Stochastic Gradient Descent as Approximate Bayesian Inference

Stephan Mandt   STEPHAN.MANDT@GMAIL.COM
Data Science Institute
Department of Computer Science
Columbia University
New York, NY 10025, USA

Matthew D. Hoffman   MATHOFFM@ADOBE.COM
Adobe Research
Adobe Systems Incorporated
601 Townsend Street
San Francisco, CA 94103, USA

David M. Blei   DAVID.BLEI@COLUMBIA.EDU
Department of Statistics
Department of Computer Science
Columbia University
New York, NY 10025, USA

Abstract

Stochastic Gradient Descent with a constant learning rate (constant SGD) simulates a Markov chain with a stationary distribution. With this perspective, we derive several new results. (1) We show that constant SGD can be used as an approximate Bayesian posterior inference algorithm. Specifically, we show how to adjust the tuning parameters of constant SGD to best match the stationary distribution to a posterior, minimizing the Kullback-Leibler divergence between these two distributions. (2) We demonstrate that constant SGD gives rise to a new variational EM algorithm that optimizes hyperparameters in complex probabilistic models. (3) We also show how to tune SGD with momentum for approximate sampling. (4) We analyze stochastic-gradient MCMC algorithms. For Stochastic-Gradient Langevin Dynamics and Stochastic-Gradient Fisher Scoring, we quantify the approximation errors due to finite learning rates. Finally (5), we use the stochastic process perspective to give a short proof of why Polyak averaging is optimal. Based on this idea, we propose a scalable approximate MCMC algorithm, the Averaged Stochastic Gradient Sampler.

Keywords: approximate Bayesian inference, variational inference, stochastic optimization, stochastic gradient MCMC, stochastic differential equations

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) (Bouleau and Lepingle, 1994). Recent studies investigate the merits of adaptive step sizes (Duchi et al., 2011; Tieleman and Hinton, 2012), gradient or iterate averaging (Toulis et al., 2016; 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, since one can often obtain noisy-but-unbiased gradients very cheaply by randomly subsampling a large dataset.

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

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

These so-called stochastic gradient 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.

The similarities between SGD as an optimization algorithm and stochastic gradient MCMC algorithms raise the question of how exactly these two types of algorithms relate to each other. The main questions we try to address in this paper are:

- 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)?

To answer these questions, we draw on the theoretical analysis tools of continuous-time stochastic differential equations (Bachelier, 1900; Gardiner et al., 1985) and variational inference (Jordan et al., 1999).

As a simple example, 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. (In contrast, traditional SGD converges to the optimum by decreasing the learning rate.) Our analysis below rests on the idea that 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. In contrast, stochastic gradient MCMC algorithms take precautions to sample from an asymptotically exact posterior, but at the expense of slower mixing. Our inexact approach enjoys minimal implementation effort and typically faster mixing. It is a hybrid of Monte Carlo and variational algorithms, and complements the toolbox of approximate Bayesian inference.

Here is how it works. We apply constant SGD as though we were trying to minimize the negative log-joint probability $-\log p(\theta, \mathbf{x})$ 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. These tuning parameters all affect the stationary distribution of constant SGD; depending on how they are set, this stationary distribution of θ will be closer to or farther from the posterior distribution $p(\theta|\mathbf{x})$. If we set these parameters appropriately, we can perform approximate Bayesian inference by simply running constant SGD.

In more detail, we make the following contributions:

1. 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 make it easy to minimize the KL divergence between the stationary distribution and the posterior analytically. We can thus relate the optimal step size or preconditioning matrix to the Hessian and noise covariances near the optimum. The optimal preconditioners relate to 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.

2. 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 demonstrate this by fitting a posterior to a Bayesian multinomial regression model.

3. We use our formalism to derive the stationary distribution for SGD with momentum (Polyak, 1964). Our results show that adding momentum only changes the scale of the covariance of the stationary distribution, not its shape. This scaling factor is a simple function of the damping coefficient. Thus we can also use SGD with momentum for approximate Bayesian inference.

4. Then, we analyze scalable MCMC algorithms. Specifically, we use the stochastic-process perspective to compute the stationary distribution of Stochastic-Gradient Langevin Dynamics (SGLD) by Welling and Teh (2011) when using constant learning rates, and analyze 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 confirm that the preconditioning matrix suggested in SGFS is indeed optimal. We also derive a criterion for the free noise parameter in SGFS that can 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).

5. 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. Based on the stochastic-process methodology, we give a shorter derivation of a known result, namely that the convergence speed of iterate averaging cannot be improved by preconditioning the stochastic gradient with any matrix. Furthermore, we show that (under certain assumptions), Polyak iterate averaging can yield an *optimal* stochastic-gradient MCMC algorithm, and that this optimal sampler can generate exactly one effectively independent sample per pass through the dataset. This result is both positive and negative; it suggests that iterate averaging can be used as a powerful Bayesian sampler, but it also argues that no SG-MCMC algorithm can generate more than one useful sample per pass through the data, and so the cost of these algorithms must scale linearly with dataset size.

Our paper is organized as follows. In Section 3 we review the continuous-time limit of SGD, showing that it can be interpreted as an OU process. In Section 4 we present consequences of this perspective: the interpretation of SGD as variational Bayes and results around preconditioning and momentum. Section 5 discusses SG Langevin Dynamics and SG Fisher Scoring. In Section 6 we discuss Polyak averaging for optimization and sampling. In the empirical study (Section 7), we show that our theoretical assumptions are satisfied for different models, that we can use SGD to perform gradient-based hyperparameter optimization, and that iterate averaging gives rise to a Bayesian sampler with fast mixing.

2. 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, as the step size decays, comes to dominate the SGD noise. Also see Sato and Nakagawa (2014) for a detailed convergence analysis of the algorithm. Though elegant, one disadvantage of SGLD is that the step size must be decreased to arrive at the correct sampling regime, and as step sizes get small so does mixing speed. Other research suggests improvements to this issue, using Hamiltonian Monte Carlo (Chen et al., 2014) or thermostats (Ding et al., 2014). Shang et al. (2015) build on thermostats and use a similar continuous-time formalism as used in this paper.

Ma et al. (2015) give a complete classification of possible stochastic gradient-based MCMC schemes.

Below, we will analyze properties of stochastic gradient Fisher scoring (SGFS; Ahn et al., 2012), an extention to SGLD. This algorithm speeds up mixing times in SGLD by preconditioning gradients with the inverse gradient noise covariance. Ahn et al. (2012) show that (under some assumptions) SGFS can eliminate the bias associated with running SGLD with non-vanishing learning rates. Our approach to analysis can extend and sharpen the results of Ahn et al. (2012). For example, they propose using a diagonal approximation to the gradient noise covariance as a heuristic; in our framework, we can analyze and rigorously justify this choice as a variational Bayesian approximation.

Maclaurin et al. (2016) also interpret SGD as a non-parametric 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 consider the Bayesian setting. Dieuleveut et al. (2017) also analyzed SGD with constant step size and its relation to

Markov chains. Their analysis resulted in a novel extrapolation scheme to improve the convergence behavior of iterate averaging.

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 $2/\mathrm{Tr}(BB^\top)$, where $BB^\top$ is the SGD noise covariance (as discussed in Section 4 or this paper), can also be derived based on stability arguments. This was done in the context of least mean square filters (Widrow and Stearns, 1985).

Finally, Chen et al. (2016) also draw analogies between SGD and scalable MCMC. They suggest annealing the posterior over time 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. Li et al. (2017) give a mathematically rigorous justification of the continuous-time limit. Chen et al. (2015) analyze stochastic gradient MCMC and study 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.

Variational Inference. Variational Inference (VI) denotes a set of methods which aim at approximating a Bayesian posterior by a simpler, typically factorized distribution. This is done by minimizing Kullback-Leibler divergence or related divergences between these distributions (Jordan et al., 1999; Opper and Saad, 2001). For the class of models where the conditional distributions are in the exponential family, the variational objective can be optimized by closed-form updates (Ghahramani and Beal, 2000), but this is a restricted class of models with conjugate priors. A scalable version of VI, termed Stochastic Variational Inference (SVI), relies on stochastic gradient descent for data subsampling (Hoffman et al., 2013). For non-conjugate models, Black-Box variational inference (Ranganath et al., 2014) has enabled SVI for a large class of models, but this approach may suffer from high-variance gradients. A modified form of black-box variational inference relies on re-parameterization gradients (Salimans and Knowles, 2013; Kingma and Welling, 2014; Rezende et al., 2014; Kucukelbir et al., 2015; Ruiz et al., 2016). This version is limited to continuous latent variables but typically has much lower-variance gradients.

In this paper, we compare against the Gaussian reparameterization gradient version of black-box variational inference as used in Kingma and Welling (2014); Rezende et al. (2014); Kucukelbir et al. (2015) which we refer to as BBVI. We find that our approach performs similarly in practice, but it is different in that it does not optimize the parameters of a simple variational distribution. Rather, it controls the shape of the approximate posterior via the parameters of the optimization algorithm, such as the learning rate or preconditioning matrix.

3. 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).

3.1 Problem Setup

Consider loss functions of the following form:

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

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

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

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,

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

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

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

Above and for what follows, we assume a constant (non-decreasing) learning rate ϵ.

Eqs. 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.

3.2 SGD as an Ornstein-Uhlenbeck Process

We now show how to approximate the discrete-time Eq. 5 with a 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 empirically in Section 7.

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*

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

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

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

Assumption 2 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) that the noise covariance does not vary significantly in that region.

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

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

This can be interpreted as a finite-difference equation that approximates the following continuous-time stochastic differential equation:

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

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,*

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

(Without loss of generality, we assume that a minimum of the loss is at $\theta = 0$.) We also assume that A is positive definite.

The symmetric matrix A is thus the Hessian at the optimum. Assumption 4 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).

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):

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

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

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

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

Discussion of Assumptions 1–4. Our analysis suggests that constant SGD and Langevin-type diffusion algorithms (Welling and Teh, 2011) are very similar. Both types of algorithms can be characterized by three regimes. First, there is a search phase where the algorithm approaches the optimum. In this early phase, assumptions 1–4 are often violated and it is hard to say anything general about the behavior of the algorithm. Second, there is a phase where SGD has converged to the vicinity of a local optimum. Here, the objective already looks quadratic, but the gradient noise is small relative to the average gradient $g(\theta)$. Thus SGD takes a relatively directed path towards the optimum. This is the regime where our assumptions should be approximately valid, and where our formalism reveals its use. Finally, in the third phase the iterates are near the local optimum. Here, the average gradient $g(\theta)$ is small and the sampling noise becomes more important. In this final phase, constant SGD begins to sample from its stationary distribution.

