posterior (black), the covariance of the samples (yellow), and their prediction (red) in terms of the Ornstein-Uhlenbeck process, Eq. 13.

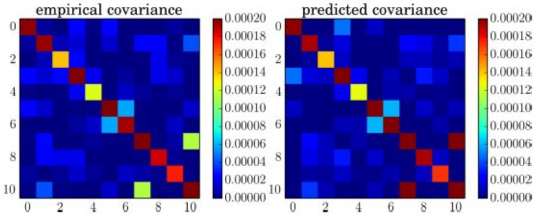


Figure 2: Empirical and predicted covariances of the iterates of stochastic gradient descent, where the prediction is based on Eq. 13. We used linear regression on the wine quality data set as detailed in Section 7.1.

Constant Stochastic Gradient Descent

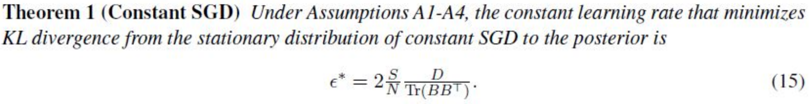
- First, we show how to tune constant SGD's parameters to minimize KL divergence to the posterior; this is a type of variational inference (Jordan et al., 1999).
- There are three versions of constant SGD—one with a constant step size, one with a full preconditioning matrix, and one with a diagonal preconditioning matrix.
- We will consider a more general SGD scheme that may involve a preconditioning matrix H instead of a scalar learning rate ϵ:

$$\theta_{t+1} = \theta_t - H\hat{g}_S(\theta(t)).$$

- We will set the parameters of SGD to minimize the KL divergence between the stationary distribution $q(\theta)$ (Eqs. 12 and 13) and the posterior $f(\theta)$ (Eq. 14):

$$\{H^*, S^*\} = \arg\min_{H,S} KL(q \,||\, f).$$

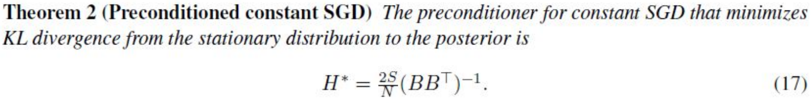
Constant Stochastic Gradient Descent



Theorem 1 (Constant SGD) *Under Assumptions A1-A4, the constant learning rate that minimizes KL divergence from the stationary distribution of constant SGD to the posterior is*

$$\epsilon^* = 2\frac{S}{N}\frac{D}{\mathrm{Tr}(BB^\top)}. \tag{15}$$

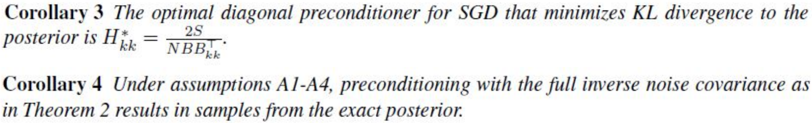
- Theorem 1 suggests that the learning rate should be chosen inversely proportional to the averag of diagonal entries of the noise covariance, and proportional to the ratio between the minibatch size and dataset size.



Theorem 2 (Preconditioned constant SGD) *The preconditioner for constant SGD that minimizes KL divergence from the stationary distribution to the posterior is*

$$H^* = \frac{2S}{N}(BB^\top)^{-1}. \tag{17}$$

- In high-dimensional applications, working with large dense matrices is impractical. In those settings we can constrain the preconditioner to be diagonal.



Corollary 3 *The optimal diagonal preconditioner for SGD that minimizes KL divergence to the posterior is* $H^*_{kk} = \frac{2S}{NBB^\top_{kk}}$.

Corollary 4 *Under assumptions A1-A4, preconditioning with the full inverse noise covariance as in Theorem 2 results in samples from the exact posterior.*

Stochastic Gradient with Momentum

- The continuous-time formalism also allows us to explore extensions of classical SGD.
- SGD with momentum doubles the dimension of parameter space in introducing an additional momentum variable v that has the same dimension as θ.
- The updates of SGD with momentum are

$$\begin{aligned} \mathrm{v}(t+1) &= (1-\mu)\mathrm{v}(t) - \epsilon\, \hat{g}_S(\theta(t)) \\ \theta(t+1) &= \theta(t) + \mathrm{v}(t+1). \end{aligned}$$

