Black-Box Variational Inference as Distilled Langevin Dynamics

Matthew Hoffman [1] **Yi-An Ma** [1 2]

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 the Markov chain not having burned in) may track that of VI (due to the optimizer not having converged), up to differences due to VI's asymptotic bias and parameterization. Empirically, we find that the transient biases of these algorithms (and their momentum-accelerated counterparts) 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, since parallel computation (especially on GPUs and TPUs) has become cheap but clock speeds have remained nearly constant.

In this paper, we will 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 & Welling, 2014; Rezende et al., 2014; Roeder et al., 2017) and Langevin-dynamics MCMC (LD; Roberts & Rosenthal, 1998). By reformulating LD as a deterministic normalizing flow (Rezende & 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 incomplete optimization rather than approximations) may track the transient bias of LD (i.e., bias due to the Markov chain not having warmed up), and that claims that VI is faster than MCMC demand closer scrutiny. 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 distribution's parameterization. We also analyze the effects of gradient noise in BBVI.

[1]Google Research [2]Halıcıoğlu Data Science Institute, University of California, San Diego, USA. Correspondence to: Matthew Hoffman <mhoffman@google.com>, Yi-An Ma <yianma@google.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

- We show empirically that the transient biases of BBVI and MCMC estimators often converge at similar speeds, even when BBVI uses very low-variance gradient estimators and can exactly match the target distribution. 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 & 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 continue to get cheaper, more and more one-off Bayesian data-analysis problems will be susceptible to this strategy.

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 view 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 & Mohamed, 2015). While this result is well known in the optimal-transport literature, and has more recently been used to prove non-asymptotic convergence rates for LD algorithms (Wibisono, 2018; Ma et al., 2019a), we are more concerned with its implications for parametric BBVI algorithms. We will show that gradient-based MCMC algorithms and parametric BBVI are following the same gradient signals (up to a tangent field due to the mapping from function space to parameter space), suggesting that BBVI's convergence behavior may track that of LD.

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 $p(\theta)$:

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

(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

$$\begin{aligned}&\frac{d}{d\phi} \int_\epsilon q_0(\epsilon)(\log q_\phi(g_\phi(\epsilon)) - \log p(g_\phi(\epsilon)))d\epsilon \\ &\quad = \int_\epsilon (\nabla_\theta \log q_g(g_\phi(\epsilon)) - \nabla_\theta \log p(g_\phi(\epsilon)))\frac{\partial g}{\partial \phi} d\epsilon.\end{aligned} \tag{2}$$

That is, the standard BBVI gradient step is the projection of the "ideal" BBVI functional gradient onto the parameter space of g_ϕ. We will show below that LD is implicitly following the ideal functional gradient in equation 1.

First, we need to review LD and its relation to the Fokker-Planck equation. In each iteration of LD, we update our state $\theta_n \in \mathbb{R}^D$ to

$$\theta_{n+1} = \theta_n + \eta \nabla \log p(\theta_n) + \sqrt{2\eta}\xi, \tag{3}$$

where η is a step size 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)

$$d\theta_t = \nabla \log p(\theta_t)dt + \sqrt{2}dW_t, \tag{4}$$

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

$$\partial \log q \overline{\partial t = \nabla_\theta \log q(t,\theta)^\top (\nabla_\theta \log q(t,\theta) - \nabla_\theta \log p(\theta)) \qquad + \mathrm{tr}(\nabla_\theta^2 \log q(t,\theta) - \nabla_\theta^2 \log}$$
(5)

$q(t, \theta) = p(\theta)$ is a stationary point for this PDE. Jordan et al. (1998) showed that q actually approaches p by a *steepest-descent* path, with "steepness" defined in terms of Wasserstein-2 distance and KL divergence between q and p; specifically, in the limit as $\eta \to 0$,

$$q(t + \eta, \cdot) = \arg\min_q \frac{1}{2}\mathcal{W}_2^2(q, q(t, \cdot)) + \eta \mathbb{E}_q \left[\log \frac{q(\theta)}{p(\theta)}\right].$$

That is, $q(t + \eta, \cdot)$ tries to minimize KL divergence as much as possible without moving too far in Wasserstein-2 distance.

[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.

Under some fairly mild assumptions on the smoothness, compactness, and boundedness of $q(0,\cdot)$ and p, we can write the squared Wasserstein-2 distance $\mathcal{W}_2^2(q(t,\cdot), q(t+\eta,\cdot))$ in terms of a transport map f_t

$$\mathcal{W}_2^2(q(t,\cdot), q(t+\eta,\cdot)) = \min_{f_t} \int_\theta q(t,\theta)||\theta - f_t(\theta)||^2. \\ s.t.\ q(t,\theta) = q(t+\eta, f_t(\theta))\left|\frac{\partial f_t}{\partial\theta}\right|.$$

The solution to this optimal-transport problem turns out to be the ideal functional BBVI gradient step from equation 1 (Villani, 2003, Chapter 8):

$$f_t(\theta) = \theta + \eta\nabla_\theta \log p(\theta) - \eta\nabla_\theta \log q(t,\theta). \tag{6}$$

Indeed, if we define $q(t+\eta,\cdot)$ by plugging f_t into the change-of-variables formula $q(t+\eta, f(\theta))|\frac{\partial f}{\partial\theta}| = q(t,\theta)$, it is not hard to verify that the result satisfies the Fokker-Planck equation (equation **??** above) to first order in η:

$$\begin{aligned}\log q(t+\eta,\theta) &= \log q(t, f_t^{-1}(\theta)) - \log\left|\frac{\partial f_t}{\partial\theta}\right| \\ &= \log q(t, \theta - \eta\nabla_\theta \frac{\log p(\theta)}{\log q(t,\theta)} + O(\eta^2)) \\ &\quad - \log\left|I + \eta\nabla_\theta^2 \frac{\log p(\theta)}{\log q(t,\theta)}\right| \\ &= \log q(t,\theta) \\ &\quad + \eta\nabla_\theta \log q(t,\theta)^\top \nabla_\theta \frac{\log q(t,\theta)}{\log p(\theta)} \\ &\quad + \eta \mathrm{tr}(\nabla_\theta^2 \frac{\log q(t,\theta)}{\log p(\theta)}) + O(\eta^2).\end{aligned} \tag{7}$$

In principle, this gives us a *deterministic* way to reproduce the behavior of LD: sample from $q(0,\theta)$, and then recursively apply equation 6[2]. This amounts to doing variational inference with a composition of normalizing flows:

$$\begin{aligned} q(t,\theta) &= q(0, g_t^{-1}(\theta))\left|\frac{\partial g_t^{-1}}{\partial\theta}\right| \\ g_t &\triangleq f_t \circ f_{t-\eta} \circ \cdots \circ f_\eta \circ f_0. \end{aligned} \tag{8}$$

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.

[2]This is not practical to do for more than a few iterations, since each iteration requires computing higher-order derivatives than the last one. An exception is if $q(0,\theta)$ and p are Gaussian, since then these derivatives vanish.

2.1. Momentum and Preconditioning

So far, we have only discussed versions of BBVI based on simple gradient descent. But practitioners often use optimization schemes such as Adam (Kingma & Ba, 2015), which can make faster progress than gradient descent by using momentum and gradient preconditioning. In this section, we find momentum-accelerated nonparametric BBVI to behave similarly to an underdamped Langevin diffusion in the space of probabilities. In section 5, we will confirm that momentum and preconditioning do indeed let both BBVI and MCMC algorithms reduce transient bias more quickly.

We begin by introducing momentum. A nonparametric BBVI-with-momentum scheme is

$$r_{k+1} = \alpha r_k + \eta\nabla_\theta \log\frac{p(\theta_k)}{q(\theta_k)}; \quad \theta_{k+1} = \theta_k + r_{k+1}, \tag{9}$$

where r is an auxiliary momentum vector with the same shape as θ, η is a step size, and α is a smoothing parameter between 0 and 1. It essentially collects an exponentially weighted moving average of functional gradients. The auxiliary vector r can also be seen as a displacement map from $q(\theta_k)$ to $q(\theta_{k+1})$, sampled at θ_k. This notion establishes the momentum BBVI method as an acceleration scheme for the Langevin dynamics. To see this, we set $v = r/\sqrt{\eta}$ and $\hat\alpha = (1-\alpha)/\sqrt{\eta}$ and examine the continuous dynamics associated with the above momentum BBVI scheme:

$$\frac{d\theta_t}{dt} = v_t; \quad \frac{dv_t}{dt} = -\hat\alpha v_t + \nabla_\theta \log\frac{p(\theta_t)}{q(\theta_t)}. \tag{10}$$

Formally, we define $T_t : \mathbb{R}^d \to \mathbb{R}^d$ as the optimal transport plan from q to p at time t. Then for every θ_t, its instantaneous velocity is $\partial_t((T_t)^{-1})T_t(\theta_t)$. Using this notion, we can represent the entire velocity field $v_t : \mathbb{R}^d \to \mathbb{R}^d$ as $v_t(\cdot) = \partial_t((T_t)^{-1})T_t(\cdot)$ (see, e.g., Y. Wang, 2019). We thereby obtain the expanded vector flow from the gradient flow of $\nabla_\theta \log\frac{q}{p}$:

$$\begin{pmatrix} v_t \\ -\hat\alpha v_t - \nabla_\theta \log\frac{q}{p}\end{pmatrix} = -\begin{pmatrix} 0 & -I \\ I & \hat\alpha I\end{pmatrix}\begin{pmatrix}\nabla_\theta \log\frac{q}{p} \\ v_t\end{pmatrix},$$

which corresponds to the dynamics of accelerated gradient descent (Su et al., 2014; Wilson et al., 2016).