Finally, we note that if the gradient noise covariance C does not have full rank, then neither will the stationary covariance Σ. This scenario complicates the analysis in Section 4, so below we will assume that C has full rank. This could be enforced easily by adding very low-magnitude isotropic artificial Gaussian noise to the stochastic gradients.

4. 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 $p(\theta \,|\, \mathbf{x})$ in Eq. 1.

We set the loss to be 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, and neural networks. In contrast, our goal here is to tune the parameters of SGD so that its stationary distribution approximates the posterior.

Fig. 1 shows an example. Here we illustrate two Bayesian posteriors—from a linear regression problem (top) and a logistic regression problem (bottom)—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 left plots optimize both a preconditioning matrix and the step size; the middle plots optimize only the step size (both are outlined in Section 4.1). We can see that the stationary distribution of constant SGD can be made close to the exact posterior.

Fig. 2 also compares the empirical covariance of the iterates with the predicted covariance in terms of Eq. 13. The close match supports the assumptions of Section 3.
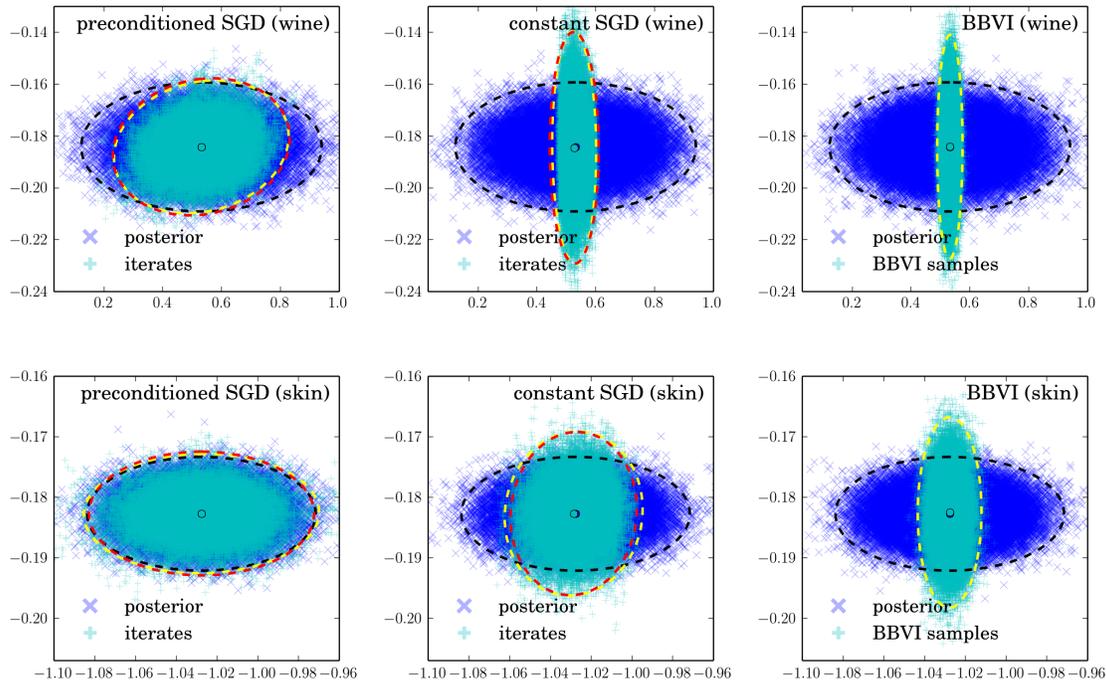
Figure 1: Posterior distribution $f(\theta) \propto \exp\{-N\mathcal{L}(\theta)\}$ (blue) and stationary sampling distributions $q(\theta)$ of the iterates of SGD (cyan) or black box variational inference (BBVI) based on reparameterization gradients. Rows: linear regression (top) and logistic regression (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.
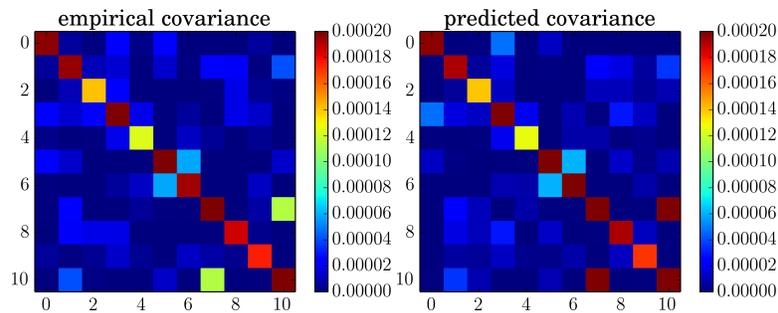

Figure 2: Empirical and predicted covariances of the iterates of stochastic gradient descent, where the prediction is based on Eq. 13. We used linear regression on the wine quality data set as detailed in Section 7.1.

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 (Duchi et al., 2011; Tieleman and Hinton, 2012). Second, we propose an algorithm for hyperparameter optimization based on constant SGD. Third, we use it to analyze the stationary distribution of stochastic gradient descent with momentum (Polyak, 1964).

4.1 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). The 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 reflects a different tradeoff between efficiency and accuracy. Finally, we show how to use these algorithms to learn hyperparameters.

Assumption 4 from Section 3 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. Furthermore, in this section 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). This involves the the learning rate ϵ or more generally the preconditioning matrix H and the minibatch size S:

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

First, consider a scalar learning rate ϵ (or a trivial preconditioner $H = \epsilon\mathbf{I}$). 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

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

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 that minimizes KL divergence from the stationary distribution of constant SGD to the posterior is*

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

Proof 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 $\mathrm{Tr}(A\Sigma) = \frac{1}{2}(\mathrm{Tr}(A\Sigma) + \mathrm{Tr}(\Sigma A)) = \frac{\epsilon}{2S}\mathrm{Tr}(BB^\top)$. The KL divergence is therefore, up to constant terms,

$$KL(q||f) \overset{\mathrm{c}}{=} \frac{\epsilon N}{2S}\mathrm{Tr}(BB^\top) - D\log(\epsilon/S). \tag{16}$$

Minimizing KL divergence over ϵ/S results in Eq. 15 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, and proportional to the ratio between the minibatch size and dataset size.

Since both the posterior f and the variational distribution q are Gaussian, one might wonder if also the reverse KL divergence is viable. While the KL divergence can be computed up to a constant, we cannot remove its dependence on the unknown stationary distribution Σ using Eq. 13, unless A and Σ commute. This setup is discussed in Appendix C.

Instead of using a scalar learning rate, we now consider a positive-definite preconditioning matrix H. This gives us more tuning parameters to better approximate the posterior.

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

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

Proof To prove this, we need the Ornstein-Uhlenbeck process which corresponds to preconditioned SGD. Replacing the constant learning rate in Eq. 11 with a positive-definite preconditioning matrix H results in

$$d\theta(t) = -HA\,\theta(t)dt + \frac{1}{\sqrt{S}}HB\,dW(t).$$

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

$$A\Sigma + H^{-1}\Sigma AH = \frac{1}{S}BB^\top H. \tag{18}$$

Using the cyclic property of the trace, this implies that

$$\mathrm{Tr}(A\Sigma) = \frac{1}{2}(\mathrm{Tr}(A\Sigma) + \mathrm{Tr}(H^{-1}A\Sigma H)) = \frac{1}{2S}\mathrm{Tr}(BB^\top H). \tag{19}$$

Consider now the log determinant term, $\log|\Sigma|$, which still has an implicit dependence on H. We first define $Q = \Sigma H^{-1}$, hence $Q^\top = H^{-1}\Sigma$ since Σ, H and H^{-1} are symmetric. Eq. 18 can be written as $A\Sigma H^{-1} + H^{-1}\Sigma A = \frac{1}{S}BB^\top$, which is equivalent to $QA + AQ^\top = \frac{1}{S}BB^\top$.

Thus, we see that Q is independent of H. The log determinant term is up to a constant $\log|\Sigma| = \log|H| + \log|Q|$. Combining Eq. 19 with this term, the KL divergence is up to a constant

$$KL(q||f) \stackrel{c}{=} \frac{N}{2S}\text{Tr}(BB^\top H) + \log|H| + \log|Q|. \tag{20}$$

Taking derivatives with respect to the entries of H results in Eq. 17. ∎

In high-dimensional applications, working with large dense matrices is impractical. In those settings we can constrain the preconditioner to be diagonal. The following corollaries are based on the proof of Theorem 2:

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

Proof This follows from Eq. 20, where we restrict the preconditioning matrix to be diagonal. ∎

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

Proof Consider Eq. 18. Inserting $H = \frac{2S}{N}(BB^\top)^{-1}$ results in $A\Sigma + H^{-1}\Sigma AH = \frac{2}{N}\mathbf{I}$ which is solved by $\Sigma = A^{-1}/N$ which is the posterior covariance. ∎

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 Appendix F we derive an optimal global learning rate for AdaGrad-style diagonal preconditioners. In Section 7, 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.

Remark on estimating the noise covariance. In order to use our theoretical insights in practice, we need to estimate the stochastic gradient noise covariance $C \equiv BB^\top$. We do this in an online manner. As before, let g_t be the full gradient, $\hat{g}_{S,t}$ be the stochastic gradient of the full minibatch, and $\hat{g}_{1,t}$ be the stochastic gradient of the first sample in the minibatch at time t (which has a much larger variance if $S \gg 1$). For large S, we can approximate $g_t \approx \hat{g}_{S,t}$, and thus obtain an estimator of the noise covariance by $(\hat{g}_{1,t} - \hat{g}_{S,t})(\hat{g}_{1,t} - \hat{g}_{S,t})^\top$. Following Ahn et al. (2012), we can now build an online estimate C_t that approaches C by the following recursion,

$$C_t = (1 - \kappa_t)C_{t-1} + \kappa_t(\hat{g}_{1,t} - \hat{g}_{S,t})(\hat{g}_{1,t} - \hat{g}_{S,t})^\top. \tag{21}$$

Above, κ_t is a decreasing learning rate. Ahn et al. (2012) have proven that such an online average converges to the noise covariance in the optimum at long times (provided that $\kappa_t \sim 1/t$ and that N is sufficiently large). We found that this online estimator works well in practice, even though our theoretical assumptions would require preconditioning SGD with the true noise covariance in finite

time. Regarding the computational overhead of this procedure, note that similar online estimates of the gradient noise are carried out in adaptive SGD schemes such as AdaGrad (Duchi et al., 2011) or RMSProp (Tieleman and Hinton, 2012). When using a diagonal approximation to the noise covariance, the costs are proportional in the number of dimensions and mini-batch size; this efficiency means that the online estimate does not spoil the efficiency of SGD. Full preconditioning scales quadratically in the dimension and is therefore impractical in many real-word setups.

