Variance Reduction and Quasi-Newton for Particle-Based Variational Inference

Michael H. Zhu [1] **Chang Liu** [2] **Jun Zhu** [3]

Abstract

Particle-based Variational Inference methods (ParVIs), like Stein Variational Gradient Descent, are nonparametric variational inference methods that optimize a set of particles to best approximate a target distribution. ParVIs have been proposed as efficient approximate inference algorithms and as potential alternatives to MCMC methods. However, to our knowledge, the quality of the posterior approximation of particles from ParVIs has not been examined before for large-scale Bayesian inference problems. We conduct this analysis and evaluate the sample quality of particles produced by ParVIs, and we find that existing ParVI approaches using stochastic gradients converge insufficiently fast under sample quality metrics. We propose a novel variance reduction and quasi-Newton preconditioning framework for ParVIs, by leveraging the Riemannian structure of the Wasserstein space and advanced Riemannian optimization algorithms. Experimental results demonstrate the accelerated convergence of variance reduction and quasi-Newton methods for ParVIs for accurate posterior inference in large-scale and ill-conditioned problems.

1. Introduction

A central problem in Bayesian inference is approximating an intractable posterior distribution p and estimating intractable expectations $\mathbb{E}_p[f(X)] = \int f(x)p(x)\,\mathrm{d}x$ with respect to p. MCMC methods (Brooks et al., 2011) are based on simulating a Markov chain with limiting distribution p, drawing samples $x_1, x_2, \ldots$ which represent p, and computing the sample average $\frac{1}{M}\sum_{i=1}^{M} f(x_i)$ which is an asymptotically exact estimator of $\mathbb{E}_p[f(X)]$ as $M \to \infty$. Variational inference (VI) methods (Wainwright et al., 2008; Blei et al., 2017) recast the inference problem as a parametric optimization problem and attempt to globally approximate the posterior distribution p with a tractable distribution from some variational family. MCMC methods are asymptotically exact but can be slow; VI methods can be fast but are generally biased.

Particle-based Variational Inference methods (ParVIs) are nonparametric variational inference methods that optimize a set of particles $\{x_1, x_2, \ldots, x_M\}$ to best represent p. Stein variational gradient descent (SVGD) (Liu & Wang, 2016) is a leading instance of ParVIs that has received an increasing number of extensions (e.g., Zhuo et al. (2018); Chen et al. (2018a;b); Wang et al. (2019)) and applications (e.g., Feng et al. (2017); Pu et al. (2017); Liu et al. (2017); Yoon et al. (2018)). ParVIs have been proposed as efficient approximate inference algorithms potentially combining the advantages of MCMC and VI for large-scale Bayesian inference problems. However, to our knowledge, an important question has not yet been explored for large-scale Bayesian inference problems: for a given posterior distribution p, how well do the particles $\{x_1, x_2, \ldots, x_M\}$ produced by ParVIs represent p in practice, and how accurate is the estimator $\frac{1}{M}\sum_{i=1}^{M} f(x_i)$ of $\mathbb{E}_p[f(X)]$ in practice?

We explore this question in the context of Bayesian linear regression and logistic regression, two fundamental real-world inference tasks. We conduct a careful empirical inspection of the sample quality of particles produced by ParVIs under various metrics, including mean squared error for estimating posterior mean and covariance, maximum mean discrepancy (Gretton et al., 2012), and kernel Stein discrepancy (Chwialkowski et al., 2016; Liu et al., 2016). We find that existing ParVI approaches using stochastic gradients converge insufficiently fast under these sample quality metrics, especially in large-scale and ill-conditioned scenarios.

For accurate posterior inference, highly accurate solutions to the ParVI optimization problem are needed. In the context of large-scale optimization, a popular method is stochastic gradient descent (SGD) because of the fast per-iteration computation time. While SGD can reach an approximate solution relatively quickly, it has slow asymptotic conver-

[1]Department of Computer Science, Stanford University, Stanford, CA, USA [2]Microsoft Research Asia, Beijing, 100080, China [3]Dept. of Comp. Sci. & Tech., Institute for AI, BNRist Center, Tsinghua-Bosch ML Center, Tsinghua University, Beijing, 100084, China. Correspondence to: J. Zhu <dcszj@tsinghua.edu.cn>, Chang Liu <changliu@microsoft.com>, Michael H. Zhu <mhzhu@cs.stanford.edu>.

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

gence due to the high variance from the random sampling of data points and the resulting need for decaying step sizes. Variance reduction methods for stochastic gradient descent (Roux et al., 2012; Johnson & Zhang, 2013; Defazio et al., 2014) have been proven, in theory and in practice, to accelerate convergence for strongly convex problems when highly accurate solutions are needed.