Another approach to accelerated gradient descent in the space of probability is via introducing a joint distribution over $z = (\theta, v)$. Letting the joint target distribution $p(z) \propto p(\theta)\exp\left(-\frac{1}{2}\|v\|_2^2\right)$, we can extend the gradient flow as:

$$dz_t = -\begin{pmatrix} 0 & -I \\ I & \hat\alpha I\end{pmatrix}\nabla_z \log\frac{\hat q(z_t)}{\hat p(z_t)}dt. \tag{11}$$

The above dynamics correspond to the underdamped Langevin diffusion, which leads to many acceleration schemes in MCMC (Cheng et al., 2018; Mangoubi & Smith, 2017; Mangoubi & Vishnoi, 2018; Lee et al., 2018; Dalalyan & Riou-Durand, 2018; Ma et al., 2019a; Shen & Lee, 2019).

For both forms of continuous dynamics (in equations 10 and 11), an $\mathcal{O}(\sqrt{m})$ rate of convergence can be achieved for m-strongly log-concave target distributions $p(\theta)$ (Y. Wang, 2019; Y. Cao, 2020), which corresponds to accelerated gradient descent in finite dimensions (Su et al., 2014; Wilson et al., 2016).

In summary, VI leverages momentum to achieve similar acceleration behavior as underdamped Langevin MCMC, but in a more deterministic fashion due to its use of a finite-dimensional parameterization over the space of probabilities. It would be interesting to see how other variations of Langevin dynamics (see, e.g., Tzen & Raginsky, 2019; Mou et al., 2019; Ma et al., 2018) correspond and contribute to the optimization processes of VI.

3. Implications

The equivalence between BBVI and LD has implications for how the bias of BBVI and LD estimators evolves as a function of time. Estimators based on BBVI with a nonparametric normalizing flow should have roughly the same bias as LD estimators if both are run with the same step size and number of steps[3].

One reason that the bias of BBVI with a parametric flow g_ϕ may differ from that of LD is due to the influence of the tangent field $\frac{\partial g}{\partial \phi}$ from equation 2 that translates from function space to parameter space. This is clearly a point against BBVI in the "non-realizable" setting where there does not exist a ϕ such that $q_\phi(\theta) = p(\theta)$. But even in the realizable setting, the tangent field $\frac{\partial g}{\partial \phi}$ distorts the geometry of the gradient flow $\nabla_\theta \log \frac{p}{q}$. This distortion will often be harmful, insofar as normalizing flows are usually designed without this sort of geometric consideration in mind. However, there are cases where it can be helpful; we will give an example in section 5.1.

Gradient noise can also affect the transient bias of parametric BBVI. Approximating the gradient in equation 2 using a small number of samples from q will divert the evolution of the variational distribution from its ideal gradient flow, but this issue can be mitigated using variance reduction (e.g., Roeder et al., 2017) or by averaging gradient signals from many samples computed in parallel. This is discussed in more detail in section 4.

At this point one might ask: if LD and BBVI are so similar, why do practitioners find that BBVI often gives useful results faster than LD? We claim that the answer has to do with the *variance* of LD estimators. Once a parametric variational distribution q_ϕ has been fit, one can usually draw many samples from q_ϕ cheaply to accurately estimate expectations $\mathbb{E}_q[h(\theta)]$; insofar as these estimates diverge from the true expectations $\mathbb{E}_p[h(\theta)]$, it is mostly due to bias due to $q_\phi \neq p$. By contrast, if one compares BBVI with single-sample gradient estimators to a single chain of LD (so that both methods do roughly the same computational work per step), then the LD estimator will have error due to both transient bias (if the chain is not run to convergence) and variance (since the estimator will be based on a small number of samples). If LD and BBVI are run for a relatively short time so that LD and BBVI have similar transient bias (and this transient bias dominates BBVI's asymptotic bias), then LD's higher variance will lead to less-accurate estimates than BBVI. On the other hand, if it is cheap to reduce this variance by running many LD chains in parallel, LD's lack of asymptotic bias and ability to follow the undistorted nonparametric gradient flow $\nabla_\theta \log \frac{p}{q}$ may yield more-accurate estimates for any wall-clock time budget.

How many parallel MCMC chains do we need to run to reduce variance to an acceptable level? One way to answer this question is to suppose we are doing posterior inference on a Bayesian model, and that the dataset we are analyzing was drawn from that model's prior-predictive distribution. We want to estimate some quantity ϕ whose variance under the posterior is v. If we can draw M independent samples from the posterior, the expected squared error $\mathbb{E}[(\hat{\phi} - \phi^\star)^2]$ from our estimator $\hat{\phi}$ to the true value $\phi^\star$ is $v(1 + \frac{1}{M})$—the Monte Carlo error $\frac{v}{M}$ plus the posterior variance v (which can only be reduced by gathering more data). So the scale of the error of $\hat{\phi}$ will be $\sqrt{v(1 + \frac{1}{M})} \approx \sqrt{v}(1 + \frac{1}{2M})$. So if we run 50 parallel MCMC chains to near-convergence and take the last sample, we can expect our estimators to have about one percent higher error than they would given infinite computation. A single GPU can quickly run 50 parallel chains for many Bayesian inference problems, and as cheap GPUs get more powerful, more and more problems will be amenable to this sort of "last-sample" workflow.

4. Convergence of VI and MCMC

The goal of this section is to explore the convergence speed of VI with gradient descent to attain a local approximation and how it compares to the convergence of an MCMC method. We consider the simple problem of approximating a centered normal distribution: $p(\theta) \propto \exp\left(-\frac{1}{2}\theta^\top \Lambda^* \theta\right)$, where the precision matrix Λ^* is non-singular.

Two scenarios are considered here. One is the "open-box" scenario where the algorithm knows that the posterior is a centered normal distribution and performs gradient descent to converge to it. In this oversimplified scenario, we prove that the posterior is captured exponentially fast. Another

[3] Although in practice BBVI may sometimes allow for slightly larger step sizes than LD; see section 5.

scenario is the "black-box" case where the algorithm is only given queries to the posterior values and its gradient information. Stochastic approximation must be made with samples from the approximating distribution, $q(\theta|\cdot)$. In this case, we prove that the convergence is much slower. To approximate the posterior in $\mathbb{R}^d$ up to ϵ accuracy, $\Omega(d/\epsilon)$ iterations as well as $\Omega(d)$ samples from $q(\theta|\cdot)$ in each iteration are required, which is comparable to the computational complexity of the Langevin algorithm provided in section 4.4.

Before delving into further details, we quickly note that both VI and MCMC can make use of acceleration techniques discussed in section 2.1 to improve their convergence speed. This point is further manifested in the experiment section.

We choose our variational approximation family to be $q(\theta|\Lambda) \propto \exp\left(-\frac{1}{2}\theta^\top \Lambda\theta\right)$, parameterized by the precision matrix Λ. Our goal is to perform gradient descent over the space of Λ and converge to Λ^*. In VI practice, one chooses $\mathrm{KL}\left(q(\theta|\Lambda)\|p(\theta)\right)$ to minimize:

$$\mathrm{KL}\left(q(\theta|\Lambda)\|p(\theta)\right) = \mathbb{E}_{q(\theta|\Lambda)} \log \frac{q(\theta|\Lambda)}{p(\theta)}.$$

In a gradient descent step, one plugs the variational approximation $\log q(\theta|\Lambda) = -\frac{1}{2}\theta^\top\Lambda\theta + \frac{1}{2}\log|\Lambda| + C$ into the gradient of $\mathrm{KL}\left(q(\theta|\Lambda)\|p(\theta)\right)$ and obtains

$$\begin{aligned}&\nabla_\Lambda \mathrm{KL}\left(q(\theta|\Lambda)\|p(\theta)\right)\\ &= \mathbb{E}_{q(\theta|\Lambda)}\left[\nabla_\Lambda \log q(\theta|\Lambda)\log\frac{q(\theta|\Lambda)}{p(\theta)}\right]\\ &= \frac{1}{2}\mathbb{E}_{q(\theta|\Lambda)}\left[\left(\Lambda^{-1} - \theta\theta^\top\right)\log\frac{q(\theta|\Lambda)}{p(\theta)}\right], \end{aligned} \tag{12}$$

where $\mathbb{E}_{q(\theta|\Lambda)}\left[\theta\theta^\top\right] = \Lambda^{-1}$. We discuss in the rest of this section the "open-box" and "black-box" approaches to perform (preconditioned) gradient descent with equation 12.

4.1. Convergence of "Open-Box" VI

In the "open-box" case, the algorithm knows the form of the log-posterior, $\log p(\theta) = -\frac{1}{2}\theta^\top\Lambda^*\theta + \frac{1}{2}\log|\Lambda^*| + C$, and can compute the expectation in Eq. 12 exactly:

$$\nabla_\Lambda \mathrm{KL}\left(q(\theta|\Lambda)\|p(\theta)\right) = \frac{1}{2}\left(\Lambda^{-1} - \Lambda^{-1}\Lambda^*\Lambda^{-1}\right).$$

After preconditioning $\nabla_\Lambda \mathrm{KL}\left(q(\theta|\Lambda)\|p(\theta)\right)$ with $\Lambda\otimes\Lambda$, we obtained an update rule of preconditioned gradient descent:

$$\Lambda_n = \Lambda_{n-1} - h_{n-1}\, g(\Lambda_{n-1}). \tag{13}$$

We prove in the following theorem that the above update enjoys an exponential convergence guarantee of Λ_n to Λ^*.

