2nd Symposium on Advances in Approximate Bayesian Inference, 2019 1–10

Langevin Dynamics as Nonparametric Variational Inference

Anonymous Authors
Anonymous Institution

Abstract

Variational inference (VI) and Markov chain Monte Carlo (MCMC) are approximate posterior inference algorithms that are often said to have complementary strengths, with VI being fast but biased and MCMC being slower but asymptotically unbiased. In this paper, we analyze gradient-based MCMC and VI procedures and find theoretical and empirical evidence that these procedures are not as different as one might think. In particular, a close examination of the Fokker-Planck equation that governs the Langevin dynamics (LD) MCMC procedure reveals that LD implicitly follows a gradient flow that corresponds to a variational inference procedure based on optimizing a nonparametric normalizing flow. This result suggests that the transient bias of LD (due to too few warmup steps) may track that of VI (due to too few optimization steps), up to differences due to VI's parameterization and asymptotic bias. Empirically, we find that the transient biases of these algorithms (and momentum-accelerated versions) do evolve similarly. This suggests that practitioners with a limited time budget may get more accurate results by running an MCMC procedure (even if it's far from burned in) than a VI procedure, as long as the variance of the MCMC estimator can be dealt with (e.g., by running many parallel chains).

1. INTRODUCTION

The central computational problem in Bayesian data analysis is posterior inference. Exact inference is usually intractable, so practitioners resort to approximations. Two of the most popular classes of approximate inference algorithms are Markov chain Monte Carlo (MCMC) and variational inference (VI). VI chooses a family of tractable distributions, and tries to find the member of that family with the lowest KL divergence to the posterior, whereas MCMC simulates a Markov chain whose stationary distribution is the posterior.

VI and MCMC are often said to have complementary strengths: VI is faster but biased, whereas MCMC is slower but asymptotically unbiased. But statements like this are imprecise; the question is not "how much longer does MCMC take to converge than VI?" but "for a given computational budget, will VI or MCMC give more accurate estimates?" For that matter, the notion of a one-dimensional computation budget is an oversimplification of modern reality, where parallel computation (especially on GPUs and TPUs) has become cheap but clock speeds have remained nearly constant. MCMC error due to variance (a.k.a. small effective sample sizes) can be reduced by running more parallel chains on more cores without affecting latency, whereas transient bias (a.k.a. incomplete burn-in or warmup) can only be reduced by running longer chains, necessarily increasing latency. Likewise, one can reduce the variance of stochastic-gradient VI estimators using parallel computation in the form of minibatches, but zero-variance gradients do not translate to instant convergence.

In this paper, we will mostly be motivated by the following question: *for a given parallel-compute budget, will VI or MCMC reach a given level of accuracy faster?* We examine this question both theoretically and empirically for two popular gradient-based VI and MCMC algorithms: reparameterized black-box VI (BBVI; Ranganath et al., 2014; Kingma and Welling, 2014; Rezende et al., 2014; Roeder et al., 2017) and Langevin-dynamics MCMC (LD; Roberts and Rosenthal, 1998). By reformulating LD as a deterministic normalizing flow (Rezende and Mohamed, 2015) via the Fokker-Planck equation (Jordan et al., 1998; Villani, 2003), we arrive at a reinterpretation of BBVI as a parametric approximation to the nonparametric LD MCMC procedure. This interpretation suggests that the transient bias (Angelino et al., 2016) of BBVI (i.e., bias due to insufficient optimization) may track the transient bias of LD (i.e., bias due to insufficient burn-in), and suggesting that the claim that VI is faster than MCMC is an oversimplification. Empirically, we find that BBVI's transient bias indeed tracks that of LD on several problems. Our main results are:

- We show theoretically that LD and BBVI both follow the same gradient flow, up to gradient noise and a tangent field induced by the variational parameterization.
- We show empirically that the transient bias of BBVI and MCMC estimators often converges at similar speeds, even when BBVI uses very low-variance gradient estimators and can exactly match the posterior. When BBVI is asymptotically biased, we likewise find similar convergence behavior until this asymptotic bias kicks in.