For ill-conditioned problems, however, the convergence speeds of first-order gradient methods can be slow. One general solution is quasi-Newton methods (Nocedal & Wright, 2006), like L-BFGS, which use the history of gradients to approximate the inverse Hessian of the objective function and scale each step by the approximate inverse Hessian to account for the curvature of the function. Extending traditional, full-batch L-BFGS methods to the stochastic setting (Byrd et al., 2016) is challenging since noisy Hessian approximations combined with high-variance stochastic gradients can be unstable. Combining variance reduction and stochastic quasi-Newton methods (Moritz et al., 2016) leads to stable Hessian approximations and low-variance stochastic gradients and has been shown to accelerate convergence when highly accurate solutions are needed.

We propose a novel variance reduction and quasi-Newton preconditioning framework for ParVIs. We follow the gradient flow perspective of ParVIs as optimization methods for the KL divergence on the Wasserstein space, a manifold of probability distributions (Chen et al., 2018a; Liu et al., 2019). We develop our framework for ParVIs by leveraging the Riemannian structure of the Wasserstein space and Riemannian variance reduction (Zhang et al., 2016; Zhou et al., 2019) and quasi-Newton methods (Roychowdhury & Parthasarathy, 2017; Kasai et al., 2018) that enjoy proven acceleration. Our approach is more principled than intuitively applying optimization techniques to the update rule of each particle, as the ParVI optimization problem is not defined on the support space. In handling the Riemannian structure of the Wasserstein space, we trade-off the accuracy and computational cost of geometric approximation and get stable and practical algorithms. Moreover, as the Wasserstein optimization perspective is general for ParVIs, our framework is applicable to various ParVI instances. Under the same experimental setup, our proposed methods greatly improve the convergence rate.

2. Related work

Variance reduction Dubey et al. (2016) develop variance reduction techniques for Stochastic Gradient Langevin Dynamics (SGLD), and Chatterji et al. (2018) prove sharp theoretical bounds showing that variance reduction methods for SGLD converge faster when highly accurate solutions are needed. Li et al. (2019) develop variance reduction for Hamiltonian Monte Carlo. Zhang et al. (2018b) propose variance reduction techniques for SPOS (Zhang et al., 2018a), an algorithm combining SGLD and SVGD. Their VR-SPOS algorithm, however, is limited to SPOS since they rely on a convergence analysis only applicable to SPOS.

ParVIs and second-order information The Stein variational Newton method (Detommaso et al., 2018) uses an approximate Newton-like update based on a computable approximation to a functional Newton direction. Their method explicitly computes the Hessian of the log-density, so it is not a quasi-Newton method. Wang et al. (2019) present a generalization of SVGD with matrix-valued kernels and propose using preconditioning matrices, such as the Hessian and Fisher information matrix, to incorporate geometric information into SVGD updates. They propose a practical algorithm based on a weighted average of Hessians at anchor points, which is an intuitive approximation. As the Hessian is a local property, distant anchor points do not necessarily hold useful and consistent information for the Hessian at the current position. Our method is based on L-BFGS, where the assumption on the Hessian approximator is clear (i.e., the matrix satisfying secant equation that is the closest to the previous approximator). L-BFGS variants also have convergence bound guarantees (e.g., Kasai et al. (2018)). Moreover, the two methods above are developed on the $\mathcal{P}_{\mathcal{H}}$ manifold (probability space that has RKHS as its tangent space) (Liu, 2017), which may not be well-defined (Liu et al., 2019) and only benefits the particular ParVI of SVGD. Another higher-order method is the Riemannian SVGD method (Liu & Zhu, 2018), which generalizes SVGD to Riemannian support space (i.e., sample/particle space). Riemannian SVGD also requires the exact Hessian, so it is not quasi-Newton. While Riemannian SVGD requires a properly conceived metric, our method directly utilizes the geometry of the Wasserstein objective and is more flexible.

3. Preliminaries

Our variance reduction and quasi-Newton framework for ParVIs is developed in the context of Riemannian geometry, where we utilize the Riemannian structure of the Wasserstein space. We first introduce these related concepts.

3.1. Riemannian Manifold

A manifold $\mathcal{M}$ is a topological space that *locally* behaves like a Euclidean space (i.e., *locally* homeomorphic to an Euclidean open subset; see e.g., Do Carmo (1992); Abraham et al. (2012)). A manifold releases linear structures and generalizes linear space to allow curvature, while still coming with handy structures. To begin with, a tangent vector at $x \in \mathcal{M}$ admits a general definition as a directional derivative operator at x, and all such vectors form a linear space regarded as the tangent space $T_x\mathcal{M}$ at x. When every tangent space is endowed with an inner product $\langle \cdot, \cdot \rangle_{T_x\mathcal{M}}$,

