A Variational Analysis of Stochastic Gradient Algorithms

Stephan Mandt SM3976@COLUMBIA.EDU
Columbia University, Data Science Institute, New York, USA

Matthew D. Hoffman MATHOFFM@ADOBE.COM
Adobe Research, San Francisco, USA

David M. Blei DAVID.BLEI@COLUMBIA.EDU
Columbia University, Departments of CS and Statistics, New York, USA

Abstract

Stochastic Gradient Descent (SGD) is an important algorithm in machine learning. With constant learning rates, it is a stochastic process that, after an initial phase of convergence, generates samples from a stationary distribution. We show that SGD with constant rates can be effectively used as an approximate posterior inference algorithm for probabilistic modeling. Specifically, we show how to adjust the tuning parameters of SGD such as to match the resulting stationary distribution to the posterior. This analysis rests on interpreting SGD as a continuous-time stochastic process and then minimizing the Kullback-Leibler divergence between its stationary distribution and the target posterior. (This is in the spirit of variational inference.) In more detail, we model SGD as a multivariate Ornstein-Uhlenbeck process and then use properties of this process to derive the optimal parameters. This theoretical framework also connects SGD to modern scalable inference algorithms; we analyze the recently proposed stochastic gradient Fisher scoring under this perspective. We demonstrate that SGD with properly chosen constant rates gives a new way to optimize hyperparameters in probabilistic models.

1. 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 (Duchi et al., 2011; Tieleman and Hinton, 2012), gradient or iterate averaging (Toulis et al.; Défossez and Bach, 2015), and constant step-sizes (Bach and Moulines, 2013; Flammarion and Bach, 2015). Stochastic gradient descent has enabled efficient optimization with massive data.

Recently, stochastic gradients (SG) have also been used in the service of scalable Bayesian Markov Chain Monte-Carlo (MCMC) methods, where the goal is to generate samples from a conditional distribution of latent variables given a data set. In Bayesian inference, we assume a probabilistic model $p(\theta, \mathbf{x})$ with data $\mathbf{x}$ and hidden variables θ; our goal is to approximate the posterior

$$p(\theta \,|\, \mathbf{x}) = \exp\{\log p(\theta, \mathbf{x}) - \log p(\mathbf{x})\}. \qquad (1)$$

New scalable 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)—employ stochastic gradients of $\log p(\theta, \mathbf{x})$ to improve convergence and computation of existing sampling algorithms. Also see Ma et al. (2015) for a complete classification of these algorithms.

These methods all take precautions to sample from an asymptotically exact posterior. In contrast to this and specifically in the limit of large data, we will show how to effectively use the simplest stochastic gradient descent algorithm as a sensible *approximate* Bayesian inference method. Specifically, we consider SGD with a constant learning rate (constant SGD). Constant SGD first marches toward an optimum of the objective function and then bounces around its vicinity because of the sampling noise in the gradient. (In contrast, traditional SGD converges to the optimum by decreasing the learning rate.) Our analy-

Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

sis below rests on the idea that constant SGD can be interpreted as 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.

Here is how it works. The particular profile of the stationary distribution depends on the parameters of the algorithm—the constant learning rate, the preconditioning matrix, and the minibatch size, all of which affect the noise and the gradients. Thus we can set $\log p(\theta, \mathbf{x})$ as the objective function and set the parameters of constant SGD such that its stationary distribution is close to the exact posterior (Eq. 1). Specifically, in the spirit of variational Bayes (Jordan et al., 1999b), we set those parameters to minimize the Kullback-Leibler (KL) divergence. With those settings, we can perform approximate inference by simply running constant SGD. In more detail, we make the following contributions:

- First, we develop a variational Bayesian view of stochastic gradient descent. Based on its interpretation as a continuous-time stochastic process—specifically a multivariate Ornstein-Uhlenbeck (OU) process (Uhlenbeck and Ornstein, 1930; Gardiner et al., 1985)—we compute stationary distributions for a large class of SGD algorithms, all of which converge to a Gaussian distribution with a non-trivial covariance matrix. The stationary distribution is parameterized by the learning rate, minibatch size, and preconditioning matrix.

  Results about the multivariate OU process enable us to compute the KL divergence between the stationary distribution and the posterior analytically. Minimizing the KL, we can relate the optimal step size or preconditioning matrix to the Hessian and noise covariances near the optimum. The resulting criteria strongly resemble AdaGrad (Duchi et al., 2011), RMSProp (Tieleman and Hinton, 2012), and classical Fisher scoring (Longford, 1987). We demonstrate how these different optimization methods compare, when used for approximate inference.

- Then, we analyze scalable MCMC algorithms. Specifically, we use the stochastic process perspective to compute the stationary distribution of stochastic gradient Fisher scoring (SGFS) by Ahn et al. (2012). The view from the multivariate OU process reveals a simple justification for this method: we show that the preconditioning matrix suggested in SGFS is indeed optimal. We also derive a criterion for the free noise parameter in SGFS such as to enhance numerical stability, and we show how the stationary distribution is modified when the preconditioner is approximated with a diagonal matrix (as is often done in practice for high-dimensional problems).

- Finally, we show how using SGD with a constant learning rate confers an important practical advantage: it allows simultaneous inference of the posterior and optimization of meta-level parameters, such as hyperparameters in a Bayesian model. We demonstrate this technique on a Bayesian multinomial logistic regression model with normal priors.