Taken together, these results have important implications for practitioners choosing between BBVI and gradient-based MCMC algorithms. In particular, we argue that BBVI is unlikely to be significantly faster than MCMC unless we can use an amortized-inference strategy (Gershman and Goodman, 2014) to spread the cost of BBVI across many problems, or we do not have access to enough parallel computation that we can reduce the variance of our MCMC estimator to acceptable levels by running many chains in parallel. Otherwise, as an alternative to BBVI we recommend running as many short MCMC chains as possible, possibly discarding all but the last sample of each chain. As GPUs and TPUs get more powerful, this strategy will apply to more and more one-off Bayesian-data-analysis problems.

2. LANGEVIN AS AN IMPLICIT NORMALIZING FLOW

In this section, we show that the Langevin dynamics (LD) algorithm can be interpreted as implicitly doing black-box variational inference (BBVI) with a nonparametric normalizing flow. One can think of this derivation as a translation of the classic "JKO" result of Jordan et al. (1998) to the language of modern flow-based variational inference (Rezende and Mohamed, 2015). The result will make it clear that gradient-based MCMC algorithms and parametric BBVI are following essentially the same gradient signals, up to a tangent field introduced by the mapping from function space to parameter space.

We begin by considering BBVI with a *nonparametric* normalizing flow g, and taking the functional derivative of the Kullback-Leibler (KL) divergence between the resulting variational distribution $q_g(\theta) = q_0(g^{-1}(\theta))|\frac{\partial g^{-1}}{\partial \theta}|$ and the target distribution $p(\theta)$:

$$\frac{d}{\delta g(\epsilon)} \int_\epsilon q_0(\epsilon)(\log q_g(g(\epsilon)) - \log p(g(\epsilon)))d\epsilon = \nabla_\theta \log q_g(g(\epsilon)) - \nabla_\theta \log p(g(\epsilon)). \tag{1}$$

(Following Roeder et al. (2017) we omit the zero-expectation score-function term capturing the effect of g on $q_g(\cdot)$.) That is, we want to push samples towards regions of high density under p and away from regions of high density under q. If g is instead a parametric function controlled by parameters ϕ (as it almost always is in practice), then the gradient becomes

$$\tfrac{d}{d\phi}\int_\epsilon q_0(\epsilon)(\log q_\phi(g_\phi(\epsilon)) - \log p(g_\phi(\epsilon)))d\epsilon = \int_\epsilon(\nabla_\theta \log q_g(g_\phi(\epsilon)) - \nabla_\theta \log p(g_\phi(\epsilon)))\tfrac{\partial g}{\partial \phi}d\epsilon. \quad (2)$$

That is, the standard BBVI gradient step is the projection of the "ideal" BBVI functional gradient onto the parameter space of g_ϕ.

Next we show that LD implicitly follows the ideal functional gradient in equation 1. In each iteration of LD, we update our state $\theta_n \in \mathbb{R}^D$ to $\theta_{n+1} = \theta_n + \eta\nabla \log p(\theta_n) + \sqrt{2\eta}\xi$, where η is a step-size parameter and $\xi \sim \mathcal{N}(0, I)$ is a standard-normal random variable. This is a first-order discretization[1] of the Langevin stochastic differential equation (SDE) $d\theta = \nabla \log p(\theta(t))dt + \sqrt{2}dW(t)$, where $dW(t)$ is a D-dimensional Wiener process. The distribution $q(t, \theta)$ of a population of particles evolving according to the Langevin SDE from some initial distribution $q(0, \theta)$ is governed by the (deterministic) Fokker-Planck partial differential equation (PDE), which we write (in slightly non-standard form) as

$$\tfrac{\partial \log q}{\partial t} = \nabla_\theta \log q(t, \theta)^\top(\nabla_\theta \log q(t, \theta) - \nabla_\theta \log p(\theta)) + \mathrm{tr}(\nabla_\theta^2 \log q(t, \theta) - \nabla_\theta^2 \log p(\theta)). \quad (3)$$

$q(t, \theta) = p(\theta)$ is a stationary point for this PDE, so LD has p as its stationary distribution.