the manifold is called a Riemannian manifold, which induces more structures. The gradient of a function f at x is the unique tangent vector such that for any $v \in T_x\mathcal{M}$, $\langle \operatorname{grad} f(x), v\rangle_{T_x\mathcal{M}}$ is the directional derivative of f along v. As the same in the linear case, it is the steepest ascending direction for f at x, thus being the foundation of Riemannian optimization methods. The length of a smooth curve $\gamma : [a, b] \to \mathcal{M}$ can be defined as the integral over velocity: $\int_a^b \|\dot\gamma_t\|_{T_x\mathcal{M}} \,\mathrm{d}t$ where $\dot\gamma_t$ denotes the tangent vector along the curve at γ_t, and the curve with minimal length between any adjacent point pair on it is called a geodesic[1]. The exponential map $\operatorname{Exp}_x(v)$ is used as the counterpart of vector addition to update points in optimization methods. It transports a point x to another by walking along the geodesic tangent to $v \in T_x\mathcal{M}$ at x for length $\|v\|_{T_x\mathcal{M}}$. The parallel transport $\Gamma_x^y(v)$ links tangent vectors at different points. It moves a tangent vector v at x to one at y along the geodesic from x to y, in a certain way that is regarded as parallel.

3.2. The Wasserstein Space

The (2-)Wasserstein space $\mathcal{P}_2$ is the set of distributions on a support space (i.e., sample/particle space) with finite second-order moments. It is very inclusive and cannot be expressed with parametric form. Nevertheless, the structure of its tangent space makes it convenient to express its elements by samples. Here we consider Euclidean support space $\mathbb{R}^m$, and treat the corresponding $\mathcal{P}_2$ as an infinite dimensional manifold. Let q be a point on $\mathcal{P}_2$, which is a distribution on $\mathbb{R}^m$, and let $\{x^{(i)}\}_{i=1}^M$ be a set of samples, also called particles, of q. Consider updating the particles with a vector field V on $\mathbb{R}^m$ ($V(x) \in \mathbb{R}^m, \forall x \in \mathbb{R}^m$) for an infinitesimal $\varepsilon > 0$: $\{x^{(i)} + \varepsilon V(x^{(i)})\}_{i=1}^M$, and denote the distribution that this new set of particles obeys as q_ε. Taking the continuous limit $\varepsilon \to 0$ and repeatedly applying this procedure, the vector field V *induces* a smooth curve of distributions $(q_t)_t$ on $\mathcal{P}_2$ around q. Such a vector field V is not unique for inducing a given distribution curve around q, but all these vector fields form an equivalent class under the equivalent relation: $U \simeq V$ if $\nabla \cdot (qU - qV) = 0$ where "$\nabla \cdot V$" denotes the divergence of vector field V. In each equivalent class, the vector field with the minimum $\mathcal{L}_q^2$-norm $\sqrt{\mathbb{E}_q[V \cdot V]}$ can be taken as the representor of the class, where "$\cdot$" denotes the conventional vector inner product. All such representors form a linear subspace $\overline{\{\nabla\varphi \mid \varphi \in C_c^\infty\}}^{\mathcal{L}_q^2}$ of $\mathcal{L}_q^2 := \{V \mid \mathbb{E}_q[V \cdot V] < \infty\}$, where C_c^∞ is the set of compactly supported scalar-valued smooth functions, and the overline means closure. It is the orthonormal complement of the equivalent class in $\mathcal{L}_q^2$ (Erbar et al., 2010). It is shown that for any smooth curve on $\mathcal{P}_2$ passing q, there a.e.-uniquely exists a vector field in the above subspace such that it induces the curve around q (Villani (2008), Thm. 13.8; Ambrosio et al. (2008), Thm. 8.3.1, Prop. 8.4.5). So we can take the above subspace as the tangent space $T_q\mathcal{P}_2$, and the unique vector field in it (a representor) as the tangent vector of the curve at q (Ambrosio et al. (2008), Def. 8.4.1). Recalling the construction, once we have the tangent vector V at a point q on a curve, we can simulate the curve locally around q up to first order by updating the particles of q with V in its vector field form (Ambrosio et al. (2008), Prop. 8.4.6). A Riemannian structure can be defined by endowing the tangent space $T_q\mathcal{P}_2$ with the inner product of $\mathcal{L}_q^2$: $\langle U, V\rangle_{T_q\mathcal{P}_2} := \mathbb{E}_q[U \cdot V]$ (Otto, 2001; Benamou & Brenier, 2000). It is consistent with the well-known Wasserstein distance as it induces the same distance (Benamou & Brenier, 2000).

[1] It has a more basic definition as an auto-parallel curve under an affine connection. A Riemannian structure determines an affine connection, and the two definitions coincide on complete Riemannian manifolds.

4. Variance Reduction for Particle-Based Variational Inference Methods (ParVIs)

4.1. Variance Reduction Framework for ParVIs

Particle-based variational inference methods (ParVIs) approximate the posterior distribution p by driving the variational distribution q (i.e., the approximator) to p, which is typically done by minimizing the KL divergence to p. To do this, ParVIs optimize $\mathrm{KL}_p(q) := \mathbb{E}_q[\log q/p]$ on the Wasserstein space $\mathcal{P}_2$ by simulating its gradient flow, which is the set of curves that are tangent to the gradient of KL_p everywhere on $\mathcal{P}_2$.