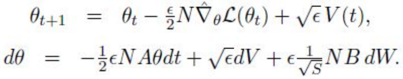
- This involves the damping coefficient $\mu \epsilon [0,1]$. For $\mu = 1$ (infinite damping or overdamping), the momentum information gets lost and we recover SGD.

$$\begin{aligned} d\mathrm{v} &= -\mu \mathrm{v} dt - \epsilon A\theta dt + \frac{1}{\sqrt{S}}\epsilon B\, dW, \qquad (23) \\ d\theta &= \mathrm{v} dt. \end{aligned}$$

- We have shown that ϵ, S, and μ play similar roles: only the combination affects the KL divergence to the posterior.

Analyzing Stochastic Gradient MCMC Algorithms
$\hookrightarrow$ SGLD with Constant Rates

- We analyze the well-known Stochastic Gradient Langevin Dynamics by Welling and Teh (2011).
- Limitations: the stochastic gradient noise vanishes as the learning rate goes to zero, and mixing becomes infinitely slow (Sato and Nakagawa, 2014).
- The discrete-time process that describes Stochastic Gradient Langevin dynamics is



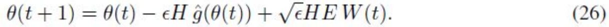
$$\begin{aligned} \theta_{t+1} &= \theta_t - \tfrac{\epsilon}{2} N \hat{\nabla}_\theta \mathcal{L}(\theta_t) + \sqrt{\epsilon}\, V(t), \\ d\theta &= -\tfrac{1}{2}\epsilon N A \theta dt + \sqrt{\epsilon} dV + \epsilon \tfrac{1}{\sqrt{S}} N B\, dW. \end{aligned}$$

where $V(t)$ is a vector of independent Gaussian noises.

Analyzing Stochastic Gradient MCMC Algorithms

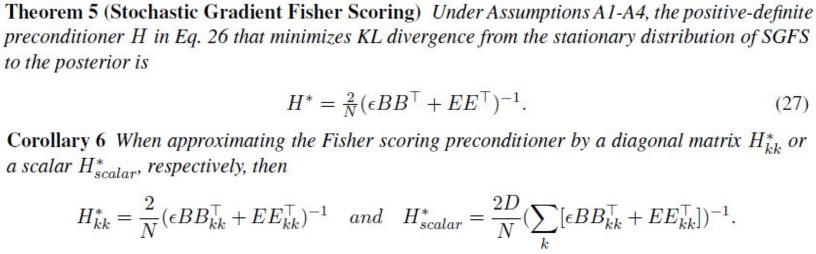
↛ Stochastic Gradient Fisher Scoring

- The basic idea here is that the stochastic gradient is preconditioned and additional noise is added to the updates such that the algorithm approximately samples from the Bayesian posterior.



$$\theta(t+1) = \theta(t) - \epsilon H\, \hat{g}(\theta(t)) + \sqrt{\epsilon} H E\, W(t). \qquad (26)$$

- The matrix H is a preconditioner and EW(t) is Gaussian noise; we control the preconditioner and the covariance of the noise.



Theorem 5 (Stochastic Gradient Fisher Scoring) *Under Assumptions A1-A4, the positive-definite preconditioner H in Eq. 26 that minimizes KL divergence from the stationary distribution of SGFS to the posterior is*

$$H^* = \tfrac{2}{N}(\epsilon B B^\top + E E^\top)^{-1}. \qquad (27)$$

Corollary 6 *When approximating the Fisher scoring preconditioner by a diagonal matrix H^*_{kk} or a scalar H^*_{scalar}, respectively, then*

$$H^*_{kk} = \frac{2}{N}(\epsilon BB^\top_{kk} + EE^\top_{kk})^{-1} \quad \textit{and} \quad H^*_{scalar} = \frac{2D}{N}(\sum_k [\epsilon BB^\top_{kk} + EE^\top_{kk}])^{-1}.$$

- An additional benefit of SGFS over simple constant SGD is that the sum of gradient noise and Gaussian noise will always look “more Gaussian” than the gradient noise on its own.