We now consider the question: is there a *deterministic* flow $f_t(\theta)$ such that, if $q(t+\eta, \theta) = q(t, f_t^{-1}(\theta))|\frac{\partial f_t^{-1}}{\partial \theta}|$, then in the limit as $\eta \to 0$ we recover the dynamics for $\frac{\partial \log q}{\partial t}$ from equation 3? The answer is yes, and it turns out to be the ideal functional BBVI gradient step from equation 1 (Jordan et al., 1998; Villani, 2003):

$$f_t(\theta) = \theta + \eta\nabla_\theta \log p(\theta) - \eta\nabla_\theta \log q(t, \theta). \quad (4)$$

In principle, this gives us a *deterministic* way to reproduce the behavior of LD: sample from $q(0, \theta)$, and then recursively apply equation 4. This amounts to doing variational inference with a composition of normalizing flows: $q(t, \theta) = q(0, g_t^{-1}(\theta))|\frac{\partial g_t^{-1}}{\partial \theta}| g_t \triangleq f_t \circ f_{t-\eta} \circ \cdots \circ f_\eta \circ f_0$. Since the difference between g_t and $g_{t-\eta}$ is the functional gradient step from equation 1, g_t can be interpreted as the result of running t/η steps of BBVI on a nonparametric normalizing flow. So (to first order in η) LD can be interpreted as an implicit VI procedure where one runs k steps of nonparametric BBVI and then draws one sample from the result.

3. EXPERIMENTS

In this section we empirically evaluate BBVI and gradient-based MCMC to see how well the theoretical results of sections 2 agree with practice. More results are in the appendix.

We evaluate BBVI with vanilla SGD and with momentum 0.9, Metropolis-adjusted Langevin, and Hamiltonian Monte Carlo (HMC; Neal, 2011) with 10 leapfrog steps. For

[1] The $O(\eta^2)$ discretization error can be addressed by a Metropolis-Hastings (Hastings, 1970) correction, but this makes the analysis much more difficult so we ignore it here and focus on the continuous-time limit. Our empirical results using the Metropolis-adjusted algorithm are consistent with the intuitions from this continuous-time limit.

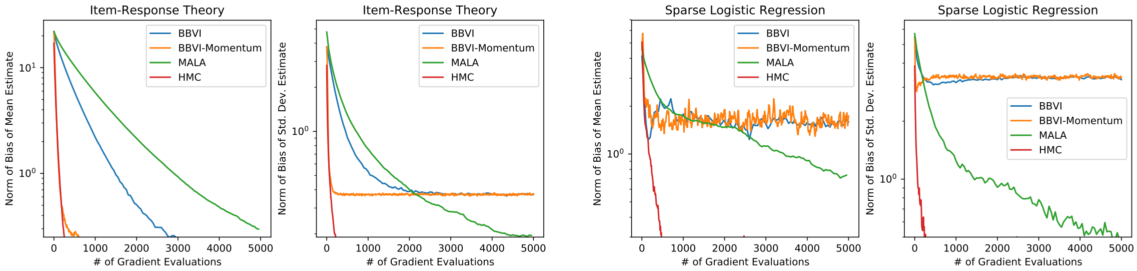

Figure 1: Bias of mean and standard deviation estimates obtained by BBVI (with and without momentum 0.9), MALA, and HMC with 10 leapfrog steps on an item-response theory model and a sparse logistic regression.

BBVI, we used a diagonal-covariance Gaussian variational family parameterized by the flow $\theta_d = \mu_d + 0.1 \log(1 + e^{10\sigma_d})\epsilon_d$. We estimated the gradients for BBVI using the sticking-the-landing estimator of Roeder et al. (2017) with a minibatch of 100 draws from q. Step sizes were tuned manually for each algorithm. We evaluate these methods on two Bayesian-data-analysis problems: an item-response theory model and a logistic regression with soft-sparsity priors (details in Section A).

Figure 1 shows the evolution of the bias of estimators based on BBVI and taking the last samples of a set of MCMC chains as a function of number of gradient evaluations (number of iterations for BBVI and MALA, number of iterations times number of leapfrog steps per iteration for HMC). For the IRT model, BBVI without momentum behaves similarly to MALA, but BBVI can use an effective step size about twice as large; for the sparse logistic regression, MALA is competitive with BBVI at each step. The accelerated algorithms (BBVI with momentum and HMC) behave almost identically early on, only diverging once BBVI's asymptotic bias kicks in. The results in figure 1 are broadly consistent with the claim that the implicit distribution governing the state of an unconverged MCMC chain has bias competitive with an explicit VI procedure run for the same amount of time.