Given a dataset of N data points, let $p_0(x)$ be the prior, and let $p_n(x) := p(D_n|x)$ be the likelihood term for data point D_n. The KL divergence can be decomposed as a summation:

$$\mathrm{KL}_p(q) = \mathbb{E}_q[\log q] - \mathbb{E}_q[\log p_0] - \sum_{n=1}^N \mathbb{E}_q[\log p_n] + c,$$

where $c := \log Z$ is the logarithm of the intractable normalization constant. With the above mentioned Riemannian structure of the Wasserstein space $\mathcal{P}_2$, the gradient of the KL divergence on $\mathcal{P}_2$ can be expressed (Villani (2008), Thm. 23.18; Ambrosio et al. (2008), Example 11.1.2) as the following $\mathbb{R}^m$ vector field:

$$-\operatorname{grad}\mathrm{KL}_p(q) = \overbrace{\nabla \log p_0 - \nabla \log q}^{U(q)} + \overbrace{\sum_{n=1}^N \underbrace{\nabla \log p_n}_{V_n(q)}}^{V(q)}.$$

The gradient flow simulation can be done by successively updating particles using an estimate of this vector field.

For SGD, the stochastic gradient at iteration k is given by $U(q_k) + NV_{n_k}(q_k)$ for a uniformly randomly chosen data

Algorithm 1 Stochastic Variance Reduced Gradient (SVRG) for ParVIs

Require: Initial particles $\{x_0^{(j)}\}_{j=1}^M$, target distribution $p_0(x)\prod_{n=1}^N p_n(x)$, update period T_s, learning rate ε.

Require: Vector field estimators $\hat{U}(\{x^{(j)}\}_j)^{(i)}$ and $\hat{V}_n(\{x^{(j)}\}_j)^{(i)}$, parallel transport estimator $\hat{\Gamma}^{\{y^{(j)}\}_j}_{\{x^{(j)}\}_j}\left(\{V^{(j)}\}_j\right)^{(i)}$.

Initialize $\tilde{x}^{(i)} \leftarrow x_0^{(i)}$ for $i = 1, \cdots, M$.
for $s = 1, 2, 3, \cdots$ **do**
  Let $x_0^{(i)} \leftarrow \tilde{x}^{(i)}$ for $i = 1, \cdots, M$.
  Let $\tilde{V}^{(i)} \leftarrow \sum_{n=1}^N \hat{V}_n(\{\tilde{x}^{(j)}\}_j)^{(i)}$ for $i = 1, \cdots, M$.
  for $k = 0, \cdots, T_s - 1$ **do**
    Uniformly randomly draw a data point $n_k \in \{1, \cdots, N\}$.
    Let $W_k^{(i)} \leftarrow \hat{U}(\{x_k^{(j)}\}_j)^{(i)} + N\hat{V}_{n_k}(\{x_k^{(j)}\}_j)^{(i)} - \hat{\Gamma}^{\{x_k^{(j)}\}_j}_{\{\tilde{x}^{(j)}\}_j}\left(\left\{N\hat{V}_{n_k}(\{\tilde{x}^{(j')}\}_{j'})^{(j)} - \tilde{V}^{(j)}\right\}_j\right)^{(i)}$ for $i = 1, \cdots, M$.
    Let $x_{k+1}^{(i)} \leftarrow x_k^{(i)} + \varepsilon W_k^{(i)}$ for $i = 1, \cdots, M$.
  end for
  Let $\tilde{x}^{(i)} \leftarrow x_{T_s}^{(i)}$ for $i = 1, \cdots, M$.
end for
return $\{\tilde{x}^{(i)}\}_{i=1}^M$.

point $n_k \in \{1, \cdots, N\}$. This stochastic gradient has high variance since SGD or mini-batch SGD approximates the full gradient using a single or small mini-batch of examples.

We propose applying Riemannian SVRG (Zhang et al., 2016) to reduce the variance of SGD. The main idea of SVRG is to maintain a reference snapshot position and a corresponding reference snapshot full-gradient. In every iteration, the stochastic gradient at the snapshot position minus the snapshot full-gradient is used in the update rule as a variance reduction term. The reference snapshot position and full-gradient are periodically updated at the start of every outer loop and subsequently used in every iteration in the inner loop.