Our paper is organized as follows. In section 2 we review the continuous-time limit of SGD, showing that it can be interpreted as an OU process. In section 3 we present consequences of this perspective: the interpretation of SGD as variational Bayes and results around stochastic gradient Fisher Scoring (Ahn et al., 2012). In the empirical study (section 4), we show that our theoretical assumptions are satisfied for different models, and that we can use SGD to perform gradient-based hyperparameter optimization.

2. Continuous-Time Limit Revisited

We first review the theoretical framework that we use throughout the paper. Our goal is to characterize the behavior of SGD when using a constant step size. To do this, we approximate SGD with a continuous-time stochastic process (Kushner and Yin, 2003; Ljung et al., 2012).

2.1. Problem setup

Consider loss functions of the following form:

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

Such loss functions are common in machine learning, where $\mathcal{L}(\theta) \equiv \mathcal{L}(\theta, x)$ is a loss function that depends on data x and parameters θ. Each $\ell_n(\theta) \equiv \ell(\theta, x_n)$ is the contribution to the overall loss from a single observation. For example, when finding a maximum-a-posteriori estimate of a model, the contributions to the loss may be

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

where $p(x_n \,|\, \theta)$ is the likelihood and $p(\theta)$ is the prior. For simpler notation, we will suppress the dependence of the loss on the data.

From this loss we construct stochastic gradients. Let $\mathcal{S}$ be a set of S random indices drawn uniformly at random from the set $\{1, \ldots, N\}$. This set indexes functions $\ell_n(\theta)$, and we call $\mathcal{S}$ a "minibatch" of size S. Based on the minibatch, we used the indexed functions to form a stochastic estimate of the loss and a stochastic gradient,

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

In expectation the stochastic gradient is the full gradient, i.e., $g(\theta) = \mathbb{E}[\hat{g}_S(\theta)]$. We use this stochastic gradient in the SGD update

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

Equations 4 and 5 define the discrete-time process that SGD simulates from. We will approximate it with a continuous-time process that is easier to analyze.

2.2. SGD as a Ornstein-Uhlenbeck process

We now show how to approximate the discrete-time Eq. 5 with the continuous-time Ornstein-Uhlenbeck process (Uhlenbeck and Ornstein, 1930). This leads to the stochastic differential equation below in Eq. 11. To justify the approximation, we make four assumptions. We verify its accuracy in Section 4.

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 variance $\propto 1/S$:

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

Assumption 2. We assume that the noise covariance is approximately constant. Further, we decompose the constant noise covariance into a product of two constant matrices:

$$C = BB^\top. \tag{7}$$

This assumption is justified when the iterates of SGD are confined to a small enough region around a local optimum of the loss (e.g. due to a small ϵ) such that the noise covariance does not vary significantly in that region.

Assumption 3. We now define $\Delta\theta(t) = \theta(t+1) - \theta(t)$ and combine Eqs. 5, 6, and 7 to rewrite the process as

$$\Delta\theta(t) = -\epsilon\, g(\theta(t)) + \sqrt{\frac{\epsilon}{S}}\, B\, \Delta W, \quad \Delta W \sim \mathcal{N}\left(0, \epsilon \mathbf{I}\right). \tag{8}$$

This is a discretization of the following continuous-time stochastic differential equation: [1]

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

We assume that this continuous-time limit is approximately justified and that we can neglect the discretization errors.

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

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

(Without loss of generality, we assume that a minimum of the loss is at $\theta = 0$.) This assumption makes sense when the loss function is smooth and the stochastic process reaches a low-variance quasi-stationary distribution around a deep local minimum. The exit time of a stochastic process is typically exponential in the height of the barriers between minima, which can make local optima very stable even in the presence of noise (Kramers, 1940).

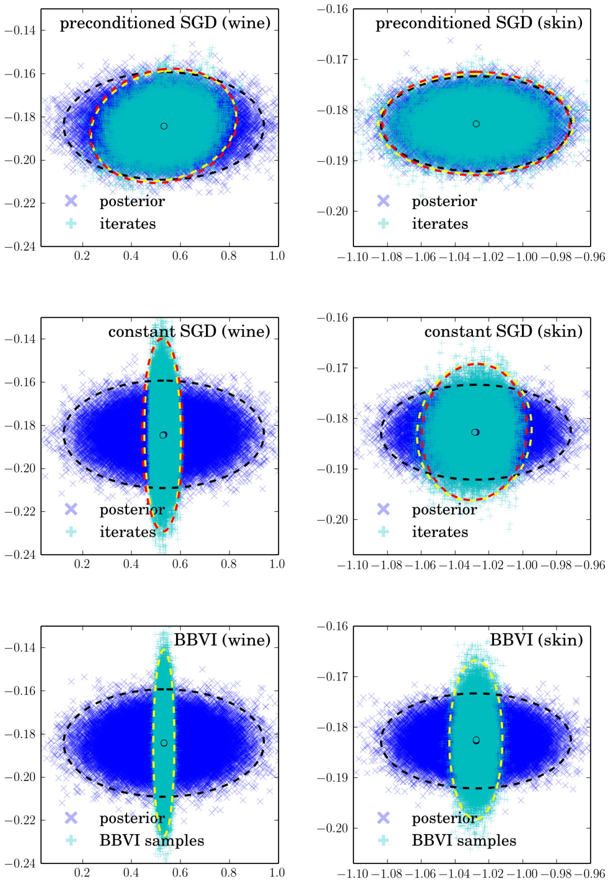

Figure 1: Posterior distribution $f(\theta) \propto \exp\{-N\mathcal{L}(\theta)\}$ (blue) and stationary sampling distributions $q(\theta)$ of the iterates of SGD (cyan) or black box variational inference (BBVI). Columns: linear regression (left) and logistic regression (right) discussed in Section 4. Rows: full-rank preconditioned constant SGD (top), constant SGD (middle), and BBVI (Kucukelbir et al., 2015) (bottom). 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.