Lemma 1. *For the update rule described in equation 13 with exact gradients, if we take a step size of $h = \frac{1}{2}$, we can obtain that when $n \geq \log_2 \frac{2}{\sigma_{\min}(\Lambda^*)}\frac{\|\Lambda_0 - \Lambda^*\|_F}{\epsilon}$, $\mathrm{KL}\left(p(\theta)\|q(\theta|\Lambda_n)\right) \leq \epsilon$, for any $\epsilon \leq 1$.*

4.2. Convergence of "Black-Box" VI

In practice, however, one often cannot compute the integral in equation 12 explicitly and instead turns to a (sample-based) stochastic estimate of this gradient (Hoffman et al., 2013; Ranganath et al., 2014). In this case, we examine the convergence of the (preconditioned) gradient descent with a relatively small amount of stochastic gradient noise as well as the cost to obtain the stochastic approximation with the required fidelity. Let's first assume that we can obtain a stochastic approximation $\hat{g}(\Lambda)$ to the preconditioned gradient $g(\Lambda) = \Lambda\nabla_\Lambda \mathrm{KL}\left(q(\theta|\Lambda)\|p(\theta)\right)\Lambda$, so that their difference $\Delta(\Lambda;\mathcal{D}_n) = \hat{g}(\Lambda;\mathcal{D}_n) - g(\Lambda)$ is unbiased and bounded in expectation for any n (see Sec. 4.3 for the origin of these constants):

$$\mathbb{E}\left[\Delta(\Lambda;\mathcal{D}_n)\right] = 0; \quad \mathbb{E}\left\|\Delta(\Lambda;\mathcal{D}_n)\right\|_F^2 \leq \sigma_{\max}^2(\Lambda^*) d\delta^2. \tag{14}$$

Under these assumptions, we can similarly implement the stochastic preconditioned gradient descent as

$$\Lambda_n = \Lambda_{n-1} - h_{n-1}\,\hat{g}(\Lambda_{n-1}). \tag{15}$$

Theorem 1. *Consider running the stochastic preconditioned gradient descent algorithm described in equation 15 for a target posterior $p(\theta) \propto \exp\left(-\frac{1}{2}\theta^\top\Lambda^*\theta\right)$. Under assumptions 14 on the stochastic gradient noise, BBVI converges exponentially up to the level of stochastic gradient noise:*

$$\begin{aligned}\mathbb{E}\|\Lambda_n - \Lambda^*\|_F^2 \leq \left(1 - \frac{h_i}{2}\right)^n \mathbb{E}\left\|\Lambda_0 - \Lambda^*\right\|_F^2\\ + 2h\sigma_{\max}^2(\Lambda^*) d\delta^2.\end{aligned}$$

If we take a step size of $h = \mathcal{O}\left(\frac{\sigma_{\min}(\Lambda^*)}{\sigma_{\max}^2(\Lambda^*)\delta^2}\cdot\frac{\epsilon}{d}\right)$ for any $\epsilon \leq 1$, we can obtain that $\mathrm{KL}\left(p(\theta)\|q(\theta|\Lambda_n)\right) \leq \epsilon$ with a high probability, when $n \geq \widetilde{\mathcal{O}}\left(\frac{\sigma_{\max}^2(\Lambda^*)\delta^2}{\sigma_{\min}^2(\Lambda^*)}\frac{d}{\epsilon}\right)$. We also prove that this bound is tight.

This dichotomy between the "open-box" and the "black-box" regimes of exponential versus linear convergence is caused by the stochastic gradient noise. Via scrutinizing the convergence behavior, we can observe that up to the accuracy level of $\epsilon = \mathcal{O}(d\delta^2)$, the convergence of BBVI is still exponential. It becomes linear when the step size must be small enough to contain the effect of stochastic gradient noise. If one can approximate the gradient with higher precision, so that $d\delta^2$ scales less than or equal to the accuracy requirement ϵ, then the more appealing fast convergence scenario of Lemma 1 can be achieved in the "black-box" setting. We explore this possibility in section 5 where the variance-reduced sticking-the-landing (Roeder et al., 2017) update significantly increases convergence speed.

4.3. Sample Complexity of "Black-Box" VI

Define $v(\Lambda;\theta) = \Lambda\nabla_\Lambda \log q(\theta|\Lambda)\Lambda \cdot \log\frac{q(\theta|\Lambda)}{p(\theta)}$. Then $\hat{g}(\Lambda;\mathcal{D}_n) = \frac{1}{|\mathcal{D}_n|}\sum_{\theta_i\in\mathcal{D}_n} v(\Lambda;\theta_i)$ and $g(\Lambda) = \mathbb{E}_{\theta\sim q}[v(\Lambda;\theta)]$. We first use the above definitions to develop the expression of

$$\begin{aligned}&\mathbb{E}\|\Delta(\Lambda;\mathcal{D}_n)\|_F^2\\ &= \frac{1}{|\mathcal{D}_n|}\mathbb{E}_{\theta\sim q}\left[\left\|v(\Lambda;\theta) - \mathbb{E}_{\hat{\theta}\sim q}\left[v\left(\Lambda;\hat{\theta}\right)\right]\right\|_F^2\right],\end{aligned}$$

where $\mathbb{E}_{\hat{\theta}\sim q}\left[v\left(\Lambda;\hat{\theta}\right)\right] = \frac{1}{2}(\Lambda - \Lambda^*)$.

It can be further calculated that:

$$\begin{aligned}&\mathbb{E}_{\theta\sim q}\left[\left\|v(\Lambda;\theta) - \mathbb{E}_{\hat{\theta}\sim q}\left[v\left(\Lambda;\hat{\theta}\right)\right]\right\|_F^2\right]\\ &= \frac{1}{2}\left(\mathrm{tr}(\Lambda-\Lambda^*)\right)^2 - \frac{1}{4}\|\Lambda-\Lambda^*\|_F^2\\ &+ \left(\frac{1}{8}\left\|I - \Lambda^*\Lambda^{-1}\right\|_F^2 + \frac{1}{16}\left(\mathrm{KL}\left(q(\theta|\Lambda)\|p(\theta)\right)\right)^2\right)\\ &\cdot\left((\mathrm{tr}(\Lambda))^2 + \mathrm{tr}\left(\Lambda^2\right)\right).\end{aligned}$$

If one initializes Λ_0 sufficiently close to Λ^*, so that $\|\Lambda_0 - \Lambda^*\|_F^2$ and $\mathrm{KL}(q(\theta|\Lambda_0)\|p(\theta))$ are upper bounded by absolute constants, then

$$\begin{aligned}&\mathbb{E}_{\theta\sim q}\left[\left\|v(\Lambda;\theta) - \mathbb{E}_{\hat{\theta}\sim q}\left[v\left(\Lambda;\hat{\theta}\right)\right]\right\|_F^2\right]\\ &= \mathcal{O}\left(\frac{1}{\sigma_{\min}(\Lambda^*)}d^2\right).\end{aligned}$$

Hence to obtain that $\mathbb{E}\|\Delta(\Lambda;\mathcal{D}_n)\|_F^2 \le \sigma_{\max}^2(\Lambda^*)d\delta^2$, we need $\mathcal{O}\left(\frac{1}{\sigma_{\min}(\Lambda^*)}\frac{d}{\delta^2}\right)$ number of samples in $\mathcal{D}_n$.

4.4. Convergence of Langevin Algorithms

It has been demonstrated that for a posterior distribution $p(\theta) \propto \exp(-U(\theta))$ with strongly convex and Lipschitz smooth potential, $U(\theta)$, the unadjusted Langevin algorithm (ULA) converges within $\mathcal{O}(d/\epsilon)$ number of steps and the Metropolis adjusted Langevin algorithm (MALA) converges within $\mathcal{O}(d\log 1/\epsilon)$ iterations when initialized close to the posterior (Dalalyan, 2017; Durmus & Moulines, 2019; Dalalyan & Karagulyan, 2017; Cheng & Bartlett, 2018; Dwivedi et al., 2018; Ma et al., 2019b).

We formally state the result of ULA below and provide a simple proof in the appendix. To quantify convergence, we follow the same standard as in the VI setting and use the KL divergence between the distribution $q_n(\theta)$ of the current iterate θ_n and the posterior $p(\theta)$.

Proposition 1 ((Dalalyan, 2017; Durmus & Moulines, 2019; Cheng & Bartlett, 2018; Ma et al., 2019b)). *Assume that the target distribution $p(\theta) \propto \exp(-U(\theta))$ with m-strongly convex and L-Lipschitz smooth potential, $U(\theta)$. For the unadjusted Langevin algorithm described in equation 3 with step size η, it converges exponentially up to the level of discretization error:*

$$\mathrm{KL}(q_n\|p) \le e^{-m\eta n}\mathrm{KL}(q_0\|p) + \left(2\frac{L^4}{m^2}\eta^2 + \frac{L^2}{m}\eta\right)d.$$

If we take a step size of $\eta = \mathcal{O}(\frac{m}{L^2}\frac{\epsilon}{d})$, we can obtain that $\mathrm{KL}(q_n\|p) \le \epsilon$ when $n \ge \widetilde{\mathcal{O}}\left(\frac{L^2}{m^2}\frac{d}{\epsilon}\right)$.

Note that ULA has exactly the same behavior as vanilla BBVI. One can substitute $\{L, m\}$ for $\{\sigma_{\max}(\Lambda^*), \sigma_{\min}(\Lambda^*)\}$ and observe this correspondence in the convergence rate in Theorem 1. More importantly, ULA also exhibits the exponential-towards-linear transition in its convergence behavior: when $\epsilon = \mathcal{O}(d)$, ULA's convergence is exponential. It becomes linear when the step size must be small enough to control the stochasticity, which has $\mathcal{O}(d)$ variance.