We now consider the derivation of SVRG for ParVIs. According to Riemannian SVRG (Zhang et al., 2016), at the start of every outer loop, the current position is recorded as a reference snapshot position $\tilde{q}$, and the corresponding full-summation over the entire dataset is computed and stored: $\tilde{V} := V(\tilde{q})$. In each subsequent iteration k, the stochastic gradient at the current position q_k is combined with the stochastic gradient at the snapshot position $\tilde{q}$ and the stored full gradient $\tilde{V}$ to get the variance-reduced gradient. Concretely, in a usual update step k, for a uniformly randomly chosen data point $n_k \in \{1, \cdots, N\}$, the update direction is calculated by $W_k := U(q_k) + NV_{n_k}(q_k) - \Gamma_{\tilde{q}}^{q_k}\left(NV_{n_k}(\tilde{q}) - \tilde{V}\right)$, which is then used to update the position: $q_{k+1} := \mathrm{Exp}_{q_k}(\varepsilon W_k)$. Compared to SGD, we propose adding the term $-\Gamma_{\tilde{q}}^{q_k}\left(NV_{n_k}(\tilde{q}) - \tilde{V}\right)$ to the update rule, which leads to a reduction in variance.

Let $\{\tilde{x}^{(i)}\}_{i=1}^M$ and $\{x_k^{(i)}\}_{i=1}^M$ be the sets of samples (particles) of $\tilde{q}$ and q_k, respectively. Then in step k, with n_k chosen, calculate $W_k^{(i)} := W_k(x_k^{(i)}) = U(q_k)(x_k^{(i)}) + NV_{n_k}(q_k)(x_k^{(i)}) - \Gamma_{\tilde{q}}^{q_k}\left(NV_{n_k}(\tilde{q}) - \tilde{V}\right)(x_k^{(i)})$ and update the particles: $x_{k+1}^{(i)} = x_k^{(i)} + \varepsilon W_k^{(i)}$.

To implement the algorithm, we estimate U and V by ParVIs, which provide various implementations of $\hat{U}(\{x^{(j)}\}_j)^{(i)}$ and $\hat{V}_n(\{x^{(j)}\}_j)^{(i)}$ that approximate $U(q)(x^{(i)})$ and $V_n(q)(x^{(i)})$ respectively ($\{x^{(j)}\}_j$ is a set of particles of q). Let r be another distribution with particles $\{y^{(j)}\}_{j=1}^M$, then the parallel transport $\Gamma_q^r(V)(y^{(i)})$ (here V is a general tangent vector at q acting as the operand of the parallel transport) can also be estimated by the particles, and we write the estimator as $\hat{\Gamma}^{\{y^{(j)}\}_j}_{\{x^{(j)}\}_j}\left(\{V^{(j)}\}_j\right)^{(i)}$, where $V^{(j)} := V(x^{(j)})$. The SVRG for ParVIs algorithm is presented in Algorithm 1, where the estimators $\hat{U}(\{x^{(j)}\}_j)^{(i)}$, $\hat{V}_n(\{x^{(j)}\}_j)^{(i)}$ for the vector field and $\hat{\Gamma}^{\{y^{(j)}\}_j}_{\{x^{(j)}\}_j}\left(\{V^{(j)}\}_j\right)^{(i)}$ for the parallel transport are detailed below.

4.2. Estimators for the Vector Field

Since ParVI methods are derived for a particle-based numerical approximation of the Wasserstein gradient $\mathrm{grad}\,\mathrm{KL}_p(q)$ in the vector field form, we leverage these different ways of approximation to derive respective estimators for our vector fields U and V_n.

SVGD (Liu & Wang, 2016). According to Liu et al. (2019), SVGD approximates a vector field (element of $T_q\mathcal{P}_2 \subset \mathcal{L}_q^2$) by its projection onto the vector-valued reproducing kernel Hilbert space (RKHS) $\mathcal{H}^m$ of a kernel K. Adopting this

notion, we get the vector field estimators based on SVGD:

$$\hat{V}_n(\{x^{(j)}\}_j)^{(i)} = \max_{W\in\mathcal{H}^m,\|W\|_{\mathcal{H}^m}=1} \cdot \operatorname{argmax} \langle V_n(q), W\rangle_{\mathcal{L}^2_q} = \frac{1}{M}\sum_j \hat{K}_{ij}\nabla \log p_n(x^{(j)}),$$

$$\hat{U}(\{x^{(j)}\}_j)^{(i)} = \max_{W\in\mathcal{H}^m,\|W\|_{\mathcal{H}^m}=1} \cdot \operatorname{argmax} \langle U(q), W\rangle_{\mathcal{L}^2_q} = \frac{1}{M}\sum_j \left(\hat{K}_{ij}\nabla \log p_0(x^{(j)}) + \nabla_{x^{(j)}}\hat{K}_{ij}\right),$$

where $\hat{K}_{ij} := K(x^{(i)}, x^{(j)})$, and "$\max\cdot\operatorname{argmax}$" scalar-multiplies the maximizer with the maximum. The kernel averaged gradient of the log density drives the particles towards high probability regions of p, while the other term is a repulsive force between the particles; these two forces balance each other so that the particles approximate p.