4.2 Constant SGD as Variational EM

Consider a supervised probabilistic model with joint distribution $p(y, \theta | x, \lambda) = p(y|x, \theta)p(\theta|\lambda)$ that factorizes into a likelihood and prior, respectively. Our goal is to find optimal hyperparameters λ. Jointly point-estimating θ and λ by following gradients of the log joint leads to overfitting or degenerate solutions. This can be prevented in a Bayesian approach, where we treat the parameters θ as latent variables. 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 $\mathbb{E}_q[\log p(\theta, y|x, \lambda)]$.

Constant SGD gives rise to a simple VEM algorithm that applies to a large class of differentiable models. Define $\mathcal{L}(\theta, \lambda) = -\log p(y, \theta \,|\, x, \lambda)$. 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:

$$\theta_{t+1} = \theta_t - \epsilon^* \nabla_\theta \mathcal{L}(\theta_t, \lambda_t); \quad \lambda_{t+1} = \lambda_t - \rho_t \nabla_\lambda \mathcal{L}(\theta_t, \lambda_t). \tag{22}$$

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 a type of VEM algorithm.

We stress that the optimal constant learning rate ϵ^* is not unknown, but can be estimated based on an online estimate of the noise covariance $C \equiv BB^\top$, as given in Eq. 21. In Section 7 we show that gradient-based hyperparameter learning is a cheap alternative to cross-validation.

4.3 Stochastic Gradient with Momentum

The continuous-time formalism also allows us to explore extensions of classical SGD. One of the most popular methods is stochastic gradient with momentum (Polyak, 1964; Sutskever et al., 2013). Here, we give a version of this algorithm that allows us to sample from an approximate posterior.

SGD with momentum doubles the dimension of parameter space in introducing an additional *momentum* variable v that has the same dimension as θ. Position and momentum are coupled in such a way that the algorithm keeps memory of some of its past gradients. The updates of SGD with momentum are

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

13

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

As before we assume a quadratic objective $\mathcal{L} = \frac{1}{2}\theta^\top A\theta$. Going through the same steps A1-A4 of Section 3 that allowed us to derive the Ornstein-Uhlenbeck process for SGD, we find

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

We solve this set of stochastic equations asymptotically for the long-time limit. We reformulate the stochastic equation in terms of coupled equations for the moments. (This strategy was also used by Li et al. (2015) in a more restricted setting). The first moments of the set of coupled stochastic differential equations give

$$
d\mathbb{E}[\mathrm{v}] = -\mu\mathbb{E}[\mathrm{v}]dt - \epsilon A\mathbb{E}[\theta]dt, \quad d\mathbb{E}[\theta] = \mathbb{E}[\mathrm{v}]dt.
$$

Note we used that expectations commute with the differential operators. These deterministic equations have the simple solution $\mathbb{E}[\theta] = 0$ and $\mathbb{E}[\mathrm{v}] = 0$, which means that the momentum has expectation zero and the expectation of the position variable converges to the optimum (at 0).

In order to compute the stationary distribution, we derive and solve similar equations for the second moments. These calculations are carried out in Appendix D, where we derive the following conditions:

$$
\begin{aligned}
\mathbb{E}[\mathrm{v}\mathrm{v}^\top] &= \tfrac{\epsilon}{2}\mathbb{E}[\theta\theta^\top]A + \tfrac{\epsilon}{2}A\mathbb{E}[\theta\theta^\top], \\
\mu\mathbb{E}[\mathrm{v}\mathrm{v}^\top] &= \tfrac{\epsilon^2}{2S}BB^\top.
\end{aligned}
\tag{24}
$$

Eq. 25 relate to equilibrium thermodynamics, where $\mathbb{E}[\mathrm{v}\mathrm{v}^\top]$ is a matrix of expected kinetic energies, while $\frac{1}{2}(\mathbb{E}[\theta\theta^\top]A + A\mathbb{E}[\theta\theta^\top])$ has the interpretation of a matrix of expected potential energies. The first equation says that energy conservation holds in expectation, which is the case for an equilibrium system which exchanges energy with its environment. The second equation relates the covariance of the velocities with $BB^\top$, which plays the role of a matrix-valued temperature. This is known as the fluctuation-dissipation theorem (Nyquist, 1928). Combining both equations and using $\Sigma = \mathbb{E}[\theta\theta^\top]$ yields

$$
\Sigma A + A\Sigma = \tfrac{\epsilon}{\mu S}BB^\top.
\tag{25}
$$

This is exactly Eq. 13 of SGD without momentum, however, with the difference that the noise covariance is re-scaled by a factor $\frac{\epsilon}{\mu S}$ instead of $\frac{\epsilon}{S}$.

We have shown that ϵ, S, and μ play similar roles: only the combination $\frac{\epsilon}{\mu S}$ affects the KL divergence to the posterior. Thus, no single optimal constant learning rate exists—many combinations of ϵ, μ, and S can yield the same stationary distribution. But different choices of these parameters will affect the dynamics of the Markov chain. For example, Sutskever et al. (2013) observe that, for a given effective learning rate $\frac{\epsilon}{\mu S}$, using a smaller μ sometimes makes the *discretized* dynamics of SGD more stable. Also, using very small values of μ while holding $\frac{\epsilon}{\mu}$ fixed will eventually increase the autocorrelation time of the Markov chain (but this effect is often negligible in practice).

5. Analyzing Stochastic Gradient MCMC Algorithms

We have analyzed well-known stochastic optimization algorithms such as SGD, preconditioned SGD and SGD with momentum. We now investigate Bayesian sampling algorithms. A large class of modern MCMC methods rely on stochastic gradients (Welling and Teh, 2011; Ahn et al., 2012; Chen et al., 2014; Ding et al., 2014; Ma et al., 2015). The central idea is to add artificial noise to the stochastic gradient to asymptotically sample from the true posterior.

In practice, however, the algorithms are used in combination with several heuristic approximations, such as non-vanishing learning rates or diagonal approximations to the Hessian. In this section, we use the variational perspective to quantify these biases and help understanding these algorithms better under more realistic assumptions.

5.1 SGLD with Constant Rates

To begin with, we we analyze the well-known Stochastic Gradient Langevin Dynamics by Welling and Teh (2011). This algorithm has been carefully analyzed in the long time limit where the stochastic gradient noise vanishes as the learning rate goes to zero, and where mixing becomes infinitely slow (Sato and Nakagawa, 2014). Here we analyze an approximate version of the algorithm, SGLD with a constant learning rate. We are interested in the stationary distribution of this approximate inference algorithm.

The discrete-time process that describes Stochastic Gradient Langevin dynamics is

$$\theta_{t+1} \quad = \quad \theta_t - \tfrac{\epsilon}{2} N \hat{\nabla}_\theta \mathcal{L}(\theta_t) + \sqrt{\epsilon}\, V(t),$$

where $V(t) \sim \mathcal{N}(0, \mathbf{I})$ is a vector of independent Gaussian noises. Following assumptions 1–4 of Section 3, V becomes the Wiener noise dV and the corresponding continuous-time process is

$$d\theta \quad = \quad -\tfrac{1}{2}\epsilon N A\theta dt + \sqrt{\epsilon}dV + \epsilon\tfrac{1}{\sqrt{S}} N B\, dW.$$

Above, dV and dW are vectors of independent Wiener noises, i.e. $\mathbb{E}[dW\, dV^\top] = \mathbb{E}[dV\, dW^\top] = 0$. The analog of Eq. 13 is then

$$\tfrac{1}{2}N(A\Sigma + \Sigma A) \quad = \quad \mathbf{I} + \tfrac{\epsilon N}{2S} BB^\top.$$

In the limit of $\epsilon \to 0$, we find that $\Sigma^{-1} = NA$, meaning the stationary distribution becomes identical to the posterior. However, for non-zero ϵ, there are discrepancies. These correction-terms are positive. This shows that the posterior covariance is generally overestimated by Langevin dynamics, which can be attributed to non-vanishing learning rates at long times.

5.2 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 can be cast into the following form:

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

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 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*

$$H^* = \tfrac{2}{N}(\epsilon BB^\top + EE^\top)^{-1}. \tag{27}$$

Proof To prove the claim, we go through the steps of Section 3 to derive the corresponding Ornstein-Uhlenbeck process, $d\theta(t) = -\epsilon H A\theta(t)dt + H\left[\epsilon B + \sqrt{\epsilon}E\right]dW(t)$. For simplicity, we have set the minibatch size S to 1. In Appendix E, we derive the following KL divergence between the posterior and the sampling distribution:

$$KL(q||p) = -\tfrac{N}{4}\mathrm{Tr}(H(\epsilon BB^\top + EE^\top)) + \tfrac{1}{2}\log|T| + \tfrac{1}{2}\log|H| + \tfrac{1}{2}\log|NA| + \tfrac{D}{2}.$$

(T is constant with respect to H, ϵ, and E.) We can now minimize this KL divergence over the parameters H and E. When E is given, minimizing over H gives Eq. 27. ∎

Eq. 27 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 5:

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

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

Note that we have not made any assumptions about the noise covariance E. We can adjust it in favor of a more stable algorithm. For example, in the interests of stability we might want to set 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. 27 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.

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. An extreme case is when the gradient covariance $BB^\top$ is not full rank; in this situation injecting full-rank Gaussian noise could prevent degenerate behavior.

6. A Bayesian View on Iterate Averaging

We now apply our continuous-time analysis to the technique of iterate averaging (Polyak and Juditsky, 1992). Iterate averaging was derived as an optimization algorithm, and we analyze this algorithm in Section 6.1 from this perspective. In Section 6.2 we show that iterate averaging can also be used as a Bayesian sampling algorithm.