From the convergence results of both VI and MCMC, we can observe that they are both performing approximate gradient descent via simulating continuous paths on the space of probabilities. A crucial factor in this approach is how large the step size can be taken. Without any stochasticity, VI can achieve exponential convergence rate via efficient parametrization of simple target distributions. With stochasticity, however, vanilla BBVI is subject to stochastic gradient noise comparable to the injected noise of MCMC. Hence vanilla BBVI does not enjoy any additional benefit from the finite-dimensional parametrization.

4.5. Sample Complexity of Monte Carlo Estimation

As discussed earlier, the parallelization of BBVI applies directly to MCMC: one can run a number of parallel MCMC chains to obtain mean estimates as soon as the Langevin algorithm converges. We demonstrate here that the number of chains required scales less than the number of parallel machines needed for BBVI in the (sample-based) stochastic gradient estimate.

Assume we want to estimate the mean of any L_f-Lipschitz function $f:\mathbb{R}\to\mathbb{R}$ via i.i.d. samples $\{\Theta_i\cdots\Theta_K\}$ obtained from running K parallel chains following the ULA. By the Herbst argument (see, e.g., Ledoux, 1999), we know that any random variable Θ with m-strongly convex potential is a sub-Gaussian random variable with parameter $\frac{1}{\sqrt{m}}$. We can further combine it with the Prékopa-Leindler inequality to show that any L_f-Lipschitz function of Θ is $\frac{2L_f}{\sqrt{m}}$-sub-Gaussian (see, e.g., Theorem 3.16 in Wainwright, 2019). This leads to the Hoeffding concentration bound:

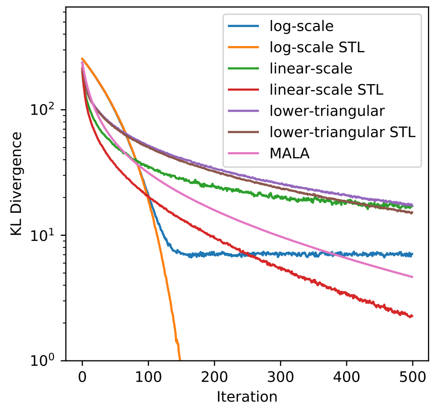

Figure 1. Kullback-Leibler (KL) divergence achieved by BBVI and MALA algorithms as a function of number of iterations. KL divergence is estimated for MALA assuming that the state of the chain at a given time is drawn from a multivariate Gaussian and estimating its parameters from 100,000 chains.

$$\Pr\left(\left|\frac{1}{K}\sum_{k=1}^{K} f(\Theta_k) - \mathbb{E}f(\Theta)\right| \geq t\right) \leq 2\exp\left(-\frac{Kmt^2}{4L_f^2}\right).$$

Therefore, the number of samples required for a mean estimate of f up to accuracy ϵ with probability $(1-\varepsilon)$ is $K = \frac{4}{m}\frac{L_f^2}{\epsilon^2}\log\left(\frac{2}{\varepsilon}\right)$. For the normalized accuracy $\delta = \epsilon/L_f$, this implies a sample complexity of $\mathcal{O}\left(\frac{1}{m}\frac{1}{\delta^2}\right)$. Comparing to the BBVI sample complexity in Sec. 4.3, we see that BBVI requires an extra dimension factor of samples.

5. Experiments

In this section we empirically evaluate various flavors of BBVI and gradient-based MCMC to see how well the theoretical results of sections 2 and 4.3 agree with practice. We begin by considering the *realizable* setting, where the true target distribution p can be exactly matched by a parametric distribution q_ϕ for some parameter vector ϕ; this eliminates BBVI's asymptotic bias and lets us focus on short-term convergence behavior. We then consider real-world data analysis problems where q_ϕ cannot be made to exactly match the target posterior $p(\theta \mid x)$. All experiments were done using TensorFlow Probability (Dillon et al., 2017; Lao et al., 2020).

5.1. Synthetic Gaussian

We begin with 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.

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

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 $A = \mathrm{diag}(e_s^\phi) + \phi_L$, 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 & 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 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 1 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.

5.2. The Unrealizable Setting

The synthetic experiments from section 5.1 suggest that MCMC and BBVI can converge at similar rates in the realizable setting, where the target distribution is in the variational family. In this section, we examine more realistic

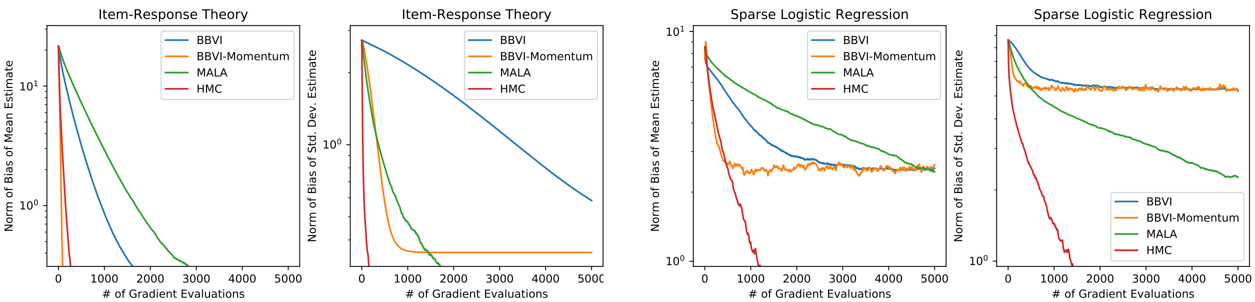

Figure 2. Bias of mean and standard deviation estimates obtained by BBVI (with and without momentum 0.9), MALA, and HMC with 10 leapfrog steps on an item-response theory model and a sparse logistic regression.

data-analysis problems where the target distribution is an intractable posterior distribution.

We evaluate BBVI with vanilla SGD and with momentum 0.9, Metropolis-adjusted Langevin, and Hamiltonian Monte Carlo with 10 leapfrog steps. For BBVI, we used a diagonal-covariance Gaussian variational family parameterized by the flow $\theta_d = \mu_d + 0.1\log(1+e^{10\sigma_d})\epsilon_d$; for positive values of σ_d, this approximates $\theta_d \sim \mathcal{N}(\mu_d, \sigma_d)$, but the scaled-softplus transformation lets us optimize σ without worrying about nonnegativity constraints. 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.

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[4] with a total of 501 parameters:

$$\delta \sim \mathcal{N}(0.75, 1); \quad \alpha_{1:400} \sim \mathcal{N}(0,1); \quad \beta_{1:100} \sim \mathcal{N}(0,1);$$
$$y_i \sim \mathrm{Bernoulli}(\sigma(\delta + \alpha_{s_i} - \beta_{r_i})),$$

where $y_{i\in\{1,\ldots,30105\}}$ 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 & Graff, 2019):

$$\beta_{1:D} \sim \mathcal{N}(0,1); \quad \gamma_{0:D} \sim \mathrm{Gamma}(0.5, 0.5);$$
$$y_n \sim \mathrm{Bernoulli}(\sigma(\gamma_0 \sum_{d=1}^{D} x_{nd}\beta_d\gamma_d)). \tag{16}$$

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

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

To estimate the ground-truth means and standard deviations for each model, we ran 500 HMC chains of 1000 iterations each, discarding the first 500 samples of each chain. We then compared these estimates to the estimates from the BBVI and MCMC methods. To estimate the expectations over time of the MCMC algorithms, we ran 500 independent chains and estimated the mean and standard deviations across chains after each update.

Figure 2 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). As in section 5.1, when estimating means BBVI without momentum behaves similarly to MALA, but BBVI can use a larger effective step size. The accelerated algorithms (BBVI with momentum and HMC) behave almost identically early on, only diverging once BBVI's asymptotic bias kicks in. When estimating standard deviations, MALA dominates BBVI, and HMC dominates BBVI with momentum. The results in figure 2 are 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. 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

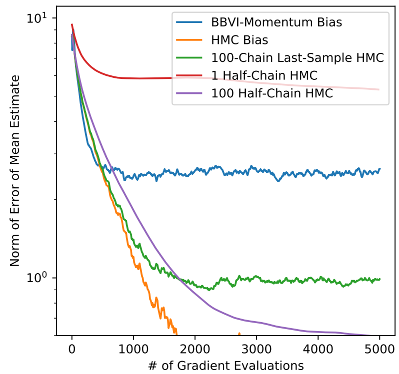

Figure 3. Total error of mean estimator for the sparse logistic regression as a function of number of gradient evaluations under various inference schemes. "M Half-chain HMC" refers to running M chains of HMC, discarding the first half of the samples as warmup, and averaging the remaining samples. "Last-Sample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

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 (which reduces the variance of its gradient estimates, and thereby lets it take larger steps), so the comparison is fair—the wallclock time per gradient evaluation of the BBVI algorithm and the 100-chain MCMC algorithms is nearly identical on a Pascal Titan X GPU.

6. Discussion

We have seen that gradient-based MCMC and VI algorithms implicitly follow the same gradient flow, and that this causes them to exhibit similar transient behavior. This suggests that VI's main advantage over MCMC is its ability to provide low-variance estimates, rather than an ability to converge to its asymptotic bias quickly. This advantage 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.

One question is how these results apply to highly non-convex and multimodal potentials like those considered by Ma et al. (2019b). One might expect VI to track gradient-based MCMC methods, since both perform first-order optimization on the space of probability. But some parametric variational families may handle multimodality better; for example, a variational mixture distribution (Jaakkola & Jordan, 1998; Graves, 2016) might more accurately weight separated modes than many independent MALA chains would. Analyzing these parametric families as ways of projecting Fokker-Planck gradient flows might inspire new MCMC or VI procedures.