Blob (Chen et al., 2018a). The Blob method uses a variational formulation of the gradient, by reformulating $-\nabla \log q$ as $\nabla(-\frac{\delta}{\delta q}\mathbb{E}_q[\log q])$, and approximates q with a smoothed density $\tilde{q} := \hat{q} * K$, where $\hat{q}$ denotes the empirical distribution of the particles and "$*$" denotes convolution. The estimators are $\hat{V}_n(\{x^{(j)}\})^{(i)} = \nabla \log p_n(x^{(i)})$, $\hat{U}(\{x^{(j)}\})^{(i)} = \nabla \log p_0(x^{(i)}) - \frac{\sum_k \nabla_{x^{(i)}}\hat{K}_{ik}}{\sum_j \hat{K}_{ij}} - \sum_k \frac{\nabla_{x^{(i)}}\hat{K}_{ik}}{\sum_j \hat{K}_{jk}}$.

GFSD (Liu et al., 2019). GFSD directly approximates q in $-\nabla \log q$ with the smoothed density $\tilde{q}$. The estimators are $\hat{V}_n(\{x^{(j)}\})^{(i)} = \nabla \log p_n(x^{(i)})$ and $\hat{U}(\{x^{(j)}\})^{(i)} = \nabla \log p_0(x^{(i)}) - \frac{\sum_k \nabla_{x^{(i)}}\hat{K}_{ik}}{\sum_j \hat{K}_{ij}}$.

GFSF (Liu et al., 2019). GFSF identifies $-\nabla \log q$ as the solution of an optimization problem, and then defines an estimator as the solution of a modified problem by taking q as $\hat{q}$ and using an RKHS as the optimization domain. The estimators are $\hat{V}_n(\{x^{(j)}\})^{(i)} = \nabla \log p_n(x^{(i)})$ and $\hat{U}(\{x^{(j)}\})^{(i)} = \nabla \log p_0(x^{(i)}) + \sum_k \hat{K}^{-1}_{ik}\sum_j \nabla_{x^{(j)}}\hat{K}_{jk}$.

4.3. Estimators for the Parallel Transport

Schild's ladder estimator. The Schild's ladder method (Ehlers et al., 1972; Kheyfets et al., 2000) constructs a first order approximation to the parallel transport using the exponential map and its inverse on the manifold: $\Gamma^r_q(V) \approx \operatorname{Exp}_r^{-1}\Big(\operatorname{Exp}_q\Big(2\operatorname{Exp}_q^{-1}\Big(\operatorname{Exp}_{\operatorname{Exp}_q(V)}\big(\tfrac{1}{2}\operatorname{Exp}^{-1}_{\operatorname{Exp}_q(V)}(r)\big)\Big)\Big)\Big)$. It is known (Villani (2008), Coro. 7.22; Ambrosio et al. (2008), Prop. 8.4.6; Erbar et al. (2010), Prop. 2.1) that $\operatorname{Exp}_q(V) = (\mathrm{id}+V)_{\#}q$ for absolutely continuous q, which means that if $\{x^{(i)}\}_i$ is a set of samples of q, then $\{x^{(i)} + V(x^{(i)})\}_i$ is a set of samples of $\operatorname{Exp}_q(V)$. The inverse exponential map $\operatorname{Exp}_q^{-1}(r)$ (with q absolutely continuous) can be expressed by the optimal transport map $\mathcal{T}^r_q$ from q to r: $\operatorname{Exp}_q^{-1}(r) = \mathcal{T}^r_q - \mathrm{id}$ (Ambrosio et al. (2008), Prop. 8.4.6). In practice, $\mathcal{T}^r_q$ can be estimated by the discrete optimal transport map from the samples $\{x^{(i)}\}_i$ of q to the samples $\{y^{(i)}\}_i$ of r, which can be done by exact methods (e.g., Pele & Werman (2009)) or faster approximate methods like the Sinkhorn methods (Cuturi, 2013; Xie et al., 2018). Applying these operations on samples, we get an implementation of $\hat{\Gamma}^{\{y^{(j)}\}_j}_{\{x^{(j)}\}_j}\left(\{V^{(j)}\}_j\right)^{(i)}$.

Pairwise-close estimator. Liu et al. (2019) consider the case where $\{x^{(j)}\}_j$ and $\{y^{(j)}\}_j$ are pairwise close, i.e., $d(x^{(i)}, y^{(i)}) \ll \min\left\{\min_{j\neq i} d(x^{(i)}, x^{(j)}), \min_{j\neq i} d(y^{(i)}, y^{(j)})\right\}$. Under this condition, the discrete optimal transport map can be approximated by $\mathcal{T}^r_q(x^{(i)}) \approx y^{(i)} - x^{(i)}$, and the above parallel transport estimator simplifies to:

$$\hat{\Gamma}^{\{y^{(j)}\}_j}_{\{x^{(j)}\}_j}\left(\{V^{(j)}\}_j\right)^{(i)} = V^{(i)}.$$

In our experiments, we use the pairwise-close estimator, which we observed works well empirically. The pairwise-close version simplifies the algorithm and computation.