6.1 Iterate Averaging for Optimization

Iterate averaging further refines SGD's estimate of the optimal parameters by averaging the iterates produced by a series of SGD steps. Polyak and Juditsky (1992) proved the remarkable result that averaged SGD achieves the best possible convergence rate among stochastic gradient algorithms[1], including those that use second-order information such as Hessians. This implies that the convergence speed of iterate averaging cannot be improved when premultiplying the stochastic gradient with any preconditioning matrix. We use stochastic calculus to show that the stationary distribution of iterate averaging is the same for any constant preconditioner. Based on slightly stronger-than-usual assumptions, we give a short proof on why this result holds.

To eliminate asymptotic biases, iterate averaging usually requires a slowly decreasing learning rate. We consider a simplified version with a constant rate, also analyzed in (Zhang, 2004; Nemirovski et al., 2009).

Algorithm. Before we begin iterate averaging, we assume that we have run ordinary constant SGD for long enough that it has reached its stationary distribution and forgotten its intialization. In this scenario, we use iterate averaging to "polish" the results obtained by SGD.

Iterate averaging estimates the location of the minimum of $\mathcal{L}$ using a sequence of stochastic gradients $\hat{g}_S$, 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 $\hat{\mu}$ and its approximation $\hat{\mu}'$ are random variables whose expected value is the minimum of the objective. (Again, this assumes $\theta(0)$ is drawn from the SGD process's stationary distribution.)

The accuracy of this estimator $\hat{\mu}'$ after a fixed number of iterations T is characterized by its covariance matrix D. Using stochastic calculus, we can compute D from the autocorrelation matrix of the stationary distribution of the OU process, shown in Appendix G. We prove that

$$
D \equiv \mathbb{E}\left[\hat{\mu}'\hat{\mu}'^\top\right] \approx \tfrac{1}{\epsilon T}\left(\Sigma(A^{-1})^\top + A^{-1}\Sigma\right).
\tag{30}
$$

---

1. To be more precise, averaged SGD is optimal among methods that only have access to a stochastic gradient oracle—if more is known about the source of the noise then sometimes better rates are possible (e.g., Johnson and Zhang, 2013; Defazio et al., 2014).

This approximation ignores terms of order $1/(T\epsilon)^2$. This term is small since we assume that the number of iterations T is much larger than the inverse learning rate $1/\epsilon$. The covariance shrinks linearly over iterations as we expect from Polyak and Juditsky (1992). We use Eq. 13 to derive

$$D \approx \tfrac{1}{TS} A^{-1} B B^\top (A^{-1})^\top.$$

Note that this covariance depends only on the number of iterations T times the minibatch size S, not on the step size ϵ. Since TS is the total number of examples that are processed, this means that this iterate averaging scheme's efficiency does not depend on either the minibatch size or the step size, proven first in (Polyak and Juditsky, 1992).

We can make a slightly stronger statement. If we precondition the stochastic gradients with a positive-definite matrix H (for example, the inverse of the Hessian evaluated at the minimum), it turns out that the covariance of the estimator remains unchanged. The resulting Ornstein-Uhlenbeck process is $d\theta = -HA\theta(t)dt + \tfrac{1}{\sqrt{S}}HB\,dW(t)$. The resulting stationary covariance D' of preconditioned iterate averaging is the same as above:

$$
\begin{aligned}
D' &\approx \tfrac{1}{T}(HA)^{-1}(\tfrac{1}{\sqrt{S}}HB)(\tfrac{1}{\sqrt{S}}HB)^\top((HA)^\top)^{-1} \\
&= \tfrac{1}{TS}A^{-1}H^{-1}HBB^\top HH^{-1}A^{-1} \\
&= \tfrac{1}{TS}A^{-1}BB^\top A^{-1}.
\end{aligned}
\tag{31}
$$

Both the stationary distribution and the autocorrelation matrix change as a result of the preconditioning, and these changes exactly cancel each other out.

This optimality of iterate averaging was first derived by Polyak and Juditsky (1992), using quasi-martingales. Our derivation is based on stronger assumptions, but is shorter.

6.2 Finite-Window Iterate Averaging for Posterior Sampling

Above, we used the continuous-time formalism to quickly rederive known results about the optimality of iterate averaging as an optimization algorithm. Next, as in Section 4, we will analyze iterate averaging as an algorithm for approximate posterior inference.

We will show that (under some optimistic assumptions), iterate averaging requires exactly N gradient calls to generate one sample drawn from the exact posterior distribution, where N is the number of observations. That is, there exist conditions under which iterate averaging generates one true posterior sample per pass over the data. This result is both exciting and discouraging; it implies that, since our assumptions are all optimistic and iterate averaging is known to saturate the Cramér-Rao bound, *no* black-box stochastic-gradient MCMC algorithm can generate samples in time sublinear in the number of data points.

In addition to Assumptions 1–4 of Section 3, we need an additional assumption for our theoretical considerations to hold:

Assumption 5 *Assume that the sample size N is large enough that the Bernstein-von Mises theorem (Le Cam, 1986) applies (hence the posterior is Gaussian). Also assume that the observed dataset was drawn from the model $p(y\,|\theta)$ with parameter $\theta = 0$. Then $A = BB^\top$, that is, the Fisher information matrix equals the Hessian.*

This simplifies equation 13:

$$A\Sigma + \Sigma A = \tfrac{\epsilon}{S}BB^\top \overset{A5}{\Longrightarrow} \Sigma = \frac{\epsilon}{2S}\mathbf{I}. \tag{32}$$

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 $\{\mu_1, \ldots, \mu_M\}$.

That is, the sampling distribution of SGD is isotropic.

Stationary distribution. We will consider the sampling distribution of the *average* of T successive samples from the stationary SGD process with step size ϵ. Going to the continuous-time OU formalism, we show in Appendix G that the stationary covariance of the iterate averaging estimator defined in Eq. 30 is

$$D = \frac{1}{ST}A^{-1} + \frac{1}{\epsilon ST^2}U\Lambda^{-2}(e^{-\epsilon T\Lambda} - \mathbf{I})U^\top, \tag{33}$$

where U is orthonormal, Λ is diagonal, and $U\Lambda U^\top = A$ is the eigendecomposition of the Hessian A. We have previously assumed that the posterior has covariance $\frac{1}{N}A^{-1}$. Thus, to leading order in the ratio $1/(\epsilon T\Lambda)$, the stationary distribution of fixed-window iterate averaging is a scaled version of the posterior.

If we choose $T = N/S$, so that we average the iterates of a single pass through the dataset, then the iterate-averaging sampling distribution will have approximately the same covariance as the posterior,

$$\begin{aligned}
D^\star &= \frac{1}{N}A^{-1} + \frac{S}{\epsilon}\frac{1}{N^2}U\Lambda^{-2}(e^{-\frac{\epsilon}{S}N\Lambda} - \mathbf{I})U^\top \\
&= \frac{1}{N}U\Lambda^{-1}\left(\mathbf{I} + \frac{S}{\epsilon}\frac{1}{N}\Lambda^{-1}(e^{-\frac{\epsilon}{S}N\Lambda} - \mathbf{I})\right)U^\top.
\end{aligned} \tag{34}$$

$D^\star$ and A^{-1} have identical eigenvectors, and their eigenvalues differ by a factor that goes to zero as $\frac{\epsilon}{S}$ becomes large. Conversely, as $\frac{\epsilon}{S}$ approaches zero, all of these eigenvalues approach $\frac{\epsilon}{2S}$ as in Eq. 32. (This can be shown by taking a second-order Maclaurin approximation of $e^{-\frac{\epsilon N}{S}\Lambda}$.)

Our analysis gives rise to the Iterate Averaging Stochastic Gradient sampler (IASG), described in Algorithm 1. We now investigate its approximation error and efficiency.

Approximation error, step size, and minibatch size. We now focus on the correction terms that lead to deviations between the iterate averaging estimator's covariance D^* and the posterior covariance $\frac{1}{N}A^{-1}$.

The analysis above tells us that we can ignore these correction terms if we choose a large enough $\frac{\epsilon}{S}$. But the analysis in previous chapters assumes that $\frac{\epsilon}{S}$ is small enough that assumptions 1–4 hold. These considerations are in tension.

When is $\frac{\epsilon}{S}$ "large enough"? Eq. 34 shows that the relative error is largest in the direction of the smallest eigenvalue $\lambda_{\min} \equiv \min_k \Lambda_{kk}$ of A (corresponding to the least-constrained direction in the

posterior). We will focus our analysis on the relative error in this direction, which is given by

$$\text{err}_{\max} \equiv \frac{S}{\epsilon} \frac{1}{N\lambda_{\min}} (e^{-\frac{S}{\epsilon}N\lambda_{\min}} - 1).$$

We are given N and $\lambda_{\min}$, but we can control $\frac{\epsilon}{S}$. To make $\text{err}_{\max}$ small, we must therefore choose $\frac{\epsilon}{S} > \frac{c}{N\lambda_{\min}}$ for some constant c. So larger datasets and lower-variance posteriors let us use smaller stepsizes.

It is hard to say in general how small $\frac{\epsilon}{S}$ needs to be to satisfy assumptions 1–4. But if assumption 4 is satisfied (i.e., the cost is approximately quadratic), then assumption 3 (no discretization error) cannot hold if $\epsilon > \frac{2}{\lambda_{\max}}$. This is the step size at which the discretized noise-free gradient descent process becomes unstable for quadratic costs. (We define $\lambda_{\max} \equiv \max_k \Lambda_{kk}$ analogous to $\lambda_{\min}$.)

Combining this observation with the analysis above, we see that this iterate averaging scheme will not work unless

$$\frac{2}{S\lambda_{\max}} > \frac{\epsilon}{S} > \frac{c}{N\lambda_{\min}} \Rightarrow \frac{2}{c} > \frac{S}{N} \frac{\lambda_{\max}}{\lambda_{\min}}.$$

That is, we need the dataset size N to be large enough relative to the condition number $\frac{\lambda_{\max}}{\lambda_{\min}}$ of the Hessian A if this simple iterate-averaging scheme is to generate good posterior samples. If the condition number is large relative to N, then it may be necessary to replace the scalar step size ϵ with a preconditioning matrix $H \approx A^{-1}$ to reduce the effective condition number of the Hessian A.