SGD as an Ornstein-Uhlenbeck process. For what follows, define $B_{\epsilon/S} = \sqrt{\frac{\epsilon}{S}}B$. The four assumptions above result in a specific kind of stochastic process, the multivari-

[1] We performed the conventional substitution rules when discretizing a continuous-time stochastic process. These substitution rules are $\Delta\theta(t) \to d\theta(t)$, $\epsilon \to dt$ and $\Delta W \to dW$, see e.g. (Gardiner et al., 1985).

Method	Wine	Skin	Protein
constant SGD	18.7	0.471	1000.9
constant SGD-d	14.0	0.921	678.4
constant SGD-f	0.7	0.005	1.8
SGLD (Welling and Teh, 2011)	2.9	0.905	4.5
SGFS-d (Ahn et al., 2012)	12.8	0.864	597.4
SGFS-f (Ahn et al., 2012)	0.8	0.005	1.3
BBVI (Kucukelbir et al., 2015)	44.7	5.74	478.1

Table 1: KL divergences between the posterior and stationary sampling distributions applied to the data sets discussed in Section 4.1. We compared constant SGD without preconditioning and with diagonal (-d) and full rank (-f) preconditioning against Stochastic Gradient Langevin Dynamics and Stochastic Gradient Fisher Scoring (SGFS) with diagonal (-d) and full rank (-f) preconditioning, and BBVI.

ate *Ornstein-Uhlenbeck* process (Uhlenbeck and Ornstein, 1930). It is

$$d\theta(t) = -A\,\theta(t)dt + B_{\epsilon/S}\,dW(t) \tag{11}$$

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

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

The covariance Σ satisfies

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

Without explicitly solving this equation, we see that the resulting covariance is proportional to the learning rate ϵ and inversely proportional to the magnitude of A and minibatch size S. (More details are in the Appendix.) This characterizes the stationary distribution of running SGD with a constant step size.

3. SGD as Approximate Inference

We discussed a continuous-time interpretation of SGD with a constant step size (constant SGD). We now discuss how to use constant SGD as an approximate inference algorithm. To repeat the set-up from the introduction, consider a probabilistic model $p(\theta, \mathbf{x})$ with data $\mathbf{x}$ and hidden variables θ; our goal is to approximate the posterior in Eq. 1.

We set the loss to be proportional to the negative log-joint distribution (Eqs. 2 and 3), which equals the negative log posterior up to an additive constant. The classical goal of SGD is to minimize this loss, leading us to a maximum-a-posteriori point estimate of the parameters. This is how SGD is used in many statistical models, including logistic regression, linear regression, matrix factorization, neural network classifiers, and regressors. In contrast, our goal here is to tune the parameters of SGD such that we approximate the posterior with its stationary distribution. Thus we use SGD as a posterior inference algorithm.

Fig. 1 shows an example. Here we illustrate two Bayesian posteriors—from a linear regression problem (left) and a logistic regression problem (right)—along with iterates from a constant SGD algorithm. In these figures, we set the parameters of the optimization to values that minimize the Kullback-Leibler (KL) divergence between the stationary distribution of the OU process and the posterior—these results come from our theorems below. The top plots optimize both a preconditioning matrix and the step size; the middle plots optimize only the step size. (The middle plots are from a more efficient algorithm, but it is less accurate.) We can see that the stationary distribution of constant SGD can be made close to the exact posterior.

Fig. 1 also compares the empirical covariance of the iterates with the predicted covariance in terms of Eq. 13. The close match supports the assumptions of Sec. 2.

We will use this perspective in three ways. First, we develop optimal conditions for constant SGD to best approximate the posterior, connecting to well-known results around adaptive learning rates and preconditioners. Second, we use it to analyze Stochastic Gradient Fisher Scoring (Ahn et al., 2012), both in its exact form and its more efficient approximate form. Third, we propose an algorithm for hyperparameter optimization based on constant SGD.

3.1. 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., 1999a). This analysis leads to three versions of constant SGD—one with a constant step size, one with a full preconditioning matrix, and one with a diagonal preconditioning matrix. Each yields samples from an approximate posterior, and each trades off efficiency and accuracy differently. Finally, we show how to use these algorithms to learn hyperparameters.