4.4. SPIDER for ParVIs

We propose applying Riemannian SPIDER (Stochastic Path Integrated Differential Estimator) (Zhou et al., 2019) as another variance reduction method for ParVIs. SPIDER for ParVIs uses a recursive equation to estimate the full gradient along the trajectory and employs normalized gradient updates. In contrast to SVRG, SPIDER only relies on the previous position instead of a reference snapshot position for variance reduction. Instead of using a reference snapshot full-gradient, SPIDER uses the estimate of the full gradient at the previous position.

At the start of every outer loop, we first compute a full gradient $\{W_0^{(j)}\}_{j=1}^M$ over the entire dataset and then apply normalized gradient ascent. In each of the subsequent iterations $k \geq 1$, the stochastic gradient at the current particle position $\{x_k^{(j)}\}_j$ is combined with the stochastic gradient at the previous particle position $\{x_{k-1}^{(j)}\}_j$ and the previous estimate of the full gradient $\{W_{k-1}^{(j)}\}_j$ to get the new estimate of the full gradient $\{W_k^{(j)}\}_j$. The particle positions are updated with normalized gradient ascent using the formula $x_{k+1}^{(i)} \leftarrow x_k^{(i)} + \varepsilon W_k^{(i)}/\|W_k\|$, where $\|W_k\|^2 = \frac{1}{M}\sum_{j=1}^M \|W_k^{(j)}\|^2$ is the discretization of the $\mathcal{L}^2_q$ norm. For space reasons, the SPIDER for ParVIs algorithm is presented in the supplement.

4.5. Stochastic Quasi-Newton with Variance Reduction (SQN-VR) for ParVIs

Algorithm 2 Stochastic Quasi-Newton with Variance Reduction (SQN-VR) for ParVIs (simplified under pairwise-close approximation)

Require: Initial particles $\{x_0^{(j)}\}_{j=1}^M$, target distribution $p_0(x)\prod_{n=1}^N p_n(x)$, number of epochs S, update period T_s, learning rates $\varepsilon_1, \varepsilon_2$, L-BFGS memory size.
Require: Vector field estimators $\hat{U}(\{x^{(j)}\}_j)^{(i)}$ and $\hat{V}_n(\{x^{(j)}\}_j)^{(i)}$.
Initialize $\tilde{x}_1^{(i)} \leftarrow x_0^{(i)}$ for $i = 1, \cdots, M$.
Let $\tilde{V}_1^{(i)} \leftarrow \sum_{n=1}^N \hat{V}_n(\{\tilde{x}_1^{(j)}\}_j)^{(i)}$ for $i = 1, \cdots, M$.
for $s = 1, 2, 3, \cdots, S$ **do**
  Let $x_0^{(i)} \leftarrow \tilde{x}_s^{(i)}$ for $i = 1, \cdots, M$.
  for $k = 0, \cdots, T_s - 1$ **do**
    Sample a data point $n_k \in \{1, \cdots, N\}$.
    Let $W_k^{(i)} = \hat{U}(\{x_k^{(j)}\}_j)^{(i)} + N\hat{V}_{n_k}(\{x_k^{(j)}\}_j)^{(i)} - \left(N\hat{V}_{n_k}(\{\tilde{x}_s^{(j')}\}_{j'})^{(i)} - \tilde{V}_s^{(i)}\right)$ for $i = 1, \cdots, M$.
    if $s > 2$ **then**
      Compute the quasi-Newton update $[Z_k^{(j)}]_{j=1}^M$ from $[W_k^{(j)}]_{j=1}^M$ by L-BFGS two-loop recursion.
      Let $x_{k+1}^{(i)} \leftarrow x_k^{(i)} - \varepsilon_2 Z_k^{(i)}$ for $i = 1, \cdots, M$.
    else
      Let $x_{k+1}^{(i)} \leftarrow x_k^{(i)} + \varepsilon_1 W_k^{(i)}$ for $i = 1, \cdots, M$.
    end if
  end for
  Let $\tilde{x}_{s+1}^{(i)} \leftarrow x_{T_s}^{(i)}$ for $i = 1, \cdots, M$.
  Let $\tilde{V}_{s+1}^{(i)} \leftarrow \sum_{n=1}^N \hat{V}_n(\{\tilde{x}_{s+1}^{(j)}\}_j)^{(i)}$ for $i = 1, \cdots, M$.
  Let $S_{s+1}^{(i)} \leftarrow \tilde{x}_{s+1}^{(i)} - \tilde{x}_s^{(i)}$ for $i = 1, \cdots, M$.
  Let $Y_{s+1}^{(i)} \leftarrow \hat{U}(\{\tilde{x}_{s+1}^{(j)}\}_j)^{(i)} + \tilde{V}_{s+1}^{(i)} - \hat{U}(\{\tilde{x}_s^{(j)}\}_j)^{(i)} - \tilde{V}_s^{(i)}$ for $i = 1, \cdots, M$.
  Store the L-BFGS pair $([S_{s+1}^{(j)}]_{j=1}^M, [Y_{s+1}^{(j)}]_{j=1}^M)$, and discard the oldest pair if the memory size is exceeded.
end for
return $\{\tilde{x}_S^{(i)}\}_{i=1}^M$.