Efficiency. Next, we theoreticaly analyze the efficiency with which iterate averaging can draw samples from the posterior, and compare this method to other approaches. We assume that the cost of analyzing a minibatch is proportional to S. We have shown above that we need to average over $T = N/S$ samples of SGD to create a sample of iterate averaged SGD. Since this averaging induces a strong autocorrelation, we can only use every Tth sample of the chain of averaged iterates. Furthermore, every sample of SGD incurs a cost at least proportional to D where D is the dimensionality of θ. This means that we cannot generate an independent sample from the posterior in less than $O(S * T * D) = O(ND)$ time; we must analyze at least N observations per posterior sample.

We compare this result with the more classical strategy of estimating the posterior mode via Newton's method (which has an $O(ND^2 + D^3)$ cost) and then estimating the posterior covariance by computing the inverse-Hessian at the mode, again incurring an $O(ND^2 + D^3)$ cost. By contrast, getting an unbiased full-rank estimate of the covariance using MCMC requires generating at least D samples, which again costs $O(ND^2)$. If $N > D$, then this is within a constant cost of the classical approach.

However, it is conceivable that Polyak averaging (Section 6.2) could be used to estimate the first few principal components of the posterior relatively quickly (i.e., in $O(ND)$ time). This corresponds to finding the smallest principal components of A, which cannot be done efficiently in general. A related question is investigated experimentally in Section 7.2.

The analysis above implies an upper bound on the efficiency of stochastic-gradient MCMC (SGMCMC) methods. The argument is this: given that assumptions 1–5 hold, suppose that there exists an SGMCMC method that, for large N, is able to generate effectively independent posterior samples using $d < O(N)$ operations. Then, if we wanted to estimate the posterior mode, we could simply average some large number M of those samples to obtain an estimator whose covariance would be $\frac{1}{M}A^{-1}$. This approach would require dM operations, whereas iterate averaging would require $O(MN) > dM$ operations to obtain an estimator with the same covariance. But this contradicts the result of Polyak and Juditsky (1992) that *no* stochastic-gradient-oracle algorithm can

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 7.1. To estimate the KL divergence, we fitted a multivariate Gaussian to the iterates of our sampling algorithms and used a Laplace approximation for the posterior. 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.

outperform iterate averaging. Thus, the optimality of iterate averaging as an optimization algorithm, taken with assumptions 1–5, implies that *no SGMCMC algorithm can generate posterior samples in sublinear time*[2].

This argument relies on assumptions 1–5 being true, but one can easily construct scenarios in which they are violated. However, these assumptions are all optimistic; there seems (to us) little reason to think that problems that violate assumptions 1–5 will be easier than those that do not.

7. Experiments

We test our theoretical assumptions from Section 3 and find good experimental evidence that they are reasonable in some settings. We also investigate iterate averaging and show that the assumptions outlined in 6.2 result in samples from a close approximation to the posterior. We also compare against other approximate inference algorithms, including SGLD (Welling and Teh, 2011), NUTS (Hoffman and Gelman, 2014), and black-box variational inference (BBVI) using Gaussian reparametrization gradients (Kucukelbir et al., 2015). In Section 7.3 we show that constant SGD lets us optimize hyperparameters in a Bayesian model.

7.1 Confirming the Stationary Distribution's 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.

Real-world data. We first considered the following data sets.

- The *Wine Quality Data Set*[3], containing $N = 4,898$ instances, 11 features, and one integer output variable (the wine rating).

---

2. At least, not without exploiting additional knowledge about the source of gradient noise as do methods like SVRG and SAGA (Johnson and Zhang, 2013; Defazio et al., 2014).

3. P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis, 'Wine Quality Data Set', UCI Machine Learning Repository.
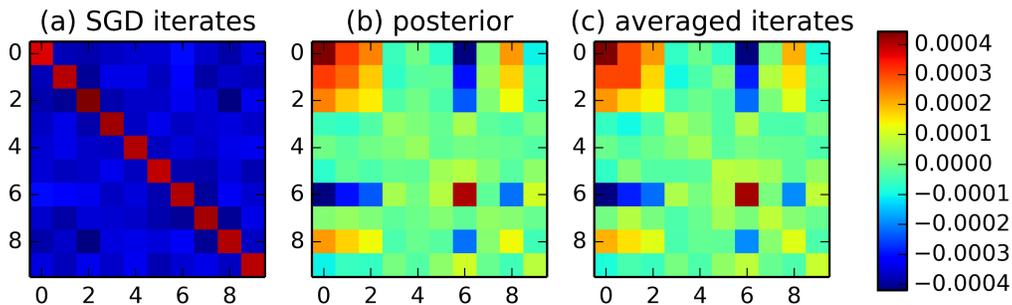
Figure 3: Iterate averaging on linear regression, where we generated artificial data as generated from the model. (a) shows the empirical covariance of the iterates of SGD, whereas (c) shows the averaged iterates with optimally chosen time window. The resulting covariance strongly resembles the true posterior covariance in (b). This shows that iterate averaging may result in posterior sampling.

- A data set of *Protein Tertiary Structure*[4], containing $N = 45,730$ instances, 8 features and one output variable.

- The *Skin Segmentation Data Set*[5], containing $N = 245,057$ 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 size $S = 100$, $S = 100$ and $S = 10000$ for the three data sets, respectively. The quadratic regularizer was 1. The constant learning rate was adjusted according to Eq. 15.

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. Since the predicted covariance is based on approximating SGD as a multivariate Ornstein-Uhlenbeck process, we conclude that our modeling assumptions are satisfied to a very good extent. Since the wine dataset is smaller than the skin segmentation data set, it has a broader posterior and therefore requires a larger learning rate to match it. For this reason, discretization effects play a bigger role and the stationary distribution of preconditioned SGD on wine does not exactly match the posterior. The unprojected 11-dimensional covariances on wine data are also compared in Fig. 2. The rightmost column 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)

4. Prashant Singh Rana, 'Protein Tertiary Structure Data Set', UCI Machine Learning Repository.
5. Rajen Bhatt, Abhinav Dhall, 'Skin Segmentation Dataset', UCI Machine Learning Repository.

and BBVI. For SG Fisher Scoring, we set the learning rate to ϵ^* of Eq. 15, 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). Table 1 summarizes the results. 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.
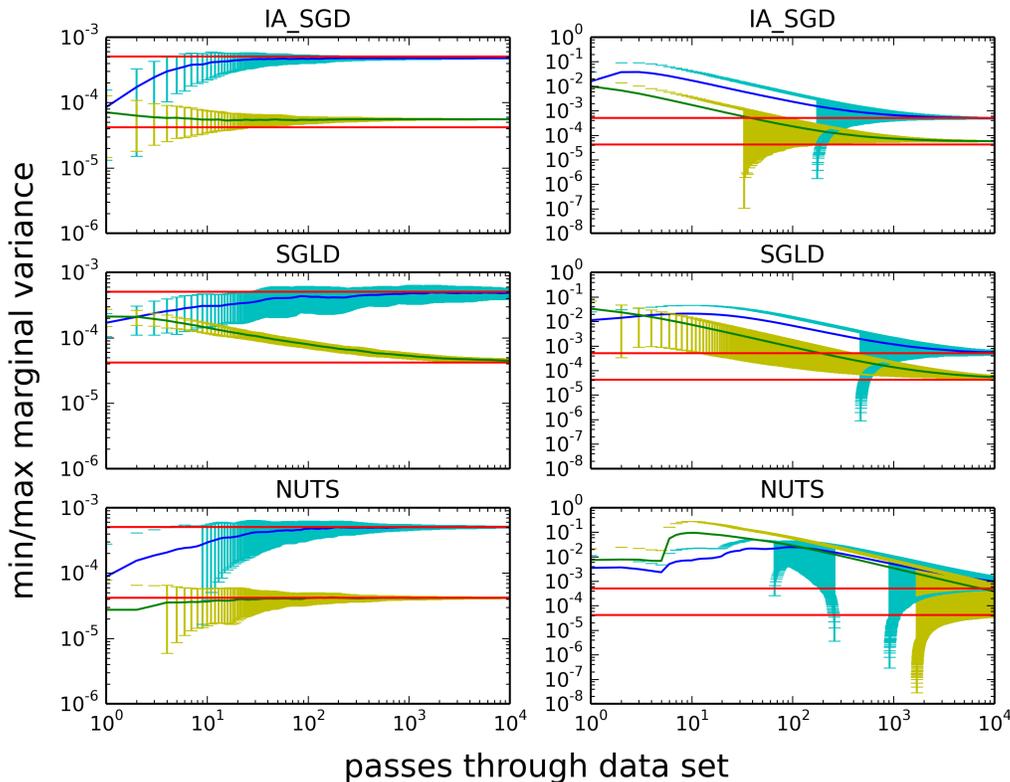
Figure 4: Convergence speed comparison between IASG (top), SGLD (middle), and NUTS (bottom) on linear regression. The plots show minimal (yellow) and maximal (blue) posterior marginal variances, respectively, as a function of iterations, measured in units of passes through the data. Error bars denote one standard deviation. Red solid lines show the ground truth. Left plots were initialized in the posterior maximum, whereas in the right column, we initialized randomly.

7.2 Iterate Averaging as Approximate MCMC

In the following, we show that under the assumptions specified in Section 6.2, iterate averaging with a constant learning rate and fixed averaging window results in samples from the posterior.

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 $D = 10$
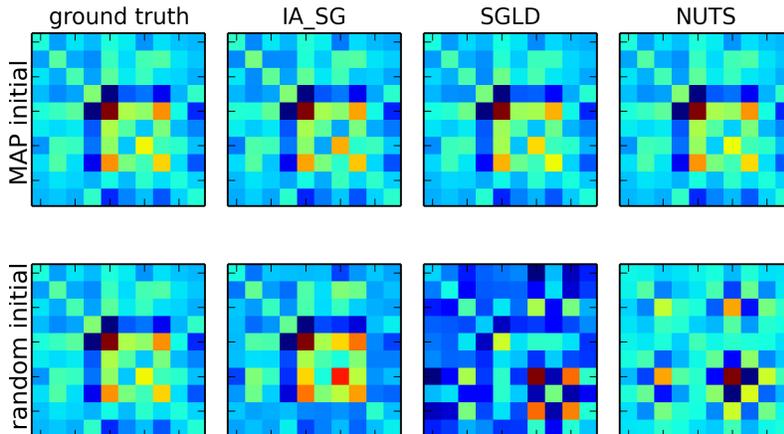
Figure 5: Posterior covariances as estimated by different methods, see also Fig. 4. The top row shows results where we initialized the samplers in the maximum posterior mode. The bottom rows were initialized randomly. For MAP initialization, all samplers find a good estimate of the posterior. When initializing randomly, IASG performs better than NUTS and SGLD.

dimensional Gaussian with unit covariance. We drew the true weight vector from the prior. We then generated the corresponding response variables from the linear regression model.