Of course, bias is not the only source of error in MCMC; we must also consider variance. In Section C we consider total MCMC error for a single-chain and 100-chain workflow. When running only one chain, one must average samples from a long chain to get low enough variance to compete with BBVI. But running 100 chains eliminates most of this variance, yielding estimators that strictly dominate BBVI for any number of steps.

4. DISCUSSION

We showed that gradient-based MCMC and VI algorithms implicitly follow the same gradient flow, which causes them to exhibit similar transient behavior. This suggests that MCMC's main disadvantage over VI is not slow convergence, but high variance. This disadvantage evaporates when one can cheaply run many parallel MCMC chains, e.g., on modern commodity GPUs. As such parallel hardware gets cheaper, we predict that MCMC will become attractive relative to VI for more and more problems.

References

Elaine Angelino, Matthew James Johnson, Ryan P Adams, et al. Patterns of scalable Bayesian inference. *Foundations and Trends® in Machine Learning*, 9(2-3):119–247, 2016.

Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2019. URL http://archive.ics.uci.edu/ml.

Samuel Gershman and Noah Goodman. Amortized inference in probabilistic reasoning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 2014.

W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. doi: 10.1093/biomet/57.1.97. URL http://dx.doi.org/10.1093/biomet/57.1.97.

Matthew Hoffman, Pavel Sountsov, Joshua V. Dillon, Ian Langmore, Dustin Tran, and Srinivas Vasudevan. Neutra-lizing bad geometry in hamiltonian monte carlo using neural transport. *arXiv preprint arXiv:1903.03704*, 2019.

Martin Jankowiak and Fritz Obermeyer. Pathwise derivatives beyond the reparameterization trick. *arXiv preprint arXiv:1806.01851*, 2018.

Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker–planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17, 1998.

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

Radford M. Neal. MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo*. CRC Press New York, NY, 2011.

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

Danilo J 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*, pages 1278–1286, 2014.

D.J. Rezende and S. Mohamed. Variational Inference with Normalizing Flows. *ArXiv e-prints*, May 2015.

Gareth O. Roberts and Jeffrey S. Rosenthal. Optimal scaling of discrete approximations to langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1):255–268, 1998.

Geoffrey Roeder, Yuhuai Wu, and David K Duvenaud. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. In *Advances in Neural Information Processing Systems*, pages 6925–6934, 2017.

Cédric Villani. *Topics in optimal transportation*. Number 58. American Mathematical Soc., 2003.

Appendix A. Details of Target Distributions

Item-Response-Theory Model: This is the posterior of a one-parameter-logistic item-response-theory (IRT) model from the Stan (Carpenter et al., 2017) examples repository[2] with a total of 501 parameters:

$$\begin{aligned}&\delta \sim \mathcal{N}(0.75, 1); \quad \alpha_{1:400} \sim \mathcal{N}(0, 1); \quad \beta_{1:100} \sim \mathcal{N}(0, 1);\\ &y_i \sim \text{Bernoulli}(\sigma(\delta + \alpha_{s_i} - \beta_{r_i})),\end{aligned} \tag{5}$$

where $y_{i\in\{1,\dots,30105\}}$ indicates whether student s_i got question r_i correct.

Sparse Logistic Regression: This is the logistic regression model with soft-sparsity priors considered by Hoffman et al. (2019) applied to the German credit dataset (Dua and Graff, 2019):

$$\begin{aligned}&\beta_{1:D} \sim \mathcal{N}(0, 1); \quad \gamma_{0:D} \sim \text{Gamma}(0.5, 0.5);\\ &y_n \sim \text{Bernoulli}(\sigma(\gamma_0 \textstyle\sum_{d=1}^{D} x_{nd}\beta_d\gamma_d)).\end{aligned} \tag{6}$$

The $\gamma_{1:D}$ variables act as soft masks on the regression coefficients $\beta_{1:D}$; the Gamma$(0.5, 0.5)$ priors assign significant prior mass to settings of γ_d close to 0. We log-transform the γ variables to eliminate the nonnegativity constraint.