Assumption 4 from Sec. 2 says that the posterior is approximately Gaussian in the region that the stationary distribution focuses on,

$$f(\theta) \propto \exp\left\{-\tfrac{N}{2}\theta^\top A\theta\right\}. \tag{14}$$

(The scalar N corrects the averaging in equation 2.) In setting the parameters of SGD, we minimize the KL divergence between the posterior $f(\theta)$ and the stationary distribution $q(\theta)$ (Eqs. 12 and 13) as a function of the learning rate ϵ and minibatch size S. We can optionally include a *preconditioning matrix* H, i.e. a matrix that premultiplies the stochastic gradient to modify its convergence behavior. Hence, we minimize

$$\{\epsilon^*, S^*, H^*\} = \arg\min_{\epsilon,S,H} KL(q(\theta) \parallel f(\theta)). \tag{15}$$

First, consider the case without H. The distributions $f(\theta)$ and $q(\theta)$ are both Gaussians. Their means coincide, at the minimum of the loss, and so their KL divergence is

$$\begin{aligned} KL(q\|f) &= \mathbb{E}_{q(\theta)}[\log f(\theta)] - \mathbb{E}_{q(\theta)}[\log q(\theta)] \\ &= \tfrac{1}{2}\left(N\mathrm{Tr}(A\Sigma) - \log|NA| - \log|\Sigma| - D\right), \end{aligned} \tag{16}$$

where $|\cdot|$ is the determinant and D is the dimension of θ.

We suggest three variants of constant SGD that generate samples from an approximate posterior.

Theorem 1 (constant SGD). *Under assumptions A1-A4, the constant learning rate minimizing KL divergence from the stationary distribution of SGD to the posterior is*

$$\epsilon^* = \frac{2DS}{N\mathrm{Tr}(BB^\top)}. \tag{17}$$

To prove this claim, we face the problem that the covariance of the stationary distribution depends indirectly on ϵ through Eq. 13. Inspecting this equation reveals that $\Sigma_0 \equiv \frac{S}{\epsilon}\Sigma$ is independent of S and ϵ. This simplifies the entropy term $\log|\Sigma| = D\log(\epsilon/S) + \log|\Sigma_0|$. Since Σ_0 is constant, we can neglect it when minimizing KL divergence.

We also need to simplify the term $\mathrm{Tr}(A\Sigma)$, which still depends on ϵ and S through Σ. To do this, we again use Eq. 13, from which follows that $\mathrm{Tr}(A\Sigma) = \frac{1}{2}(\mathrm{Tr}(A\Sigma) + \mathrm{Tr}(\Sigma A^\top)) = \frac{\epsilon}{2S}\mathrm{Tr}(BB^\top)$. The KL divergence is therefore, up to constant terms,

$$KL(q\|f) \overset{c}{=} \frac{\epsilon N}{2S}\mathrm{Tr}(BB^\top) - D\log(\epsilon/S) \tag{18}$$

Minimizing KL divergence over ϵ/S results in Eq. 17 for the optimal learning rate. □

Theorem 1 suggests that the learning rate should be chosen inversely proportional to the average of diagonal entries of the noise covariance. We can also precondition SGD with a matrix H. This gives more tuning parameters to better approximate the posterior. Under the same assumptions, we ask for the optimal preconditioner.

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

$$H^* = \frac{2S}{\epsilon N}(BB^\top)^{-1} \tag{19}$$

To prove this claim, we need the Ornstein-Uhlenbeck process which corresponds to preconditioned SGD. Preconditioning Eq. 11 with H results in

$$d\theta(t) = -HA\,\theta(t)dt + HB_{\epsilon/S}\,dW(t). \tag{20}$$

All our results carry over after substituting $A \leftarrow HA$, $B \leftarrow HB$. Eq. 13, after the transformation and multiplication by H^{-1} from the left, becomes

$$A\Sigma + H^{-1}\Sigma A^\top H = \frac{\epsilon}{S}BB^\top H \tag{21}$$

Using the cyclic property of the trace, this implies that $\mathrm{Tr}(A\Sigma) = \frac{1}{2}(\mathrm{Tr}(A\Sigma) + \mathrm{Tr}(H^{-1}A\Sigma H) = \frac{\epsilon}{2S}\mathrm{Tr}(BB^\top H)$. Hence up to constant terms, the KL divergence is

$$\begin{aligned} KL(q\|f) &\overset{c}{=} \frac{\epsilon N}{2S}\mathrm{Tr}(BB^\top H) + \frac{1}{2}\log\left(\frac{\epsilon}{S}|H\Sigma^{-1}H|\right) \\ &= \frac{\epsilon N}{2S}\mathrm{Tr}(BB^\top H) + \mathrm{Tr}\log(H) + \frac{D}{2}\log\frac{\epsilon}{S} - \frac{1}{2}\log|\Sigma|. \end{aligned} \tag{22}$$

(We used that $\log(\det H) = \mathrm{Tr}\log H$.) Taking derivatives with respect to the entries of H results in Eq. 19. □

In high-dimensional applications where working with large dense matrices is impractical, the preconditioner may be constrained to be diagonal. The following corollary is a direct consequence of Eq. 22:

Corollary 1 *The optimal diagonal preconditioner for SGD that minimizes KL divergence is* $H^*_{kk} = \frac{2S}{\epsilon NBB^\top_{kk}}$. We showed that the optimal diagonal preconditioner is the inverse of the diagonal part of the noise matrix. Similar preconditioning matrices have been suggested earlier in optimal control theory based on very different arguments, see (Widrow and Stearns, 1985). Our result also relates to AdaGrad and its relatives (Duchi et al., 2011; Tieleman and Hinton, 2012), which also adjust the preconditioner based on the square root of the diagonal entries of the noise covariance. In the supplement we derive an optimal global learning rate for AdaGrad-style diagonal preconditioners.