Acknowledgements

We thank Boris Alexeev, Marco Cuturi, Josh Dillon, Katherine Heller, Ghassen Jerfel, Dave Moore, Brian Patton, Dan Piponi, Alexey Radul, Rif A. Saurous, Pavel Sountsov, Chris Suter, Srinivas Vasudevan, and Sharad Vikram for helpful and enjoyable discussions and feedback, as well as the anonymous reviewers for their helpful suggestions.

References

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

Bakry, D. and Emery, M. Diffusions hypercontractives. In Azema, J. and Yor, M. (eds.), *Séminaire de Probabilités XIX 1983/84*, pp. 177–206. Springer, 1985.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017.

Cheng, X. and Bartlett, P. L. Convergence of Langevin MCMC in KL-divergence. In *Proceedings of the 29th International Conference on Algorithmic Learning Theory (ALT)*, pp. 186–211, 2018.

Cheng, X., Chatterji, N. S., Bartlett, P. L., and Jordan, M. I. Underdamped Langevin MCMC: A non-asymptotic analysis. In *Proceedings of the 31st Conference on Learning Theory (COLT)*, pp. 300–323, 2018.

Dalalyan, A. S. Theoretical guarantees for approximate sampling from smooth and log-concave densities. *J. Royal Stat. Soc. B*, 79(3):651–676, 2017.

Dalalyan, A. S. and Karagulyan, A. G. User-friendly guarantees for the Langevin Monte Carlo with inaccurate gradient. arXiv:1710.00095, 2017.

Dalalyan, A. S. and Riou-Durand, L. On sampling from a log-concave density using kinetic Langevin diffusions. arXiv:1807.09382, 2018.

Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M. D., and Saurous, R. A. TensorFlow distributions. November 2017.

Dua, D. and Graff, C. UCI machine learning repository, 2019. URL `http://archive.ics.uci.edu/ml`.

Durmus, A. and Moulines, E. High-dimensional Bayesian inference via the unadjusted Langevin algorithm. *Bernoulli*, 25(4A):2854–2882, 2019.

Dwivedi, R., Chen, Y., Wainwright, M. J., and Yu, B. Log-concave sampling: Metropolis-Hastings algorithms are fast! arXiv:1801.02309, 2018.

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

Graves, A. Stochastic backpropagation through mixture density distributions. July 2016.

Hastings, W. K. 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`.

Hoffman, M., Sountsov, P., Dillon, J. V., Langmore, I., Tran, D., and Vasudevan, S. Neutra-lizing bad geometry in hamiltonian monte carlo using neural transport. *arXiv preprint arXiv:1903.03704*, 2019.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *Journal of Machine Learning Research*, 2013.

Jaakkola, T. S. and Jordan, M. I. Improving the mean field approximation via the use of mixture distributions. In Jordan, M. I. (ed.), *Learning in Graphical Models*, pp. 163–173. Springer Netherlands, Dordrecht, 1998.

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

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

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

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

Lao, J., Suter, C., Langmore, I., Chimisov, C., Saxena, A., Sountsov, P., Moore, D., Saurous, R. A., Hoffman, M. D., and Dillon, J. V. tfp.mcmc: Modern markov chain monte carlo tools built for modern hardware. February 2020.

Ledoux, M. Concentration of measure and logarithmic Sobolev inequalities. In *Seminaire de probabilites XXXIII*, pp. 120–216. 1999.

Lee, Y.-T., Song, Z., and Vempala, S. S. Algorithmic theory of ODEs and sampling from well-conditioned logconcave densities. arXiv:1812.06243, 2018.

Ma, Y.-A., Fox, E. B., Chen, T., and Wu, L. Irreversible samplers from jump and continuous Markov processes. *Stat. Comput.*, pp. 1–26, 2018.

Ma, Y.-A., Chatterji, N. S., Cheng, X., Flammarion, N., Bartlett, P. L., and Jordan, M. I. Is there an analog of Nesterov acceleration for MCMC? arXiv:1902.00996, 2019a.

Ma, Y.-A., Chen, Y., Jin, C., Flammarion, N., and Jordan, M. I. Sampling can be faster than optimization. *Proc. Natl. Acad. Sci. U.S.A.*, 2019b.

Mangoubi, O. and Smith, A. Rapid mixing of Hamiltonian Monte Carlo on strongly log-concave distributions. arXiv:1708.07114, 2017.

Mangoubi, O. and Vishnoi, N. K. Dimensionally tight running time bounds for second-order Hamiltonian Monte Carlo. In *Advances in Neural Information Processing Systems (NeurIPS) 32*, pp. 6030–6040. 2018.

Mou, W., Ma, Y.-A., Wainwright, M. J., Bartlett, P. L., and Jordan, M. I. High-order Langevin diffusion yields an accelerated MCMC algorithm. 2019.

Ranganath, R., Gerrish, S., and Blei, D. M. Black box variational inference. In *International Conference on Artificial Intelligence and Statistics*, pp. 814–822, 2014.

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

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 1278–1286, 2014.

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

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

Shen, R. and Lee, Y.-T. The randomized midpoint method for log-concave sampling. arXiv:1909.05503, 2019.

Su, W., Boyd, S., and Candes, E. A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights. In *Advances in Neural Information Processing Systems (NIPS) 27*, pp. 2510–2518. 2014.

Tzen, B. and Raginsky, M. Theoretical guarantees for sampling and inference in generative models with latent diffusions. In *Proceedings of the 32nd Conference on Learning Theory (COLT)*, pp. 3084–3114, 2019.

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

Wainwright, M. J. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019.

Wibisono, A. Sampling as optimization in the space of measures: The Langevin dynamics as a composite optimization problem. In *Proceedings of the 31st Conference On Learning Theory*, volume 75, pp. 2093–3027, 2018.

Wilson, A., Recht, B., and Jordan, M. I. A Lyapunov analysis of momentum methods in optimization. arXiv:1611.02635, 2016.

Y. Cao, J. Lu, L. W. On explicit L^2-convergence rate estimate for underdamped Langevin dynamics. arXiv:1908.04746, 2020.

Y. Wang, W. L. Accelerated information gradient flow. arXiv:1909.02102v1, 2019.

A. Proofs for convergence of variational inference

We study convergence of Λ_n to Λ^* in terms of the KL divergence from $p(z)$ to $q(z|\Lambda_n)$. Before proving the convergence rates for (stochastic) variational inference, we first derive a useful bound for the KL divergence, which will be used frequently in the proofs to follow.

Lemma 2. *The KL divergence between two normal distributions $p(z)$ and $q(z|\Lambda_n)$ is upper bounded by their difference in the Frobenius norm:*

$$\begin{aligned}&\mathrm{KL}\left(p(z)\|q(z|\Lambda_n)\right)\\&\leq \frac{1}{2}\|\left(\Lambda^*\right)^{-1}\|_2\cdot\|\left(\Lambda_n\right)^{-1}\|_2\cdot\|\Lambda^*-\Lambda_n\|_F^2\,.\end{aligned}$$

Proof of Lemma 2

$$\begin{aligned}&\mathrm{KL}\left(p(z)\|q(z|\Lambda_n)\right)\\&=\frac{1}{2}\left(-\log\frac{|\Lambda_n|}{|\Lambda^*|}+\mathrm{tr}((\Lambda^*)^{-1}\Lambda_n)-d\right)\\&=\frac{1}{2}\left(-\log\left|(\Lambda^*)^{-1}\Lambda_n\right|+\mathrm{tr}\left((\Lambda^*)^{-1}\Lambda_n-I\right)\right)\end{aligned}$$

Since

$$\log\left|(\Lambda^*)^{-1}\Lambda_n\right|\geq\mathrm{tr}\left(I-\Lambda_n^{-1}\Lambda^*\right),$$

$$\begin{aligned}&\mathrm{KL}\left(p(z)\|q(z|\Lambda_n)\right)\\&\leq\frac{1}{2}\mathrm{tr}\left((\Lambda^*)^{-1}\Lambda_n+\Lambda_n^{-1}\Lambda^*-2I\right)\\&=\frac{1}{2}\mathrm{tr}\left(\left((\Lambda^*)^{-1}(\Lambda_n-\Lambda^*)\right)+\left(\Lambda_n^{-1}(\Lambda^*-\Lambda_n)\right)\right)\\&=\frac{1}{2}\mathrm{tr}\left(\left(\Lambda_n^{-1}-(\Lambda^*)^{-1}\right)(\Lambda^*-\Lambda_n)\right)\\&\leq\frac{1}{2}\left\|\Lambda_n^{-1}-(\Lambda^*)^{-1}\right\|_F\cdot\|\Lambda^*-\Lambda_n\|_F\\&\leq\frac{1}{2}\|\left(\Lambda^*\right)^{-1}\|_2\cdot\|\left(\Lambda_n\right)^{-1}\|_2\cdot\|\Lambda^*-\Lambda_n\|_F^2\,.\end{aligned}$$

□

A.1. Proof for "Open-Box" VI Convergence

Proof of Lemma 1 The update rule in equation 13 can be explicitly expressed as:

$$\begin{aligned}\Lambda_n&=\Lambda_{n-1}-h_{n-1}\,g(\Lambda_{n-1})\\&=\Lambda_{n-1}-\frac{h_{n-1}}{2}\left(\Lambda_{n-1}-\Lambda^*\right).\end{aligned}$$

When we take a constant step size $h_k=h=\frac{1}{2}$, we can obtain that

$$\Lambda_n=\left(\frac{1}{2}\right)^n\Lambda_0+\left(1-\left(\frac{1}{2}\right)^n\right)\Lambda^*.$$

Therefore,

$$\|\Lambda_n-\Lambda^*\|_F=\left(\frac{1}{2}\right)^n\|\Lambda_0-\Lambda^*\|_F$$