In a first experiment, we confirmed that iterate averaging may result in samples from the exact posterior, shown in Fig. 3. The left panel shows the empirical covariance matrix of the iterates of SGD. The middle and right panel show the posterior covariance and empirical covariance of the averaged iterates, respectively. As predicted by our theory, there is a very strong resemblance which demonstrates that constant-rate iterate averaging with the right rates and averaging windows may actually result in samples from the posterior. To generate this plot, we then ran constant SGD with a constant learning rate $\epsilon = 0.005$ for $10 * D * N = 10^6$ iterations and a minibatch size $S = 1$ and used an averaging window of $N/S = 10^4$, as predicted by the theory presented earlier in order to achieve samples from the posterior.

Next, we analyzed the convergence speed of Iterate-Averaged Stochastic Gradients (IASG, see Algorithm 1) compared to related methods. Fig. 4 shows these comparisons for three modern scalable MCMC algorithms: SGLD (Welling and Teh, 2011) and NUTS (Hoffman and Gelman, 2014). We investigated how quickly these algorithms could give us estimates of the posterior covariances. To better visualize the convergence behavior, we focussed on diagonal entries of the posterior covariance, the marginal variances. As before, the data were generated from the model such the theoretical assumptions of 6.2 applied. We then ran the samplers for up to $10^3 * D$ effective passes through the data. For IASG and SGLD we used a minibatch size of $S = 10$ and an averaging window of $N/S = 1000$. The constant learning rate of IASG was $\epsilon = 0.003$ and for SGLD we decreased the learning rate according to the Robbins-Monro schedule of $\epsilon_t = \frac{\epsilon_0}{\sqrt{1000 + t}}$ where we found $\epsilon_0 = 10^{-3}$ to be optimal. NUTS automatically adjusts its learning rate and uses non-stochastic gradients.

The left column of Fig. 4 shows the results of these experiments. We found the convergence speeds of the samplers to be highly dependent on whether we optimized the samplers in the maximum posterior mode (termed MAP-initialization: this partially eleminates the initial bias and burn-in phase) or whether the samplers were initialized randomly, as in a real-world application. Thus, we show both results: while the the right column shows random initializations, the left one shows MAP-initialization. The rows of Fig. 4 show results of IASG (top), SGLD (middle), and NUTS (bottom). Each entry shows the smallest and largest marginal variance of the posterior over iterations, as estimated from these methods, where we excluded the first 10 iterations to avoid large biases. We also give the standard deviations for these estimates based on 100 independent Markov chains. The solid red lines show the ground truth of these marginal variances. Fig. 5 shows additional results on the same experiments, where we display the posterior estimates of the three different samplers under the two different initializations.

We found that in both initializations, IASG can find a fast approximate solution to the posterior. It uses stochastic gradients which gives it a competitive advantage over NUTS (which uses full gradients) in particular in the early search phase of the sampler. Langevin dynamics behaves similarly at early iterations (it also employs stochastic gradients). However, compared to IASG, we see that the Langevin algorithm has a much larger standard error when estmating the posterior covariances. This is because it uses a decreasing Robbins-Monro learning rate that slows down equilibration at long times. In contrast, IASG uses a constant learning rate and therefore converges fast. Note that in practice a large variance may be as bad as a large bias, especially if posterior estimates are based on a single Markov chain. This is also evident in Fig. 5, which shows that for random initialization, both NUTS and SGLD reveal less of the structure of the true posterior covariance compared to IASG.

When initialized in the posterior maximum, we see that all algorithms perform reasonably well (though SGLD's estimates are still highly variable even after $10,000$ sweeps through the dataset). IASG converges to a stable estimate much faster than SGLD or NUTS, but produces slightly biased estimates of the smallest variance.

7.3 Optimizing Hyperparameters

We test the hypothesis of Section 4.2, namely that constant SGD as a variational algorithm gives rise to a variational EM algorithm where we jointly optimize hyperparameters using gradients while drawing samples form an approximate posterior. To this end, we experimented with a Bayesian multinomial logistic (a.k.a. softmax) regression model with normal priors. The negative log-joint is

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

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. Eq. 35 has the degenerate maximizer $\lambda = \infty, \theta = 0$, which has infinite posterior density which we hope to avoid in our approach.

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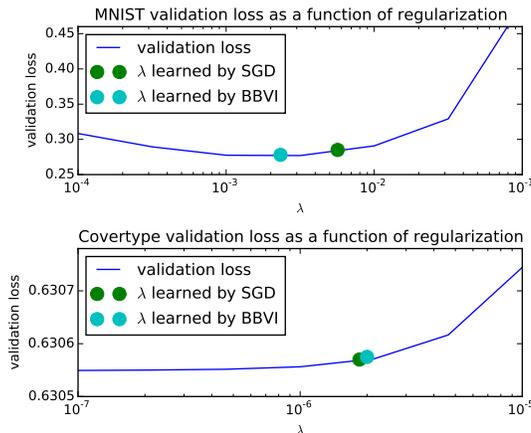
Figure 6: Validation loss as a function of L2 regularization parameter λ. Circles show the values of λ that were automatically selected by SGD and BBVI.

Fig. 6 shows the validation loss achieved by maximizing equation 35 over θ for various values of λ. This gives rise to the continuous blue curve. The value for constant SGD was obtained using Eq. 22, hence using constant learning rates for θ and decreasing learning rates for λ. BBVI was carried out by optimizing a variational lower bound in mean-field variational inference, and optimizing hyperparameters based on this lower bound.

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.

8. Conclusions

In this paper, we built on a stochastic process perspective of stochastic gradient descent and various extensions to derive several new results. Under specified assumptions, SGD is approximated by a multivariate Ornstein-Uhlenbeck process, which possesses an analytic solution. We computed the stationary distribution of constant SGD and analyzed its properties.

We analyzed SGD together with several extensions, such as momentum, preconditioning, and iterate averaging. The shape of the stationary distribution is controlled by the parameters of the algorithm such as the constant learning rate, preconditioning matrix, or averaging period. We can thus tune these parameters to minimize the Kullback-Leibler divergence between the stationary distribution and a Bayesian posterior. This view uses these stochastic optimization algorithms as approximate inference. We also analyzed stochastic gradient Langevin dynamics and stochastic gradient Fisher scoring and were able to analyze approximation errors for these algorithms.

The Bayesian view on constant-rate SGD allows us to use this algorithm as a new variational EM algorithm. We suggested and tested a double SGD scheme which uses decreasing learning rates on the hyperparameters and constant learning rates on the main parameters. We showed that this is both easy to implement and prevents us from finding degenerate solutions; it is a cheap alternative to cross-validation for many complex models.

Last, our analysis suggests the many similarities between sampling and optimization algorithms that can be explored using the stochastic process perspective. A future direction might be to explore

26

similarities in the noise characteristics of black-box variational inference algorithms and Langevin-type MCMC. Further exploring the use of iterate averaging as a Bayesian algorithm is another interesting avenue for further studies.

Acknowledgments

Appendix A. Examples: Ornstein-Uhlenbeck Formalism

Let us illustrate the Ornstein-Uhlenbeck formalism based on two simple examples. First, consider the following quadratic loss,

$$\mathcal{L}(\theta) = -\tfrac{1}{2N} \sum_{n=1}^{N} ||x_n - \theta||^2. \tag{36}$$

Let us define $\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n$ as the empirical mean of the data points. The gradient is $g(\theta) = (\bar{x} - \theta)$, and the stochastic gradient is $\hat{g}(\theta) = \frac{1}{S} \sum_{s=1}^{S}(x_s - \theta)$. Because the gradient is linear in x, the noise covariance is just the covariance of the data: $\Sigma_x \equiv C/S = \frac{1}{S^2}\mathbb{E}\left[\sum_{s,s'}(x_s - \bar{x})(x_{s'} - \bar{x})^\top\right] = \frac{1}{S}\mathbb{E}[(x_n - \bar{x})(x_n - \bar{x})^\top]$. We can shift the parameter $\theta \to \theta + \bar{x}$, resulting in $\theta^* = 0$. Note that the Hessian $A \equiv \mathbf{I}$ is just unity. According to Eq. 38,

$$q(\theta) \propto \exp\left\{-\tfrac{S}{2\epsilon}(\theta - \bar{x})^\top \Sigma_x^{-1}(\theta - \bar{x})\right\}.$$

as the resulting stationary distribution. Next, consider linear regression, where we minimize

$$\mathcal{L}(\theta) = -\tfrac{1}{2N} \sum_n (y_n - x_n^\top \theta)^2. \tag{37}$$

We can write the stochastic gradient as $\hat{g} = \hat{A}\theta - \hat{\mu}$, where $\hat{\mu} = \frac{1}{S}\sum_s x_s y_s$ and $\hat{A} = \frac{1}{S}\sum_s x_s x_s^\top$ are estimates based on a mini-batch of size S. The sampling noise covariance is $C(\theta) = \mathbb{E}[(\hat{g} - g)(\hat{g} - g)^\top] = \mathbb{E}[\hat{g}\hat{g}^\top] - gg^\top$, where $\mathbb{E}[\hat{g}\hat{g}^\top] = \mathbb{E}[(\hat{A}\theta - \hat{\mu})(\hat{A}\theta - \hat{\mu})^\top]$. We see that the noise covariance is quadratic, but unfortunately it cannot be further simplified.

Fig. 1 shows the objective function of linear regression (blue) and the sampling distribution of stochastic gradient descent (yellow) on simulated data. We see that both distributions do not coincide, because the sampling distribution is also affected by the noise covariance.