In Sec. 4, we compare three versions of constant SGD for approximate posterior inference: one with a scalar step size, one with a dense preconditioner, and one with a diagonal preconditioner.

3.2. Stochastic Gradient Fisher Scoring

We now investigate Stochastic Gradient Fisher Scoring (Ahn et al., 2012), a scalable Bayesian MCMC algorithm. We use the variational perspective to rederive the Fisher scoring update and identify it as optimal. We also analyze the sampling distribution of the truncated algorithm, one with diagonal preconditioning (as it is used in practice), and quantify the bias that this induces.

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. More precisely, the update is

$$\theta(t+1) = \theta(t) - \epsilon H\,\hat{g}(\theta(t)) + \sqrt{\epsilon}HE\,W(t). \tag{23}$$

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

$EE^\top$ of the noise. Stochastic gradient Fisher scoring suggests a preconditioning matrix H that leads to samples from the posterior even if the learning rate ϵ is not asymptotically small. We show here that this preconditioner follows from our variational analysis.

Theorem 3 (SGFS) *Under assumptions A1-A4, the preconditioner H in Eq. 23 that minimizes KL divergence from the stationary distribution of SGFS to the posterior is*

$$H^* = \tfrac{2}{N}(\epsilon BB^\top + EE^\top)^{-1}. \tag{24}$$

To prove the claim, we go through the steps of section 2 to derive the corresponding Ornstein-Uhlenbeck process, $d\theta(t) = -HA\theta(t)dt + H\left[B_\epsilon + E\right]dW(t)$. For simplicity, we have set the minibatch size S to 1, hence $B_\epsilon \equiv \sqrt{\epsilon}B$. In the supplement, we derive the following KL divergence between the posterior and the sampling distribution: $KL(q||p) = -\frac{N}{4}\mathrm{Tr}(H(B_\epsilon B_\epsilon^\top + EE^\top)) + \frac{1}{2}\log|T| + \frac{1}{2}\log|H| + \frac{1}{2}\log|NA| + \frac{D}{2}$. We can now minimize this KL divergence over the parameters H and E. When E is given, minimizing over H gives Eq. 24 □.

The solution given in Eq. 24 not only minimizes the KL divergence, but makes it 0, meaning that the stationary sampling distribution *is* the posterior. This solution corresponds to the suggested Fisher Scoring update in the idealized case when the sampling noise distribution is estimated perfectly (Ahn et al., 2012). Through this update, the algorithm thus generates posterior samples without decreasing the learning rate to zero. (This is in contrast to Stochastic Gradient Langevin Dynamics by Welling and Teh (2011).)

In practice, however, SGFS is often used with a diagonal approximation of the preconditioning matrix (Ahn et al., 2012; Ma et al., 2015). However, researchers have not explored how the stationary distribution is affected by this truncation, which makes the algorithm only approximately Bayesian. We can quantify its deviation from the exact posterior and we derive the optimal diagonal preconditioner, which follows from the KL divergence in theorem 3:

Corollary 2 (approximate SGFS). *When approximating the Fisher scoring preconditioner by a diagonal matrix or a scalar, respectively, then* $H^*_{kk} = \frac{2}{N}(\epsilon BB^\top_{kk} + EE^\top_{kk})^{-1}$ *and* $H^*_{scalar} = \frac{2D}{N}(\sum_k[\epsilon BB^\top_{kk} + EE^\top_{kk}])^{-1}$*, respectively.*

Note that we have not made any assumptions about the noise covariance E. We can adjust it in favor of a more stable algorithm. This can be achieved by setting a maximum step size h^{max}, so that $H_{kk} \leq h^{max}$ for all k. We can adjust E such that $H_{kk} \equiv h^{max}$ in Eq. 24 becomes independent of k. Solving for E yields $EE^\top_{kk} = \frac{2}{h^{max}N} - \epsilon BB^\top_{kk}$.

Hence, to keep the learning rates bounded in favor of stability, one can inject noise in dimensions where the variance of the gradient is too small. This guideline is opposite to the advice of Ahn et al. (2012) to choose B proportional to E, but follows naturally from the variational analysis.

3.3. A new VEM algorithm for hyperparameter tuning

One of the major benefits to the Bayesian approach is the ability to fit hyperparameters to data without expensive cross-validation runs by placing hyperpriors on those hyperparameters. In Empirical Bayes (or type-II maximum likelihood), we maximize the *marginal* likelihood of the data, integrating out the main model parameters:

$$\lambda^\star = \arg\max_\lambda \log p(y|x,\lambda) = \arg\max_\lambda \log \int_\theta p(y,\theta|x,\lambda)d\theta.$$

When this marginal log-likelihood is intractable, a common approach is to use *variational expectation-maximization (VEM)* (Bishop, 2006), which iteratively optimizes a variational lower bound on the marginal log-likelihood over λ. If we approximate the posterior $p(\theta|x,y,\lambda)$ with some distribution $q(\theta)$, then VEM tries to find a value for λ that maximizes the expected log-joint probability $\mathbb{E}_q[\log p(\theta, y|x,\lambda)]$.