Using the result of Lemma 2, we can obtian that

$$\begin{aligned}&\mathrm{KL}\left(p(z)\|q(z|\Lambda_n)\right)\\&\leq \frac{1}{2}\|\left(\Lambda^*\right)^{-1}\|_2\cdot\|\left(\Lambda_n\right)^{-1}\|_2\cdot\|\Lambda^*-\Lambda_n\|_F^2.\end{aligned}$$

By Weyl's theorem, we know that the distance from any eigenvalue of Λ_n to the closest eigenvalue of Λ^* is upper bounded by $\|\Lambda_n-\Lambda^*\|_2\leq\|\Lambda_n-\Lambda^*\|_F$. Therefore, $\sigma_{\min}(\Lambda_n)\geq\sigma_{\min}(\Lambda^*)-\left(\frac{1}{2}\right)^n\|\Lambda_0-\Lambda^*\|_F$, resulting in the upper bound for the spectral norm of Λ_n that

$$\|\left(\Lambda_n\right)^{-1}\|_2\leq\frac{1}{\sigma_{\min}(\Lambda^*)-\left(\frac{1}{2}\right)^n\|\Lambda_0-\Lambda^*\|_F}.$$

Therefore, for any

$$n\geq\log_2\frac{2}{\sigma_{\min}(\Lambda^*)}\frac{\|\Lambda_0-\Lambda^*\|_F}{\epsilon},$$

$\mathrm{KL}\left(p(z)\|q(z|\Lambda_n)\right)\leq\epsilon$, for any $\epsilon\leq 1$. □

A.2. Proofs for "Black-Box" VI Convergence

Proof of Theorem 1 We obtain the convergence bound in $\|\Lambda^*-\Lambda_n\|_F^2$ and then incur Lemma 2 to finish the proof.

Lemma 3. *For the stochastic preconditioned gradient descent algorithm described in equation 15, if we take a step size of* $h=\frac{\tilde{\epsilon}}{4\sigma_{\max}^2(\Lambda^*)d\delta^2}$*, we can obtain that when*

$$n\geq\frac{4\sigma_{\max}^2(\Lambda^*)d\delta^2}{\nu\tilde{\epsilon}}\log\frac{2\|\Lambda_0-\Lambda^*\|_F^2}{\nu\tilde{\epsilon}},$$

$\|\Lambda_n-\Lambda^*\|_F^2\leq\tilde{\epsilon}$*, with probability* $1-\nu$.

Then by Weyl's theorem, we know that the distance from any eigenvalue of Λ_n to the closest eigenvalue of Λ^* is upper bounded by $\|\Lambda_n-\Lambda^*\|_2\leq\|\Lambda_n-\Lambda^*\|_F$. Therefore, $\sigma_{\min}(\Lambda_n)\geq\sigma_{\min}(\Lambda^*)-\sqrt{\tilde{\epsilon}}$, resulting in the upper bound for the spectral norm of Λ_n that

$$\|\left(\Lambda_n\right)^{-1}\|_2\leq\frac{1}{\sigma_{\min}(\Lambda^*)-\sqrt{\tilde{\epsilon}}}. \tag{17}$$

Applying equation 17 to Lemma 2, we upper bound the KL divergence by $\tilde{\epsilon}$:

$$\mathrm{KL}\left(p(z)\|q(z|\Lambda_n)\right)\leq\frac{1}{2\sigma_{\min}(\Lambda^*)}\frac{\tilde{\epsilon}}{\sigma_{\min}(\Lambda^*)-\sqrt{\tilde{\epsilon}}}.$$

Choosing $\tilde{\epsilon}=\frac{\sigma_{\min}^2(\Lambda^*)}{2}\epsilon$ completes the proof that after

$$n\geq 8\frac{\sigma_{\max}^2(\Lambda^*)\delta^2}{\nu\sigma_{\min}^2(\Lambda^*)}\frac{d}{\epsilon}\cdot\log\frac{4\|\Lambda_0-\Lambda^*\|_F^2}{\nu\sigma_{\min}^2(\Lambda^*)\epsilon}=\widetilde{\mathcal{O}}\left(\frac{\sigma_{\max}^2(\Lambda^*)\delta^2}{\sigma_{\min}^2(\Lambda^*)}\frac{d}{\epsilon}\right).$$

number of iterations, $\mathrm{KL}\left(p(z)\|q(z|\Lambda_n)\right)\leq\epsilon$. □

Proof of Lemma 3 We first prove the convergence in $\mathbb{E}\|\Lambda_n-\Lambda^*\|_F^2$ then invoke the Chebychev inequality for the high probability statement.

Since we assumed in equation 14 that $\mathbb{E}\left[\Delta(\Lambda; \mathcal{D}_n)\right] = 0$, and that $\mathbb{E}\left\|\Delta(\Lambda; \mathcal{D}_n)\right\|_F^2 \leq \sigma_{\max}^2(\Lambda^*) d\delta^2$, for any n,

$$
\begin{aligned}
&\mathbb{E}\|\Lambda_n - \Lambda^*\|_F^2 \\
&= \mathbb{E}\left\|\Lambda_{n-1} - \Lambda^* - h_{n-1}\hat{g}(\Lambda)\right\|_F^2 \\
&= \mathbb{E}\left\|\left(1 - \frac{h_{n-1}}{2}\right)(\Lambda_{n-1} - \Lambda^*) + h_{n-1}\Delta(\Lambda_{n-1}; \mathcal{D}_{n-1})\right\|_F^2 \\
&= \left(1 - \frac{h_{n-1}}{2}\right)^2 \mathbb{E}\left\|\Lambda_{n-1} - \Lambda^*\right\|_F^2 + 2h_{n-1}\left(1 - \frac{h_{n-1}}{2}\right)\mathbb{E}\left\langle \Lambda_{n-1} - \Lambda^*, \Delta(\Lambda_{n-1}; \mathcal{D}_{n-1})\right\rangle_F + h_{n-1}^2\mathbb{E}\|\Delta(\Lambda_{n-1}; \mathcal{D}_{n-1})\|_F^2 \\
&= \left(1 - \frac{h_{n-1}}{2}\right)^2 \mathbb{E}\left\|\Lambda_{n-1} - \Lambda^*\right\|_F^2 + h_{n-1}^2\mathbb{E}\|\Delta(\Lambda_{n-1}; \mathcal{D}_{n-1})\|_F^2 \\
&\leq \left(1 - \frac{h_{n-1}}{2}\right)\mathbb{E}\left\|\Lambda_{n-1} - \Lambda^*\right\|_F^2 + h_{n-1}^2\mathbb{E}\|\Delta(\Lambda_{n-1}; \mathcal{D}_{n-1})\|_F^2 \\
&\leq \prod_{i=0}^{n-1}\left(1 - \frac{h_i}{2}\right)\mathbb{E}\left\|\Lambda_0 - \Lambda^*\right\|_F^2 + \sum_{j=0}^{n-1} h_j^2 \left(\prod_{i=j+1}^{n-1}\left(1 - \frac{h_i}{2}\right)\right)\mathbb{E}\|\Delta(\Lambda_{j-1}; \mathcal{D}_{j-1})\|_F^2 \\
&\leq \prod_{i=0}^{n-1}\left(1 - \frac{h_i}{2}\right)\mathbb{E}\left\|\Lambda_0 - \Lambda^*\right\|_F^2 + \sum_{j=0}^{n-1} h_j^2 \prod_{i=j+1}^{n-1}\left(1 - \frac{h_i}{2}\right)\, \sigma_{\max}^2(\Lambda^*) d\delta^2.
\end{aligned}
$$

When we take a constant step size, $h_k = h$, the above expression simplifies to:

$$
\begin{aligned}
\mathbb{E}\|\Lambda_n - \Lambda^*\|_F^2 &\leq \left(1 - \frac{h_i}{2}\right)^n \mathbb{E}\left\|\Lambda_0 - \Lambda^*\right\|_F^2 + 2h\left(\left(1 - \frac{h_i}{2}\right) - \left(1 - \frac{h_i}{2}\right)^n\right)\sigma_{\max}^2(\Lambda^*) d\delta^2 \\
&\leq \left(1 - \frac{h_i}{2}\right)^n \mathbb{E}\left\|\Lambda_0 - \Lambda^*\right\|_F^2 + 2h\sigma_{\max}^2(\Lambda^*) d\delta^2. \qquad (18)
\end{aligned}
$$

We then invoke the following Chebyshev inequality to obtain the high probability statement:

$$
\mathbb{P}\left(\|\Lambda_n - \Lambda^*\|_F^2 \geq \tilde{\epsilon}\right) \leq \frac{1}{\tilde{\epsilon}}\mathbb{E}\|\Lambda_n - \Lambda^*\|_F^2.
$$

For $\|\Lambda_n - \Lambda^*\|_F^2 \leq \tilde{\epsilon}$ to hold with $1 - \nu$ probability, we need $\|\Lambda_n - \Lambda^*\|_F^2 \leq \nu\tilde{\epsilon}$.

Choosing $h = \frac{\nu\tilde{\epsilon}}{4\sigma_{\max}^2(\Lambda^*) d\delta^2}$, we arrive at our conclusion that $\|\Lambda_n - \Lambda^*\|_F^2 \leq \tilde{\epsilon}$ with probability $1 - \nu$, when

$$
n \geq \frac{4\sigma_{\max}^2(\Lambda^*) d\delta^2}{\nu\tilde{\epsilon}} \log \frac{2\left\|\Lambda_0 - \Lambda^*\right\|_F^2}{\nu\tilde{\epsilon}},
$$

where the log factor can be shaved off by employing a decreasing step size. □

Tightness of the bounds We now demonstrate that the convergence upper bound in Theorem 1 is tight up to a logarithmic factor. We first prove that the Frobenius norm bound in Lemma 3, instead of a spectral norm bound, is indeed necessary to guarantee the convergence in KL divergence.