Appendix B. Stationary Covariance

The Ornstein-Uhlenbeck process has an analytic solution in terms of the stochastic integral (Gardiner et al., 1985),

$$\theta(t) = \exp(-At)\theta(0) + \sqrt{\tfrac{\epsilon}{S}} \int_0^t \exp[-A(t - t')]BdW(t') \tag{38}$$

Following Gardiner's book and using $A = A^\top$, we derive an algebraic relation for the stationary covariance of the multivariate Ornstein-Uhlenbeck process. Define $\Sigma = \mathbb{E}[\theta(t)\theta(t)^\top]$. Using the

formal solution for $\theta(t)$ given in the main paper, we find

$$
\begin{aligned}
A\Sigma + \Sigma A &= \tfrac{\epsilon}{S} \int_{-\infty}^{t} A \exp[-A(t-t')]BB^\top \exp[-A(t-t')]dt' \\
&\quad + \tfrac{\epsilon}{S} \int_{-\infty}^{t} \exp[-A(t-t')]BB^\top \exp[-A(t-t')]dt' A \\
&= \tfrac{\epsilon}{S} \int_{-\infty}^{t} \frac{d}{dt'} \left(\exp[-A(t-t')]BB^\top \exp[-A(t-t')] \right) \\
&= \tfrac{\epsilon}{S} BB^\top .
\end{aligned}
$$

We used that the lower limit of the integral vanishes by the positivity of the eigenvalues of A.

Appendix C. Reverse KL Divergence Setup

It is interesting to also consider the case of trying to minimize the reverse KL divergence, i.e. $KL(f||q)$ instead of $KL(q||f)$. One might assume that this is possible since both the variational distribution and the posterior are assumed to be Gaussian. This turns out to lead only to a feasible algorithm in the special case where the Hessian in the optimum A and the stationary covariance Σ commute. In more detail, the KL-divergence between the posterior and the stationary distribution is (up to constants):

$$
\begin{aligned}
KL(f||q) &= \mathbb{E}_f[\log f] - \mathbb{E}_f[\log q] \qquad\qquad\qquad (39) \\
&= \tfrac{1}{2}\mathbb{E}_f\left[\theta^\top \Sigma^{-1}\theta\right] + \tfrac{1}{2}\log|\Sigma| + \text{const.} \\
&= \tfrac{1}{2N}\text{Tr}(A^{-1}\Sigma^{-1}) + \tfrac{1}{2}\log|\Sigma| + \text{const.}
\end{aligned}
$$

While we were able to derive this divergence, it turns out that we cannot in general eliminate its dependence in the stationary covariance and re-express it in terms of $BB^\top$, using Eq. 13. However, if A and Σ commute, we can proceed as follows:

$$
\begin{aligned}
\text{Tr}(A^{-1}\Sigma^{-1}) &= \text{Tr}((\Sigma A)^{-1}) \qquad\qquad\qquad\qquad (40) \\
&\overset{A\Sigma = \Sigma A}{=} 2\text{Tr}((\Sigma A + A\Sigma)^{-1}) \\
&\overset{Eq.\ 13}{=} \tfrac{2S}{\epsilon}\text{Tr}((BB^\top)^{-1})
\end{aligned}
$$

Following the logic of Theorems 1 and 2, we find the following result for the optimal learning rate:

$$
\epsilon^* = \frac{2S}{ND}\text{Tr}((BB^\top)^{-1}). \qquad\qquad\qquad (41)
$$

Interestingly, when comparing Eq. 15 with Eq. 41, we find that the inverse of the trace of the noise covariance gets replaced by the trace of the inverse noise covariance. While $KL(q||f)$ suggests to choose the learning rate inversely proportional to the largest Eigenvalue of the noise, $KL(f||q)$ thus suggests to choose the learning rate proportional to the inverse of the smallest Eigenvalue of the noise. Both approaches have thus a different emphasis on how to fit the posterior mode, in a similar fashion as variational inference and expectation propagation. Note, however, that A and Σ rarely commute in practice, and thus $KL(q||f)$ is the only viable option.

Appendix D. SGD With Momentum

Here, we give more details on the deviations of the results on SGD with momentum. In order to compute the stationary distribution of Eq. 23, we need to solve the equations for the second moments:

$$
\begin{aligned}
d\mathbb{E}[\theta\theta^\top] &= \mathbb{E}[d\theta\theta^\top + \theta d\theta^\top] \\
&= (\mathbb{E}[v\theta^\top] + \mathbb{E}[\theta v^\top])dt, & (42) \\
d\mathbb{E}[\theta v^\top] &= \mathbb{E}[d\theta v^\top + \theta dv^\top] \\
&= \mathbb{E}[vv^\top]dt - \mu\mathbb{E}[\theta v^\top]dt - \epsilon\mathbb{E}[\theta\theta^\top]Adt, & (43) \\
d\mathbb{E}[v\theta^\top] &= \mathbb{E}[dv\theta^\top + vd\theta^\top] \\
&= \mathbb{E}[vv^\top]dt - \mu\mathbb{E}[v\theta^\top]dt - \epsilon A\mathbb{E}[\theta\theta^\top]dt, & (44) \\
d\mathbb{E}[vv^\top] &= \mathbb{E}[dvv^\top + vdv^\top] + \mathbb{E}[dvdv^\top] \\
&= -2\mu\mathbb{E}[vv^\top]dt - \epsilon A\mathbb{E}[\theta v^\top]dt - \epsilon\mathbb{E}[v\theta^\top]Adt + \tfrac{\epsilon^2}{S}BB^\top dt. & (45)
\end{aligned}
$$

In the last equation we used the fact that according to Ito's rule, there is an additional non-vanishing contribution due to the noise, $\mathbb{E}[dvdv^\top] = \frac{\epsilon^2}{S}\mathbb{E}[B\,dW\,dW^\top B^\top] = \frac{\epsilon^2}{S}BB^\top dt$. This contribution does not exist for the other correlators; for more details see e.g. (Gardiner et al., 1985).

Since we are looking for a stationary solution, we set the left hand sides of all equations to zero. Eq. 42 implies that $\mathbb{E}[v\theta^\top] + \mathbb{E}[\theta v^\top] = 0$, hence the cross-correlation between momentum and position is anti-symmetric in the stationary state. We can thus add Eqs. 43 and 44 to find $0 = d\mathbb{E}[v\theta^\top + \theta v^\top] = 2\mathbb{E}[vv^\top]dt - \epsilon A\mathbb{E}[\theta\theta^\top]dt - \epsilon\mathbb{E}[\theta\theta^\top]Adt$. Combining this with Eq. 45 yields

$$
\begin{aligned}
\mathbb{E}[vv^\top] &= \tfrac{\epsilon}{2}\mathbb{E}[\theta\theta^\top]A + \tfrac{\epsilon}{2}A\mathbb{E}[\theta\theta^\top] \\
\mu\mathbb{E}[vv^\top] &= \tfrac{\epsilon^2}{2S}BB^\top - \underbrace{\tfrac{1}{2}\epsilon(A\mathbb{E}[\theta v^\top] + \mathbb{E}[v\theta^\top]A)}_{=0}.
\end{aligned}
$$

Last, we show that the underbraced term is zero, which gives Eq. 25 in the main paper. Denote $\xi = \mathbb{E}[\theta v^\top] = -\xi^\top$ which is antisymmetric due to Eq. 42. First of all, $A\xi + \xi^\top A$ is obviously symmetric. It is simultaneously antisymmetric due to the following calculation: $\sum_k A_{ik}\xi_{kj} + \sum_k \xi_{ik}^\top A_{kj}^\top = \sum_k A_{ik}\xi_{kj} - \sum_k \xi_{ki}A_{jk} = \sum_k A_{ik}\xi_{kj} - \sum_k A_{jk}\xi_{ki}$. This term swaps the sign as i and j are interchanged. Being both symmetric and antisymmetric, it is zero.

Appendix E. Stochastic Gradient Fisher Scoring

We start from the Ornstein-Uhlenbeck process with minibatch size $S = 1$,

$$
\begin{aligned}
d\theta(t) &= -\epsilon HA\theta(t)dt + H\left[\epsilon BdW(t) + \sqrt{\epsilon}EdV(t)\right] \\
&\Leftrightarrow \\
d\theta(t) &= -A'\theta(t)dt + B'dW(t),
\end{aligned}
$$

where we define $A' \equiv \epsilon HA$ and $B' \equiv H\sqrt{\epsilon^2 BB^\top + \epsilon EE^\top}$, and use the fact that the dynamics of $CdW(t) + DdV(t)$ are equivalent to $\sqrt{CC^\top + DD^\top}dW(t)$. Here we are using matrix square roots, so $B'B'^\top = \epsilon H(\epsilon BB^\top + EE^\top)H$.

As derived in the paper, the variational bound is (up to a constant)

$$KL \stackrel{\text{c}}{=} \frac{N}{2}\text{Tr}(A\Sigma) - \tfrac{1}{2}\log(|\Sigma|).$$

To evaluate it, the task is to remove the unknown covariance Σ from the bound. To this end, as before, we use the identity for the stationary covariance $A'\Sigma + \Sigma A'^{\top} = B'B'^{\top}$. The criterion for the stationary covariance is equivalent to

$$HA\Sigma + \Sigma AH \;\; = \;\; \epsilon HBB^{\top}H + HEE^{\top}H.$$

We can further simplify this expression as follows:

$$
\begin{aligned}
A\Sigma + H^{-1}\Sigma AH &= \epsilon BB^{\top}H + EE^{\top}H \\
\Rightarrow \text{Tr}(A\Sigma) &= \frac{1}{2}\text{Tr}(H(\epsilon BB^{\top} + EE^{\top})).
\end{aligned}
$$

As above, we can also reparameterize the covariance as $\Sigma = TH$, so that T does not depend on H:

$$
\begin{aligned}
HA\Sigma + \Sigma AH &= \epsilon HBB^{\top}H + HEE^{\top}H \\
A\Sigma H^{-1} + H^{-1}\Sigma A &= \epsilon BB^{\top} + EE^{\top} \\
AT + T^{\top}A &= \epsilon BB^{\top} + EE^{\top}.
\end{aligned}
$$

The KL divergence is therefore

$$
\begin{aligned}
KL &= \frac{N}{2}\text{Tr}(A\Sigma) - \frac{D}{2} - \frac{1}{2}\log(N|A|) - \frac{1}{2}\log|\Sigma| \\
&= \frac{N}{4}\text{Tr}(H(\epsilon BB^{\top} + EE^{\top})) - \frac{D}{2} - \frac{1}{2}\log(N|A|) - \frac{1}{2}\log|T| - \frac{1}{2}\log|H|, \quad (46)
\end{aligned}
$$

which is the result we give in the main text.

Appendix F. Square Root Preconditioning

We analyze the case where we precondition with a matrix that is proportional to the square root of the diagonal entries of the noise covariance.