A Bayesian View on Iterate Averaging
$\looparrowright$ Iterate Averaging for Optimization

- We now apply our continuous-time analysis to the technique of iterate averaging (Polyak and Juditsky, 1992).
- Iterate averaging estimates the location of the minimum of L using a sequence of stochastic gradients g_s, and then computes an average of the iterates in an online manner,

$$\begin{aligned} \theta_{t+1} &= \theta_t - \epsilon \hat{g}_S(\theta_t), \\ \hat{\mu}_{t+1} &= \tfrac{t}{t+1}\,\hat{\mu}_t + \tfrac{1}{t+1}\,\theta_{t+1}. \end{aligned} \tag{28}$$

- After T stochastic gradient steps and going over to continuous times, this average is

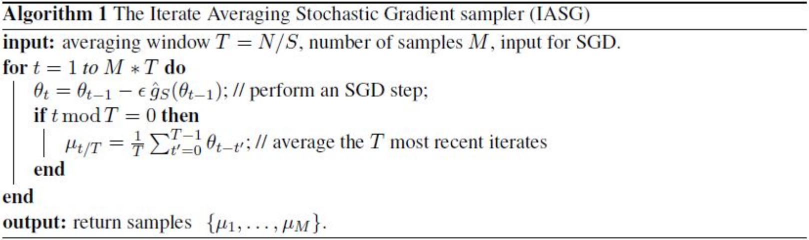
$$\hat{\mu} \approx \tfrac{1}{T}\int_0^T \theta(t)dt \equiv \hat{\mu}'. \tag{29}$$

- The average μ and its approximation are random variables whose expected value is the minimum of the objective..

A Bayesian View on Iterate Averaging

↪ Finite-Window Iterate Averaging for Posterior Sampling

- We will analyze iterate averaging as an algorithm for approximate posterior inference.
- iterate averaging requires exactly N gradient calls to generate one sample drawn from the exact posterior distribution, where N is the number of observations.



Algorithm 1 The Iterate Averaging Stochastic Gradient sampler (IASG)

```
input: averaging window $T = N/S$, number of samples $M$, input for SGD.
for $t = 1$ to $M * T$ do
    $\theta_t = \theta_{t-1} - \epsilon\, \hat{g}_S(\theta_{t-1})$; // perform an SGD step;
    if $t \bmod T = 0$ then
        $\mu_{t/T} = \frac{1}{T} \sum_{t'=0}^{T-1} \theta_{t-t'}$; // average the $T$ most recent iterates
    end
end
output: return samples $\{\mu_1, \ldots, \mu_M\}$.
```

- After T stochastic gradient steps and going over to continuous times, this average is

$$\hat{\mu} \approx \frac{1}{T} \int_0^T \theta(t) dt \equiv \hat{\mu}'. \tag{29}$$

- The average μ and its approximation are random variables whose expected value is the minimum sof the objective.

Experiments

↛ Data

- We confirm empirically that the stationary distributions of SGD with KL-optimal constant learning rates are as predicted by the Ornstein-Uhlenbeck process
- The Wine Quality Data Set, containing N = 4; 898 instances, 11 features, and one integer output variable (the wine rating).
- A data set of Protein Tertiary Structure, containing N = 45; 730 instances, 8 features and one output variable.
- The Skin Segmentation Data Set, containing N = 245; 057 instances, 3 features, and one binary output variable.

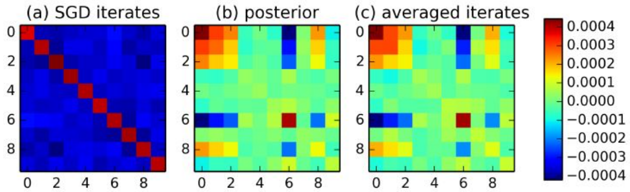


Figure 3: Iterate averaging on linear regression, where we generated artificial data as generated from the model. (a) shows the empirical covariance of the iterates of SGD, whereas (c) shows the averaged iterates with optimally chosen time window. The resulting covariance strongly resembles the