To this end, we examine an example of the posterior with the precision matrix $\Lambda^* = \frac{1}{4}I$. If the initial distribution has the precision matrix $\Lambda_0 = I$, then $\|\Lambda_0 - \Lambda^*\|_2 = \frac{3}{4}$. However,

$$
\begin{aligned}
&\mathrm{KL}\left(p(z)\|q(z|\Lambda_0)\right) \\
&= \frac{1}{2}\left(-\log\frac{|\Lambda_0|}{|\Lambda^*|} + \mathrm{tr}((\Lambda^*)^{-1}\Lambda_0) - d\right) \geq d,
\end{aligned}
$$

which can be arbitrarily large as dimension d increases.

We then use the same posterior of $\Lambda^* = \frac{1}{4}I$ and take an initial value Λ_0 so that $\|\Lambda_0 - \Lambda^*\|_F^2$ scales inclusively between $\Omega(1)$ and $\mathcal{O}(d)$. Under this mild condition, we demonstrate that the number of iterations, n, required for $\|\Lambda_0 - \Lambda^*\|_F^2$ to decrease to $\|\Lambda_n - \Lambda^*\|_F^2 \leq \frac{1}{2}\|\Lambda_0 - \Lambda^*\|_F^2$ is $n = \Omega(d)$.

We first demonstrate that $\mathbb{E}\|\Delta(\Lambda;\mathcal{D}_n)\|_F^2 = \Omega(d)$ for minibatch size $|\mathcal{D}_n| = \mathcal{O}(d)$. From Section 4.3, we know that

$$\begin{aligned}
&\mathbb{E}\|\Delta(\Lambda;\mathcal{D}_n)\|_F^2 \\
&= \frac{1}{|\mathcal{D}_n|}\mathbb{E}_{z\sim q}\left[\|v(\Lambda;z) - \mathbb{E}_{\hat{z}\sim q}[v(\Lambda;\hat{z})]\|_F^2\right] \\
&= \frac{1}{2|\mathcal{D}_n|}\left(\operatorname{tr}(\Lambda-\Lambda^*)\right)^2 - \frac{1}{4|\mathcal{D}_n|}\|\Lambda-\Lambda^*\|_F^2 \\
&+ \frac{1}{|\mathcal{D}_n|}\left(\frac{1}{8}\left\|I-\Lambda^*\Lambda^{-1}\right\|_F^2 + \frac{1}{16}\left(\mathrm{KL}\left(q(z|\Lambda)\|p(z)\right)\right)^2\right)\cdot\left(\left(\operatorname{tr}(\Lambda)\right)^2 + \operatorname{tr}\left(\Lambda^2\right)\right).
\end{aligned}$$

Since

$$\|\Lambda-\Lambda^*\|_F \le \|\Lambda\|_2 \cdot \|I-\Lambda^{-1}\Lambda^*\|_F,$$

we employ Weyl's theorem and obtain that

$$\begin{aligned}
\|I-\Lambda^{-1}\Lambda^*\|_F &\ge \frac{\|\Lambda-\Lambda^*\|_F}{\sigma_{\max}(\Lambda)} \\
&\ge \frac{\|\Lambda-\Lambda^*\|_F}{\sigma_{\max}(\Lambda^*) + \|\Lambda-\Lambda^*\|_F} = \Omega(1).
\end{aligned}$$

Therefore, $\mathbb{E}\|\Delta(\Lambda;\mathcal{D}_n)\|_F^2 = \Omega(1)$ for $|\mathcal{D}_n| = \mathcal{O}(d)$ and for $\|\Lambda-\Lambda^*\|_F^2 = \Omega(1/d)$ and $\|\Lambda-\Lambda^*\|_F^2 = \mathcal{O}(d)$.

We then analyze number of steps n required for $\|\Lambda_n-\Lambda^*\|_F^2 \le \frac{1}{2}\|\Lambda_0-\Lambda^*\|_F^2$. In the update rule of equation 15,

$$\begin{aligned}
\Lambda_n &= \Lambda_{n-1} - h_{n-1}\hat{g}(\Lambda_{n-1}) \\
&= \Lambda_{n-1} - \frac{h_{n-1}}{2}(\Lambda_{n-1}-\Lambda^*) + h_{n-1}\Delta(\Lambda_{n-1};\mathcal{D}_{n-1}).
\end{aligned}$$

Hence

$$\begin{aligned}
\Lambda_n - \Lambda^* &= \left(1-\frac{h_{n-1}}{2}\right)(\Lambda_{n-1}-\Lambda^*) + h_{n-1}\Delta(\Lambda_{n-1};\mathcal{D}_{n-1}) \\
&= \prod_{i=0}^{n-1}\left(1-\frac{h_i}{2}\right)(\Lambda_0-\Lambda^*) + \sum_{j=0}^{n-1} h_j \prod_{i=j+1}^{n-1}\left(1-\frac{h_i}{2}\right)\Delta(\Lambda_j;\mathcal{D}_j).
\end{aligned}$$

Since $\mathcal{D}_n$ are sampled in an i.i.d. fashion and that $\mathbb{E}\left[\Delta(\Lambda;\mathcal{D}_n)\right] = 0$ from assumption 14,

$$\mathbb{E}\|\Lambda_n-\Lambda^*\|_F^2 = \prod_{i=0}^{n-1}\left(1-\frac{h_i}{2}\right)^2 \mathbb{E}\|\Lambda_0-\Lambda^*\|_F^2 + \sum_{j=0}^{n-1} h_j^2 \prod_{i=j+1}^{n-1}\left(1-\frac{h_i}{2}\right)^2 \mathbb{E}\|\Delta(\Lambda_j;\mathcal{D}_j)\|_F^2.$$

Since $\mathbb{E}\|\Delta(\Lambda;\mathcal{D}_n)\|_F^2 = \Omega(d)$, to have that $\|\Lambda_n-\Lambda^*\|_F^2 \le \frac{1}{2}\|\Lambda_0-\Lambda^*\|_F^2$ with a constant probability, we must require

$$\sum_{j=0}^{n-1} h_j^2 \prod_{i=j+1}^{n-1}\left(1-\frac{h_i}{2}\right)^2 = \mathcal{O}\left(\frac{1}{d}\right),$$

which implies that $h_j = \mathcal{O}\left(\frac{1}{d}\right), \forall j = 0,\cdots,n-1$. On the other hand, to achieve $\|\Lambda_n-\Lambda^*\|_F \le \frac{1}{2}\|\Lambda_0-\Lambda^*\|_F$, we also need

$$\prod_{i=0}^{n-1}\left(1-\frac{h_i}{2}\right)\|\Lambda_0-\Lambda^*\|_F \le \frac{1}{2}\|\Lambda_0-\Lambda^*\|_F,$$

which implies that

$$\begin{aligned}
\sum_{i=0}^{n-1} h_i &\ge \sum_{i=0}^{n-1}\left(\left(1-\frac{h_i}{2}\right)^{-1} - 1\right) \\
&\ge \sum_{i=0}^{n-1}\log\left(\left(1-\frac{h_i}{2}\right)^{-1}\right) \\
&\ge \log\frac{\|\Lambda_0-\Lambda^*\|_F}{\|\Lambda_n-\Lambda^*\|_F} = \log(2).
\end{aligned}$$

Since $h_j = \mathcal{O}\left(\frac{1}{d}\right), \forall j$, we need $n = \Omega(d)$ for convergence. □

B. Proofs for convergence of Langevin algorithm

Proof of Lemma 1 Before proving Lemma 1, we first make the assumptions explicit. We are interested in generating samples from $p(\theta) \propto \exp\left(-U(\theta)\right)$, where $U(\theta)$ is L-Lipschitz smooth and m-strongly convex. We further assume, without loss of generality, that U has a fixed point at the origin 0: $\nabla U(0) = 0$.

To prove Lemma 1, we first analyze equation 3 as a discretization scheme of the Langevin diffusion of equation 4. Within each iteration, the ULA update 3 is effectively integrating the following dynamics:

$$\begin{aligned} d\theta_t &= \nabla \log p(\theta_n)dt + \sqrt{2}dW_t \\ &= \nabla \log p(\theta_t)dt + \sqrt{2}dW_t + \left(\nabla \log p(\theta_n) - \nabla \log p(\theta_t)\right)dt, \end{aligned} \tag{19}$$

for $t \in [n\eta, (n+1)\eta]$.