Define $\mathcal{L}(\theta,\lambda) \equiv -\log p(y,\theta\,|\,x,\lambda)$. If we interpret the stationary distribution of SGD as a variational approximation to a model's posterior, we can justify following a stochastic gradient descent scheme on both θ and λ:

$$\theta_{t+1} = \theta_t - \epsilon^*\nabla_\theta\mathcal{L}(\theta_t,\lambda_t); \quad \lambda_{t+1} = \lambda_t - \rho_t\nabla_\lambda\mathcal{L}(\theta_t,\lambda_t). \tag{25}$$

While the update for θ uses the optimal constant learning rate ϵ^* and therefore samples from an approximate posterior, the λ update uses a decreasing learning rate ρ_t and therefore converges to a local optimum. The result is thus a novel type of VEM algorithm.

We stress that the optimal constant learning rate ϵ^* is not unknown, but can be estimated. It relies on an online estimate of the gradient noise covariance which can be computed based on a mini-batch (Ahn et al., 2012). In Sec. 4 we show that gradient-based hyperparameter learning is a cheap alternative to cross-validation.

4. Experiments

We test our theoretical assumptions in section 4.1 and find good experimental evidence that they are correct. In this section, we compare against other approximate inference algorithms. In section 4.2 we show that constant SGD lets us optimize hyperparameters in a Bayesian model.

4.1. Confirming the stationary covariance

In this section, we confirm empirically that the stationary distributions of SGD with KL-optimal constant learning rates are as predicted by the Ornstein-Uhlenbeck process.

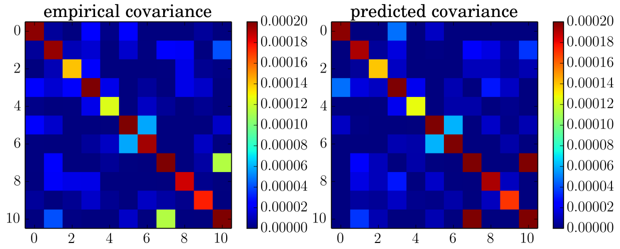

Figure 2: Empirical and predicted covariances of the iterates of stochastic gradient descent, where the prediction is based on Eq. 13 . We used linear regression on the wine quality data set as detailed in Section 4.1.

Data. We first considered the following data sets. (1) The *Wine Quality Data Set*[2], containing $N = 4898$ instances, 11 features, and one integer output variable (the wine rating). (2) A data set of *Protein Tertiary Structure*[3], containing $N = 45730$ instances, 8 features and one output variable. (3) The *Skin Segmentation Data Set*[4], containing $N = 245057$ instances, 3 features, and one binary output variable. We applied linear regression on data sets 1 and 2 and applied logistic regression on data set 3. We rescaled the feature to unit length and used a mini-batch of sizes $S = 100$, $S = 100$ and $S = 10000$, respectively. The quadratic regularizer was 1. The constant learning rate was adjusted according to Eq. 17.

Fig. 1 shows two-dimensional projections of samples from the posterior (blue) and the stationary distribution (cyan), where the directions were chosen two be the smallest and largest principal component of the posterior. Both distributions are approximately Gaussian and centered around the maximum of the posterior. To check our theoretical assumptions, we compared the covariance of the sampling distribution (yellow) against its predicted value based on the Ornstein-Uhlenbeck process (red), where very good agreement was found. The accuracy of the predicted covariance suggests that our modeling assumptions are reasonable here. The unprojected 11-dimensional covariances on wine data are also compared in Fig. 2. The bottom row of Fig. 1 shows the sampling distributions of black box variational inference (BBVI) using the reparametrization trick (Kucukelbir et al., 2015). Our results show that the approximation to the posterior given by constant SGD is not worse than the approximation given by BBVI.

We also computed KL divergences between the posterior and stationary distributions of various algorithms: constant SGD with KL-optimal learning rates and preconditioners, Stochastic Gradient Langevin Dynamics, Stochastic Gradient Fisher Scoring (with and without diagonal approximation) and BBVI. For SG Fisher Scoring, we set the learning rate to ϵ^* of Eq. 17, while for Langevin dynamics we chose the largest rate that yielded stable results ($\epsilon = \{10^{-3}, 10^{-6}, 10^{-5}\}$ for data sets 1, 2 and 3, respectively). We found that constant SGD can compete in approximating the posterior with the MCMC algorithms under consideration. This suggests that the most important factor is not the artificial noise involved in scalable MCMC, but rather the approximation of the preconditioning matrix.

4.2. Optimizing hyperparameters

To test the hypothesis of Section 3.3, namely that constant SGD as a variational algorithm allows for gradient-based hyperparameter learning, we experimented with a Bayesian multinomial logistic (a.k.a. softmax) regression model with normal priors. The negative log-joint being optimized is

$$\mathcal{L} \equiv -\log p(y, \theta | x, \lambda) = \tfrac{\lambda}{2} \textstyle\sum_{d,k} \theta_{dk}^2 - \tfrac{DK}{2} \log(\lambda) + \tfrac{DK}{2} \log 2\pi \\ + \textstyle\sum_n \log \sum_k \exp\{\sum_d x_{nd}\theta_{dk}\} - \sum_d x_{nd}\theta_{dy_n}, \qquad (26)$$

where $n \in \{1, \ldots, N\}$ indexes examples, $d \in \{1, \ldots, D\}$ indexes features and $k \in \{1, \ldots, K\}$ indexes classes. $x_n \in \mathbb{R}^D$ is the feature vector for the nth example and $y_n \in \{1, \ldots, K\}$ is the class for that example. Equation 26 has the degenerate maximizer $\lambda = \infty$, $\theta = 0$, which has infinite posterior density which we hope to avoid in our approach.