Appendix B. Synthetic Gaussian Experiment

We consider a simple ill-conditioned zero-mean synthetic 200-dimensional multivariate-Gaussian target distribution $p(\theta) = \mathcal{N}(\theta; 0, \Sigma)$. We set its covariance $\Sigma = U\Lambda U^\top$, where U is a random orthonormal matrix and Λ is a diagonal matrix with $\Lambda_{d,d} = 10^{3(d-1)/200}$, so that the eigenvalues of Σ vary over three orders of magnitude.

Clearly if our variational family q is multivariate Gaussian it can exactly match p. However, we still have to make choices about how to parameterize this family; in particular, to use the reparameterization trick we define q via an affine change of variables

$$\epsilon \sim \mathcal{N}(0, I); \quad \theta = g(\epsilon) = A(\phi)\epsilon. \tag{7}$$

There are tradeoffs for the scale matrix A. $A = \phi$ is simple, but it may lead to numerical issues if the eigenvalues of A cross 0, and the gradient of the ELBO involves explicitly forming A^{-1} at cost $O(D^3)$. One can instead use the matrix-logarithm parameterization $A = e^\phi$, which we will see achieves very fast convergence, but also requires $O(D^3)$ work per iteration. Finally, one can use a lower-triangular parameterization $A = \text{diag}(e_s^\phi) + \phi_L$, where ϕ_s is a D-dimensional vector and ϕ_L is a strictly lower-triangular matrix; computing forward gradients in this parameterization is cheap, since $|A| = \sum_d \phi_{s,d}$, and computing the log-density $\log q_\phi(\theta)$ for "sticking-the-landing" (STL) updates (Roeder et al., 2017) can be done with only $O(D^2)$ work using triangular solves, but the geometry of the tangent field $\frac{\partial g}{\partial \phi}$ may not be ideal (Jankowiak and Obermeyer, 2018).

We ran BBVI with vanilla stochastic gradient descent with the three parameterizations defined above ("linear-scale", "log-scale", and "lower-triangular" respectively), and

[2] https://github.com/stan-dev/example-models/blob/master/misc/irt/irt.stan

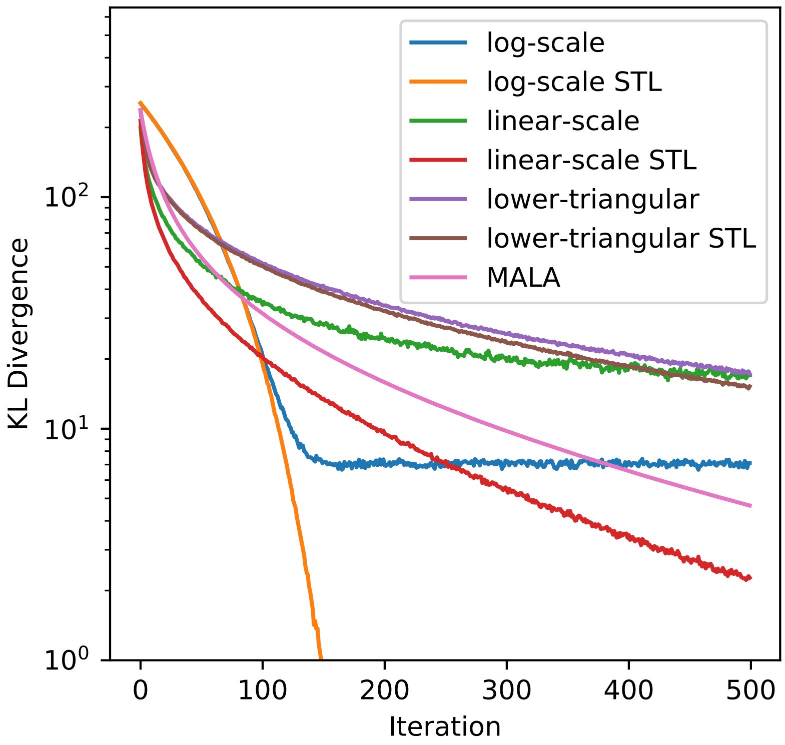

Figure 2: Kullback-Leibler (KL) divergence achieved by BBVI and MALA algorithms as a function of number of iterations. KL divergence is estimated for MALA assuming that the state of the chain at a given time is drawn from a multivariate Gaussian and estimating its parameters from $100,000$ chains.

compared the results with the Metropolis-adjusted Langevin algorithm (MALA). BBVI gradients were estimated using a minibatch of 100 samples from q. Each algorithm used a manually tuned constant step size.