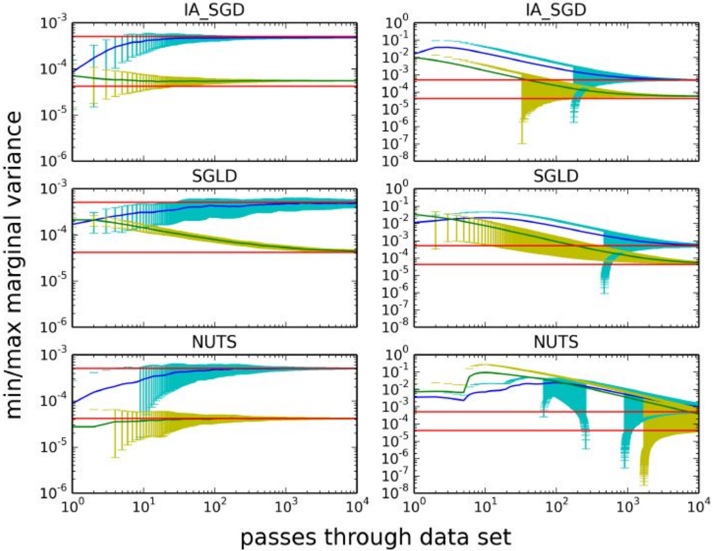


Figure 4: Convergence speed comparison between IASG (top), SGLD (middle), and NUTS (bottom) on linear regression. The plots show minimal (yellow) and maximal (blue) posterior marginal variances, respectively, as a function of iterations, measured in units of passes through the data. Error bars denote one standard deviation. Red solid lines show the ground truth. Left plots were

Iterate Averaging as Approximate MCMC

- Synthetic data: In order to strictly satisfy the assumptions outlined in Section 6.2, we generated artificial data that came from the model.
- We chose a linear regression model with a Gaussian prior with precision $\lambda = 1$. We first generated $N = 10; 000$ covariates by drawing them from a $D = 10$ dimensional Gaussian with unit covariance.

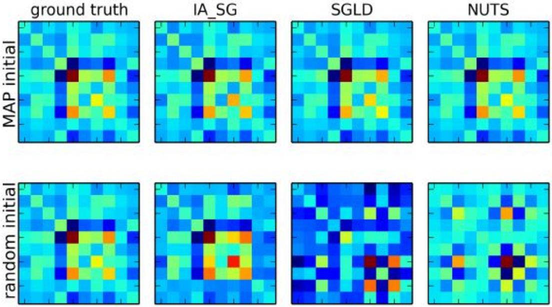


Figure 5: Posterior covariances as estimated by different methods, see also Fig. 4. The top row shows results where we initialized the samplers in the maximum posterior mode. The bottom rows were initialized randomly. For MAP initialization, all samplers find a good estimate of the posterior. When initializing randomly, IASG performs better than NUTS and SGLD.

Optimizing Hyperparameters

- We experimented with a Bayesian multinomial logistic (a.k.a. softmax) regression model with normal priors. The negative log-joint is

$$\begin{aligned}\mathcal{L} \equiv -\log p(y, \theta | x) &= \tfrac{\lambda}{2} \textstyle\sum_{d,k} \theta_{dk}^2 - \tfrac{DK}{2} \log(\lambda) + \tfrac{DK}{2} \log 2\pi \\ &+ \textstyle\sum_n \log \sum_k \exp\{\sum_d x_{nd}\theta_{dk}\} - \sum_d x_{nd}\theta_{dy_n},\end{aligned} \tag{35}$$

- In all experiments, we applied this model to the MNIST dataset (60; 000 training examples, 10; 000 test examples, 784 features) and the cover type dataset (500; 000 training examples, 81; 012 testing examples, 54 features).

Optimizing Hyperparameters

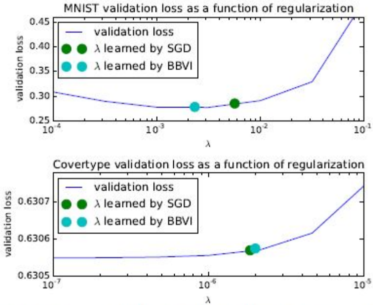


Figure 6: Validation loss as a function of L2 regularization parameter λ. Circles show the values of λ that were automatically selected by SGD and BBVI.

- The results suggest that BBVI and constant SGD yield similar results. Thus, constant SGD can be used as an inexpensive alternative to cross-validation or other VEM methods for hyperparameter selection.

Conclusions

- We built on a stochastic process perspective of stochastic gradient descent and various extensions to derive several new results.
- We analyzed SGD together with several extensions, such as momentum, preconditioning, and iterate averaging.
- The Bayesian view on constant-rate SGD allows us to use this algorithm as a new variational EM algorithm.
- Last, our analysis suggests the many similarities between sampling and optimization algorithms that can be explored using the stochastic process perspective.

Thank you!