Data. In all experiments, we applied this model to the MNIST dataset (60000 training examples, 10000 test examples, 784 features) and the cover type dataset (500000 training examples, 81012 testing examples, 54 features).

Figure 3 shows the validation loss achieved by maximizing equation 26 over θ for various values of λ, as well as the values of λ selected by SGD and BBVI. The results suggest that this approach can be used as a simple, inexpensive alternative to cross-validation or other VEM methods for hyperparameter selection.

5. Related Work

Our paper relates to Bayesian inference and stochastic optimization.

Scalable MCMC. Recent work in Bayesian statistics focuses on making MCMC sampling algorithms scalable by using stochastic gradients. In particular, Welling and Teh (2011) developed stochastic gradient Langevin dynamics (SGLD). This algorithm samples from a Bayesian posterior by adding artificial noise to the stochastic gradient which, at long times, dominates the SGD noise. Also see Sato and Nakagawa (2014) for a detailed convergence analysis of the algorithm. Though elegant, one disadvantage of

[2] P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis, 'Wine Quality Data Set', UCI Machine Learning Repository.

[3] Prashant Singh Rana, 'Protein Tertiary Structure Data Set', UCI Machine Learning Repository.

[4] Rajen Bhatt, Abhinav Dhall, 'Skin Segmentation Dataset', UCI Machine Learning Repository.

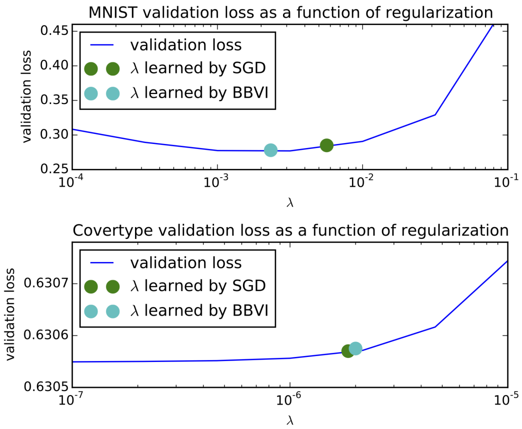

Figure 3: Validation loss as a function of L2 regularization parameter λ. Circles show the values of λ that were automatically selected by SGD and BBVI.

SGLD is that the learning rate must be decreased to achieve the correct sampling regime, and the algorithm can suffer from slow mixing times. Other research suggests improvements to this issue, using Hamiltonian Monte-Carlo (Chen et al., 2014) or thermostats (Ding et al., 2014). Ma et al. (2015) give a complete classification of possible stochastic gradient-based MCMC schemes.

Above, we analyzed properties of stochastic gradient Fisher scoring (Ahn et al., 2012). This algorithm speeds up mixing times in SGLD by preconditioning a gradient with the inverse sampling noise covariance. This allows constant learning rates, while maintaining long-run samples from the posterior. In contrast, we do not aim to sample exactly from the posterior. We describe how to tune the parameters of SGD such that its stationary distribution *approximates* the posterior.

Maclaurin et al. (2015) also interpret SGD as a nonparametric variational inference scheme, but with different goals and in a different formalism. The paper proposes a way to track entropy changes in the implicit variational objective, based on estimates of the Hessian. As such, the authors mainly consider sampling distributions that are not stationary, whereas we focus on constant learning rates and distributions that have (approximately) converged. Note that their notion of hyperparameters does not refer to model parameters but to parameters of SGD.

Stochastic Optimization. Stochastic gradient descent is an active field (Zhang, 2004; Bottou, 1998). Many papers discuss constant step-size SGD. Bach and Moulines (2013); Flammarion and Bach (2015) discuss convergence rate of averaged gradients with constant step size, while Défossez and Bach (2015) analyze sampling distributions using quasi-martingale techniques. Toulis et al. (2014) calculate the asymptotic variance of SGD for the case of decreasing learning rates, assuming that the data is distributed according to the model. None of these papers use variational arguments.

The fact that optimal preconditioning (using a decreasing Robbins-Monro schedule) is achieved by choosing the inverse noise covariance was first shown in (Sakrison, 1965), but here we derive the same result based on different arguments and suggest a scalar prefactor. Note the optimal scalar learning rate of $2/\mathrm{Tr}(BB^\top)$ can also be derived based on stability arguments, as was done in the context of least mean square filters (Widrow and Stearns, 1985).

Finally, Chen et al. (2015a) also draw analogies between SGD and scalable MCMC. They suggest annealing the posterior over iterations to use scalable MCMC as a tool for global optimization. We follow the opposite idea and suggest to use constant SGD as an approximate sampler by choosing appropriate learning rate and preconditioners.

Stochastic differential equations. The idea of analyzing stochastic gradient descent with stochastic differential equations is well established in the stochastic approximation literature (Kushner and Yin, 2003; Ljung et al., 2012). Recent work focuses on dynamical aspects of the algorithm. Li et al. (2015) discuss several one-dimensional cases and momentum. Chen et al. (2015b) analyze stochastic gradient MCMC and studied their convergence properties using stochastic differential equations.