Figure 2 shows the results. We find that, as theory predicts, BBVI with a constant step size does not converge, but BBVI with variance-reduced STL updates can achieve geometric convergence (since the gradient noise decays with the KL divergence). We also see that BBVI's performance depends strongly on parameterization; the lower-triangular parameterization is significantly slower than the linear-scale parameterization, while the log-scale parameterization (with STL updates) actually achieves superlinear convergence. MALA's performance is comparable to linear-scale STL BBVI, although MALA is a bit slower because it needs to use a smaller step size. Note that the BBVI parameterizations require some extra work per iteration compared to MALA; this work is $O(D^2)$ for the lower-triangular parameterization and $O(D^3)$ for the log-scale and linear-scale parameterizations.

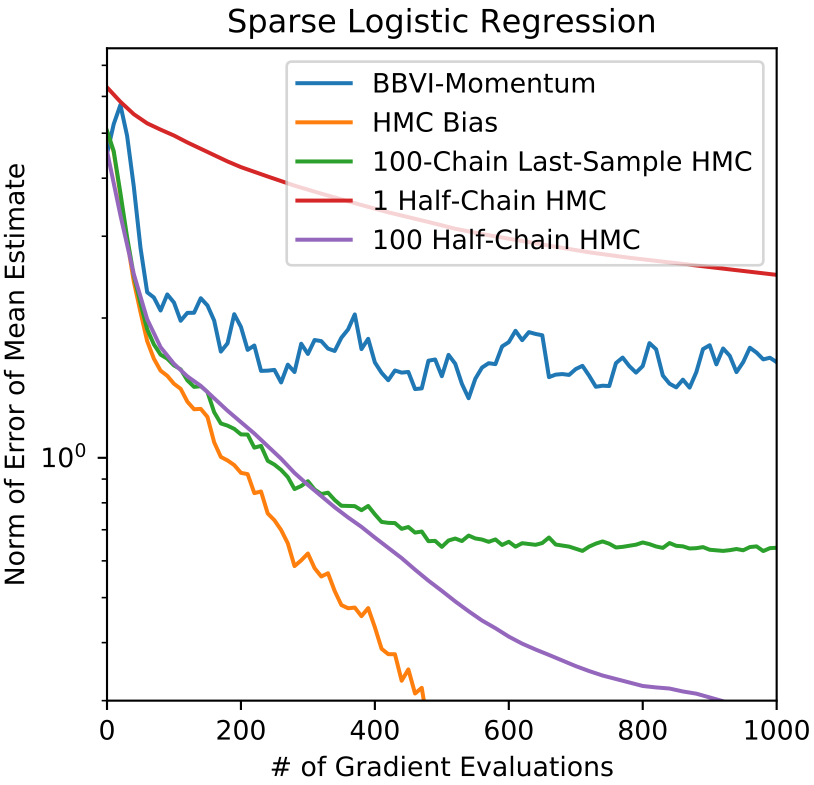

Figure 3: Total error of mean estimator for the sparse logistic regression as a function of number of gradient evaluations under various inference schemes. "M Half-chain HMC" refers to running M chains of HMC, discarding the first half of the samples as warmup, and averaging the remaining samples. "Last-Sample HMC" refers to running M chains of HMC and using only the final (least biased) sample.

Appendix C. Total Error of MCMC and BBVI

Figure 3 shows the total error of various MCMC estimator schemes for the sparse logistic regression problem. Running a single HMC chain and averaging samples from the last half of the chain quickly eliminates bias, but the error is still high due to variance. This may account for the conventional wisdom that VI is faster than MCMC—the single-chain HMC scheme would indeed require many iterations to average away enough variance to match BBVI's accuracy.

The situation is different if parallel computation is available. Averaging 100 independent chains brings the variance down to the point that total error is initially dominated by bias, which decays quickly. Either the traditional scheme of discarding the first halves of the 100 chains or the more radical approach of using only the last sample outperforms BBVI with momentum. Note that the BBVI scheme computes a minibatch of 100 gradients of the target density per step to let it take larger steps, so the comparison is fair. On an NVIDIA P100 GPU each of these batched gradient computations takes only a few milliseconds for the sparse logistic regression model.