We then analyze the time derivative of the KL divergence $\mathrm{KL}\left(q_t \| p\right)$ within each step:

$$\begin{aligned} \frac{d}{dt}\mathrm{KL}\left(q_t \| p\right) &= -\mathbb{E}\left\langle \nabla \log \frac{q_t(\theta_t)}{p(\theta_t)}, \nabla \log \frac{q_t(\theta_t)}{p(\theta_t)} + \left(\nabla \log p(\theta_n) - \nabla \log p(\theta_t)\right) \right\rangle \\ &= -\mathbb{E}\left\| \nabla \log \frac{q_t(\theta_t)}{p(\theta_t)} \right\|^2 + \mathbb{E}\left\langle \nabla \log \frac{q_t(\theta_t)}{p(\theta_t)}, \nabla \log p(\theta_t) - \nabla \log p(\theta_n) \right\rangle, \end{aligned} \tag{20}$$

where the expectation is taken with respect to the joint distribution of θ_t and θ_n. For the second term in equation 20, we invoke Young's inequality to bound:

$$\begin{aligned} \mathbb{E}\left\langle \nabla \log \frac{q_t(\theta_t)}{p(\theta_t)}, \nabla \log p(\theta_t) - \nabla \log p(\theta_n) \right\rangle &\leq \frac{1}{2}\mathbb{E}\left\| \nabla \log \frac{q_t(\theta_t)}{p(\theta_t)} \right\|^2 + \frac{1}{2}\mathbb{E}\left\| \nabla \log p(\theta_t) - \nabla \log p(\theta_n) \right\|^2 \\ &= \frac{1}{2}\mathbb{E}\left\| \nabla \log \frac{q_t(\theta_t)}{p(\theta_t)} \right\|^2 + \frac{1}{2}\mathbb{E}\left\| \nabla U(\theta_t) - \nabla U(\theta_n) \right\|^2. \end{aligned}$$

Since potential U is L-Lipschitz smooth, $\left\| \nabla U(\theta_t) - \nabla U(\theta_n) \right\|^2 \leq L^2 \|\theta_t - \theta_n\|^2$. Also note that we have set $\nabla U(0) = 0$. Hence

$$\begin{aligned} \frac{1}{2}\mathbb{E}\left\| \nabla U(\theta_t) - \nabla U(\theta_n) \right\|^2 &\leq \frac{L^2}{2}\mathbb{E}\|\theta_t - \theta_n\|^2 \\ &= \frac{L^2}{2}\mathbb{E}\left\| -(t - \eta n)\nabla U(\theta_n) + \sqrt{2}\left(W_t - W_{\eta n}\right) \right\|^2 \\ &= \frac{L^2(t - \eta n)^2}{2}\mathbb{E}_{\theta \sim q_n}\left[\|\nabla U(\theta)\|^2 \right] + L^2 d(t - \eta n) \\ &\leq \frac{L^4 \eta^2}{2}\mathbb{E}_{\theta \sim q_n}\left[\|\theta\|^2 \right] + L^2 d\eta \end{aligned}$$

Applying this result to equation 20, we obtain an upper bound for $\frac{d}{dt}\mathrm{KL}\left(q_t \| p\right)$ within each iteration:

$$\frac{d}{dt}\mathrm{KL}\left(q_t \| p\right) \leq -\frac{1}{2}\mathbb{E}\left\| \nabla \log \frac{q_t(\theta_t)}{p(\theta_t)} \right\|^2 + \frac{L^4 \eta^2}{2}\mathbb{E}_{\theta \sim q_n}\left[\|\theta\|^2 \right] + L^2 d\eta. \tag{21}$$

Since function U is m-strongly convex, we obtain the following log-Sobolev inequality from the Bakry–Emery criterion (see e.g., Bakry & Emery, 1985)

$$\mathbb{E}_{\theta \sim q_t}\left[\left\| \nabla \log \frac{q_t(\theta)}{p(\theta)} \right\|^2 \right] \geq 2m\mathrm{KL}\left(q_t \| p\right).$$

Therefore,

$$\frac{d}{dt}\mathrm{KL}\left(q_t \| p\right) \leq -m\mathrm{KL}\left(q_t \| p\right) + \frac{L^4 \eta^2}{2}\mathbb{E}_{\theta \sim q_n}\left[\|\theta\|^2 \right] + L^2 d\eta. \tag{22}$$

We prove in the Lemma 4 below that $\mathbb{E}_{\theta\sim q_n}\left[\|\theta\|^2\right] \le \frac{4d}{m}$.

Lemma 4. *For step size $\eta \le \frac{1}{L}$, and for q_n following the update of equation 3, $\forall n > 0$, $\mathbb{E}_{\theta\sim q_n}\left[\|\theta\|^2\right] \le \frac{4d}{m}$.*

Plugging this bound into equation 22, we obtain:

$$\frac{d}{dt}\mathrm{KL}\left(q_t\|p\right) \le -m\left(\mathrm{KL}\left(q_t\|p\right) - \left(2\frac{L^4}{m^2}\eta^2 + \frac{L^2}{m}\eta\right)d\right). \tag{23}$$

Invoking Grönwall's inequality, we obtain:

$$\begin{aligned}\mathrm{KL}\left(q_n\|p\right) &\le e^{-m\eta}\mathrm{KL}\left(q_{n-1}\|p\right) + \left(2\frac{L^4}{m^2}\eta^2 + \frac{L^2}{m}\eta\right)d \\ &\le e^{-m\eta n}\mathrm{KL}\left(q_0\|p\right) + \left(2\frac{L^4}{m^2}\eta^2 + \frac{L^2}{m}\eta\right)d.\end{aligned} \tag{24}$$

This means that $\mathrm{KL}\left(q_n\|p\right)$ is converging exponentially to the level of discretization error.

To obtain an accuracy guarantee of ϵ, we choose a step size of $\eta = \frac{m}{4L^2}\frac{\epsilon}{d}$ and have (for $\epsilon \le d$):

$$\mathrm{KL}\left(q_n\|p\right) \le e^{-m\eta n}\mathrm{KL}\left(q_0\|p\right) + \frac{\epsilon}{2}. \tag{25}$$

When $n \ge \frac{1}{m\eta}\log\frac{2\mathrm{KL}(q_0\|p)}{\epsilon}$, $e^{-m\eta n}\mathrm{KL}\left(q_0\|p\right) \le \frac{\epsilon}{2}$, and therefore $\mathrm{KL}\left(q_n\|p\right) \le \epsilon$.

Plugging the setting of η gives us the upper bound for number of iterations:

$$n = 4\frac{L^2}{m^2}\frac{d}{\epsilon}\log\frac{2\mathrm{KL}\left(q_0\|p\right)}{\epsilon} = \widetilde{\mathcal{O}}\left(\frac{L^2}{m^2}\frac{d}{\epsilon}\right).$$

□

Proof of Lemma 4 We prove Lemma 2 by induction. We first see that for the current choice of initialization, $\mathbb{E}_{\theta\sim q_0}\left[\|\theta\|^2\right] \le \frac{d}{m}$. We then assume that $\mathbb{E}_{\theta\sim q_n}\left[\|\theta\|^2\right] \le \frac{d}{m}$ and prove that $\mathbb{E}_{\theta\sim q_{n+1}}\left[\|\theta\|^2\right] \le \frac{d}{m}$.

We know that

$$\theta_{n+1} = \theta_n - \eta\nabla U(\theta_n) + \sqrt{2}\left(W_{\eta(n+1)} - W_{\eta n}\right).$$

To provide a bound on $\|\theta_{n+1}\|$, we first analyze the term: $\theta_n - \eta\nabla U(\theta_n)$. To this end, we construct a function: $V(\theta) = \frac{1}{2}\|\theta\|^2 - \eta U(\theta)$ and prove that it is $(1-m\eta)$-Lipschitz smooth. Since function U is assumed to be m-strongly convex,

$$\begin{aligned}\langle\nabla V(\theta) - \nabla V(\vartheta), \theta - \vartheta\rangle &= \langle(\theta - \vartheta) - \eta\left(\nabla U(\theta) - U(\vartheta)\right), \theta - \vartheta\rangle \\ &= \|\theta - \vartheta\|^2 - \eta\left\langle\left(\nabla U(\theta) - U(\vartheta)\right), \theta - \vartheta\right\rangle \\ &\le (1 - m\eta)\|\theta - \vartheta\|^2.\end{aligned}$$

Therefore, function $V(\theta) = \frac{1}{2}\|\theta\|^2 - \eta U(\theta)$ is $(1-m\eta)$-Lipschitz smooth and satisfy $\nabla V(0) = 0$, which means:

$$\|\theta_n - \eta\nabla U(\theta_n)\| = \|\nabla V(\theta_n)\| \le (1 - m\eta)\|\theta_n\|.$$

We are now in a position to bound $\mathbb{E}\|\theta_{n+1}\|^2$:

$$\begin{aligned}\mathbb{E}\left[\|\theta_{n+1}\|^2\right] &= \mathbb{E}\left[\left\|\theta_n - \eta\nabla U(\theta_n) + \sqrt{2}\left(W_{\eta(n+1)} - W_{\eta n}\right)\right\|^2\right] \\ &= \mathbb{E}\left[\|\theta_n - \eta\nabla U(\theta_n)\|^2\right] + 2\eta d \\ &\le (1 - m\eta)\mathbb{E}\left[\|\theta_n\|^2\right] + 2\eta d \qquad (26) \\ &= \mathbb{E}\left[\|\theta_n\|^2\right] + \eta\left(2d - m\mathbb{E}\left[\|\theta_n\|^2\right]\right). \qquad (27)\end{aligned}$$

By the inductive hypothesis, $\mathbb{E}\left[\|\theta_n\|^2\right] \leq \frac{4d}{m}$. If $\mathbb{E}\left[\|\theta_n\|^2\right] \geq \frac{2d}{m}$, then $\left(2d - m\mathbb{E}\left[\|\theta_n\|^2\right]\right) \leq 0$, $\mathbb{E}\left[\|\theta_{n+1}\|^2\right] \leq \mathbb{E}\left[\|\theta_n\|^2\right] \leq \frac{4d}{m}$. If $\mathbb{E}\left[\|\theta_n\|^2\right] \leq \frac{2d}{m}$ instead, then we use line 26 and that $\eta \leq \frac{1}{L}$ to obtain: $\mathbb{E}\|\theta_{n+1}\|^2 \leq \left(1 - \frac{m}{L}\right) \frac{2d}{m} + \frac{2d}{L} \leq \frac{4d}{m}$.

Therefore, we have proven that for any $n > 0$, $\mathbb{E}_{\theta \sim q_n}\left[\|\theta\|^2\right] \leq \frac{4d}{m}$ by induction. □