We define

$$G \;\; = \;\; \sqrt{\text{diag}(BB^{\top})}$$

as the diagonal matrix that contains square roots of the diagonal elements of the noise covariance. We use an additional scalar learning rate ϵ.

Theorem 7 (Taking square roots) *Consider SGD preconditioned with G^{-1} as defined above. Under the previous assumptions, the constant learning rate which minimizes KL divergence between the stationary distribution of this process and the posterior is*

$$\epsilon^{*} \;\; = \;\; \frac{2DS}{N\text{Tr}(BB^{\top}G^{-1})}. \tag{47}$$

Proof We read off the appropriate KL divergence from the proof of Theorem 2 with $G^{-1} \equiv H$:

$$KL(q||f) \overset{c}{=} \frac{\epsilon N}{2S}\text{Tr}(BB^\top G^{-1}) - \text{Tr}\log(G) + \frac{D}{2}\log\frac{\epsilon}{S} - \frac{1}{2}\log|\Sigma|$$

Minimizing this KL divergence over the learning rate ϵ yields Eq. 47. $\blacksquare$

Appendix G. Iterage Averaging

We now prove our result for the covariance of the averaged iterates. We first need an identity for the non-equal-time covariance in the stationary state:

$$\mathbb{E}[\theta(t)\theta(s)^\top] = \begin{cases} \Sigma e^{-\epsilon A(s-t)} & t < s \\ e^{-\epsilon A(t-s)}\Sigma & t \geq s. \end{cases} \tag{48}$$

To derive it, one uses the formal solution of the Ornstein-Uhlenbeck process for $\theta(t)$ in combination with Eq. 29, see also (Gardiner et al., 1985) for more details. Note that for $t = s$, it simplifies to $\mathbb{E}[\theta(t)\theta(t)^\top] = \Sigma$, as one would expect.

We are averaging over T time steps. Going to the continuous-time OU formalism, we are interested in the following quantity, which is the equal-time covariance of the time-averaged iterates:

$$D \equiv \mathbb{E}\left[\left(\frac{1}{T}\int_0^T \theta(t)dt\right)\left(\frac{1}{T}\int_0^T \theta(s)ds\right)^\top\right].$$

This can be further broken down to two contributions:

$$D = \frac{1}{T^2}\int_0^T\int_0^T \mathbb{E}[\theta(t)\theta(s)^\top]dsdt$$

$$= \frac{1}{T^2}\int_0^T\int_0^t \mathbb{E}[\theta(t)\theta(s)^\top]dsdt + \frac{1}{T^2}\int_0^T\int_t^T \mathbb{E}[\theta(t)\theta(s)^\top]dsdt$$

We now use the eigendecomposition $A = U\Lambda U^\top$ of the Hessian, the autocorrelation Eq. 48, as well as the identity $e^{cA} = Ue^{c\Lambda}U^\top$. The first term becomes

$$\frac{1}{T^2}\int_0^T\int_0^t \mathbb{E}[\theta(t)\theta(s)^\top]dsdt = \frac{1}{T^2}\int_0^T\int_0^t e^{-\epsilon A(t-s)}\Sigma dsdt$$

$$= \frac{1}{T^2}\int_0^T\int_0^t Ue^{-\epsilon\Lambda(t-s)}U^\top\Sigma dsdt$$

$$= \frac{1}{\epsilon T^2}\int_0^T U\Lambda^{-1}(\mathbf{I} - e^{-\epsilon t\Lambda})U^\top\Sigma dt \tag{49}$$

$$= \frac{1}{\epsilon T}A^{-1}\Sigma + \frac{1}{\epsilon^2 T^2}U\Lambda^{-2}(e^{-\epsilon T\Lambda} - \mathbf{I})U^\top\Sigma.$$

The calculation for the second term goes analogously and yields

$$\frac{1}{T^2}\int_0^T\int_t^T \mathbb{E}[\theta(t)\theta(s)^\top]dsdt = \frac{1}{\epsilon T}\Sigma A^{-1} + \frac{1}{\epsilon^2 T^2}\Sigma U\Lambda^{-2}(e^{-\epsilon T\Lambda} - \mathbf{I})U^\top.$$

31

Both equations combined give us

$$D = \frac{1}{\epsilon T}(A^{-1}\Sigma + \Sigma A^{-1})$$
$$+ \frac{1}{\epsilon^2 T^2}(U\Lambda^{-2}(e^{-\epsilon T\Lambda} - \mathbf{I})U^{\top}\Sigma + \Sigma U\Lambda^{-2}(e^{-\epsilon T\Lambda} - \mathbf{I})U^{\top}). \tag{50}$$

When $\epsilon T\Lambda \gg 1$ (valid for sufficiently long averaging periods T), we obtain

$$D \approx \frac{1}{\epsilon T}(A^{-1}\Sigma + \Sigma A^{-1}),$$

which is Eq. 30 in the main text. We can also simplify the expression for $\Sigma = \frac{\epsilon}{2S}I$, as motivated in Section 6.2. In this case, Eq. 50 results in

$$D = \frac{1}{ST}A^{-1} + \frac{1}{\epsilon ST^2}(U\Lambda^{-2}(e^{-\epsilon T\Lambda} - \mathbf{I})U^{\top}).$$

This is exactly Eq. 33 in the main text.

References

Sungjin Ahn, Anoop Korattikara, and Max Welling. Bayesian posterior sampling via stochastic gradient Fisher scoring. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1591–1598, July 2012.

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

Louis Bachelier. *Théorie de la spéculation*. Gauthier-Villars, 1900.

Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer New York, 2006.

Léon Bottou. Online learning and stochastic approximations. *Online Learning in Neural Networks*, 17(9):25, 1998.

Nicolas Bouleau and Dominique Lepingle. *Numerical methods for stochastic processes*, volume 273. John Wiley & Sons, 1994.

Changyou Chen, Nan Ding, and Lawrence Carin. On the convergence of stochastic gradient MCMC algorithms with high-order integrators. In *Advances in Neural Information Processing Systems*, pages 2269–2277, 2015.

Changyou Chen, David Carlson, Zhe Gan, Chunyuan Li, and Lawrence Carin. Bridging the gap between stochastic gradient MCMC and stochastic optimization. In *Artificial Intelligence and Statistics*, 2016.

Tianqi Chen, Emily B Fox, and Carlos Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *Proceedings of The 31st International Conference on Machine Learning*, pages 1683–1691, 2014.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.

Alexandre Défossez and Francis Bach. 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, 2015.

Aymeric Dieuleveut, Alain Durmus, and Francis Bach. Bridging the gap between constant step size stochastic gradient descent and markov chains. *arXiv preprint arXiv:1707.06386*, 2017.

Nan Ding, Youhan Fang, Ryan Babbush, Changyou Chen, Robert D Skeel, and Hartmut Neven. Bayesian sampling using stochastic gradient thermostats. In *Advances in Neural Information Processing Systems*, pages 3203–3211, 2014.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

Nicolas Flammarion and Francis Bach. From averaging to acceleration, there is only a step-size. In *Proceedings of the International Conference on Learning Theory*, 2015.

Crispin W Gardiner et al. *Handbook of Stochastic Methods*, volume 4. Springer Berlin, 1985.

Zoubin Ghahramani and Matthew J Beal. Propagation algorithms for variational Bayesian learning. In *Advances in Neural Information Processing Systems*, pages 507–513, 2000.

Matthew D Hoffman and Andrew Gelman. The no-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

Matthew D Hoffman, David M Blei, Chong Wang, and John William Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, 2013.

Michael Jordan, Zoubin Ghahramani, Tommi Jaakkola, and Lawrence Saul. Introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *The International Conference on Learning Representations*, 2014.

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

Alp Kucukelbir, Rajesh Ranganath, Andrew Gelman, and David Blei. Automatic variational inference in STAN. In *Advances in Neural Information Processing Systems*, pages 568–576, 2015.

Harold J Kushner and George Yin. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media, 2003.

Lucien Le Cam. Asymptotic methods in statistical decision theory. *Springer New York*, 1986.

Qianxiao Li, Cheng Tai, and Weinan E. Dynamics of stochastic gradient algorithms. *arXiv preprint arXiv:1511.06251v2*, 2015.

Qianxiao Li, Cheng Tai, and Weinan E. Stochastic modified equations and adaptive stochastic gradient algorithms. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2101–2110, 2017.

Lennart Ljung, Georg Ch Pflug, and Harro Walk. *Stochastic approximation and optimization of random systems*, volume 17. Birkhäuser, 2012.

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

Yi-An Ma, Tianqi Chen, and Emily B Fox. A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems*, 2015.

Dougal Maclaurin, David Duvenaud, and Ryan P Adams. Early stopping is nonparametric variational inference. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 1070–1077, 2016.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

Harry Nyquist. Thermal agitation of electric charge in conductors. *Physical Review*, 32:110–113, Jul 1928.

Manfred Opper and David Saad. *Advanced mean field methods: Theory and practice*. MIT press, 2001.

Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

Boris Teodorovich Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

Rajesh Ranganath, Sean Gerrish, and David M Blei. Black box variational inference. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, pages 814–822, 2014.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

Francisco Ruiz, Michaelis Titsias, and David Blei. The generalized reparameterization gradient. In *Advances in Neural Information Processing Systems*, pages 460–468, 2016.

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

Tim Salimans and David A Knowles. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882, 2013.

Issei Sato and Hiroshi Nakagawa. 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, 2014.

Xiaocheng Shang, Zhanxing Zhu, Benedict Leimkuhler, and Amos J Storkey. Covariance-controlled adaptive Langevin thermostat for large-scale Bayesian sampling. In *Advances in Neural Information Processing Systems*, pages 37–45, 2015.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1139–1147, 2013.

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5—RmsProp: Divide the Gradient by a Running Average of its Recent Magnitude. Coursera: Neural Networks for Machine Learning, 2012.

Panagiotis Toulis, Edoardo Airoldi, and Jason Rennie. 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, 2014.

Panos Toulis, Dustin Tran, and Edoardo M Airoldi. Towards stability and optimality in stochastic gradient descent. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 1290–1298, 2016.

George E Uhlenbeck and Leonard S Ornstein. On the theory of the Brownian motion. *Physical Review*, 36(5):823, 1930.

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

Bernard Widrow and Samuel D Stearns. Adaptive signal processing. *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1985, 491 p.*, 1, 1985.

Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the 21st International Conference on Machine Learning*, page 116. ACM, 2004.