Our work makes use of the same formalism but has a different focus. Instead of analyzing dynamical properties, we focus on stationary distributions. Further, our paper introduces the idea of minimizing KL divergence between multivariate sampling distributions and the posterior.

6. Conclusions

We analyzed stochastic gradient descent as an approximate Bayesian inference algorithm, deriving optimal constant learning rates and preconditioning matrices that minimize the Kullback-Leibler divergence between SGD's stationary distribution and the desired posterior distribution. This perspective, based on approximating SGD with a continuous-time Ornstein-Uhlenbeck process, uncovers connections between classical optimization-based learning algorithms, approximate MCMC algorithms used in Bayesian learning such as stochastic gradient Fisher scoring, and variational inference algorithms. This variational interpretation also leads to a simple (but effective) variational empirical Bayesian hyperparameter learning algorithm.

Acknowledgements. SM would like to thank David Carlson for discussions and Adobe Systems for supporting a research visit. This work is supported by NSF IIS-0745520, IIS-1247664, IIS-1009542, ONR N00014-11-

1-0651, DARPA FA8750-14-2-0009, N66001-15-C-4032, Adobe, Amazon, Facebook, NVIDIA, and the Seibel and John Templeton Foundations.

References

Ahn, S., Korattikara, A., and Welling, M. (2012). Bayesian posterior sampling via stochastic gradient fisher scoring. *arXiv preprint arXiv:1206.6380*.

Bach, F. and Moulines, E. (2013). Non-strongly-convex smooth stochastic approximation with convergence rate o (1/n). In *Advances in Neural Information Processing Systems*, pages 773–781.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer New York.

Bottou, L. (1998). Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):25.

Chen, C., Carlson, D., Gan, Z., Li, C., and Carin, L. (2015a). Bridging the gap between stochastic gradient mcmc and stochastic optimization. *arXiv preprint arXiv:1512.07962*.

Chen, C., Ding, N., and Carin, L. (2015b). On the convergence of stochastic gradient mcmc algorithms with high-order integrators. In *Advances in Neural Information Processing Systems*, pages 2269–2277.

Chen, T., Fox, E. B., and Guestrin, C. (2014). Stochastic gradient hamiltonian monte carlo. *arXiv preprint arXiv:1402.4102*.

Défossez, A. and Bach, F. (2015). Averaged least-mean-squares: Bias-variance trade-offs and optimal sampling distributions. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, pages 205–213.

Ding, N., Fang, Y., Babbush, R., Chen, C., Skeel, R. D., and Neven, H. (2014). Bayesian sampling using stochastic gradient thermostats. In *Advances in neural information processing systems*, pages 3203–3211.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

Flammarion, N. and Bach, F. (2015). From averaging to acceleration, there is only a step-size. *arXiv preprint arXiv:1504.01577*.

Gardiner, C. W. et al. (1985). *Handbook of stochastic methods*, volume 4. Springer Berlin.

Jordan, M., Ghahramani, Z., Jaakkola, T., and Saul, L. (1999a). Introduction to variational methods for graphical models. *Machine Learning*, 37:183–233.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999b). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.

Kramers, H. A. (1940). Brownian motion in a field of force and the diffusion model of chemical reactions. *Physica*, 7(4):284–304.

Kucukelbir, A., Ranganath, R., Gelman, A., and Blei, D. (2015). Automatic variational inference in stan. In *Advances in Neural Information Processing Systems*, pages 568–576.

Kushner, H. J. and Yin, G. (2003). *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media.

Li, Q., Tai, C., et al. (2015). Dynamics of stochastic gradient algorithms. *arXiv preprint arXiv:1511.06251*.

Ljung, L., Pflug, G. C., and Walk, H. (2012). *Stochastic approximation and optimization of random systems*, volume 17. Birkhäuser.

Longford, N. T. (1987). A fast scoring algorithm for maximum likelihood estimation in unbalanced mixed models with nested random effects. *Biometrika*, 74(4):817–827.

Ma, Y.-A., Chen, T., and Fox, E. B. (2015). A complete recipe for stochastic gradient mcmc. *arXiv preprint arXiv:1506.04696*.

Maclaurin, D., Duvenaud, D., and Adams, R. P. (2015). Early stopping is nonparametric variational inference. *arXiv preprint arXiv:1504.01344*.

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

Sakrison, D. J. (1965). Efficient recursive estimation; application to estimating the parameters of a covariance function. *International Journal of Engineering Science*, 3(4):461–483.

Sato, I. and Nakagawa, H. (2014). Approximation analysis of stochastic gradient langevin dynamics by using fokker-planck equation and ito process. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 982–990.

Tieleman, T. and Hinton, G. (2012). Lecture 6.5—RmsProp: Divide the Gradient by a Running Average of its Recent Magnitude. COURSERA: Neural Networks for Machine Learning.

Toulis, P., Airoldi, E., and Rennie, J. (2014). Statistical analysis of stochastic gradient methods for generalized linear models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 667–675.

Toulis, P., Tran, D., and Airoldi, E. M. Towards stability and optimality in stochastic gradient descent.

Uhlenbeck, G. E. and Ornstein, L. S. (1930). On the theory of the brownian motion. *Physical review*, 36(5):823.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688.

Widrow, B. and Stearns, S. D. (1985). Adaptive signal processing. *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1985, 491 p.*, 1.

Zhang, T. (2004). Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the twenty-first international conference on Machine learning*, page 116. ACM.