To address ill-conditioned Bayesian inference problems, we further incorporate quasi-Newton preconditioning techniques. Ill-conditioned problems are typically identified by an ill-conditioned Hessian of the objective, which makes the function landscape distorted along a certain direction. (Quasi-)Newton preconditioning works by stretching the optimization space to make the landscape more isotropic, resulting in longer-sighted updating direction. In the context of Bayesian inference, we depict the ill-conditionedness accordingly by the Hessian of the KL divergence on $\mathcal{P}_2$, which is now a quadratic form in the tangent space that generalizes the matrix form to the infinite-dimensional manifold. According to Example 15.9 of Villani (2008), the Hessian operator takes the form $\big(\mathrm{Hess}\,\mathrm{KL}_p(q)\big)[V] =$

$$\mathbb{E}_{q(x)}\big[\|\nabla V(x)\|_F^2 - V(x)^\top\big(\nabla\nabla^\top \log p(x)\big)V(x)\big] \quad (1)$$

for Euclidean support space, so its ill-conditionedness is related to that of the Hessian matrix $\nabla\nabla^\top \log p(x)$.

We propose applying Riemannian Stochastic Quasi-Newton with Variance Reduction (SQN-VR) (Kasai et al., 2018) to ParVIs. SQN-VR builds on SVRG by leveraging curvature information to speed up convergence on ill-conditioned problems. Like SVRG, SQN-VR computes the variance-reduced stochastic gradient at every iteration. In SQN-VR, an approximation to the inverse Hessian is computed and applied to the variance-reduced stochastic gradient to get the final update direction.

We present the SQN-VR for ParVIs algorithm in Algorithm 2 under the pairwise-close assumption. Similarly to SVRG, SQN-VR updates the snapshot position and the corresponding full-gradient once in every outer loop. In addition, the curvature pair of the QN method is updated once in every outer loop, using the difference between the current and previous snapshot positions and the difference between their corresponding full gradients computed for VR.

In the first two outer loops of SQN-VR, before two curvature pairs have been collected, the SQN-VR update rule each iteration is the same as the SVRG update rule. After two curvature pairs are collected, we apply a quasi-Newton update every iteration instead of directly applying the variance reduced gradient. Specifically, we use the L-BFGS two-loop recursion (Nocedal & Wright, 2006; Kasai et al., 2018) with the previous L curvature pairs to apply the inverse Hessian approximation operator to the variance reduced gradient to get our quasi-Newton update direction.

5. Experimental results

We present experimental results on Bayesian linear regression and logistic regression. We first describe the experimental setup that we use. For the choice of ParVI, we use SVGD with the linear kernel $k(\mathbf{x}, \mathbf{x}') = \frac{1}{d+1}(\mathbf{x}^T\mathbf{x}' + 1)$, where d is the dimension and with mean centering of the particles, which has been proven to yield exact estimation of the mean and covariance for Gaussian target distributions (Liu & Wang, 2018). We use 100 particles and a batch size of 10 in all of our experiments. We initialize the particles from a standard Gaussian, corresponding to the prior.

We compare the following optimization algorithms: AdaGrad with momentum, SGD, SVRG, SPIDER, and SQN-VR. We note that AdaGrad is not a principled Riemmanian optimization algorithm, but we include the Euclidean version of AdaGrad as an empirical algorithm because AdaGrad has been used in several SVGD papers. For every optimizer, we tune the learning rate by running a grid search over $\bigcup_{k=-1}^{2}\{\frac{10^k}{N}, \frac{3\times10^k}{N}\}$ where N is the number of data points. For AdaGrad, we additionally tune the learning rate in $\bigcup_{k=3}^{5}\{\frac{10^k}{N}, \frac{3\times10^k}{N}\}$, $\alpha \in \{0.9, 0.95, 0.99, 0.999\}$ and the fudge factor $\epsilon \in \bigcup_{k=4}^{8}\{10^{-k}\}$. For SGD, we decay the learning rate after each epoch according to the formula $\epsilon_t = a/(t+b)^\beta$ where the power $\beta \in \{0.55, 0.75, 0.95\}$ and the constants a and b are chosen so that the total learning rate decay over the total number of epochs is in $\{1, 3, 10, 30, 100, 300, 1000\}$. For SVRG and SPIDER, we use a constant learning rate for the first half of the run and decay the learning rate in the second half by a factor in $\{1, 3, 10, 30, 100, 300, 1000\}$. For SQN-VR, we use a constant learning rate in $\bigcup_{k=-5}^{0}\{10^k, 3\times10^k\}$ for the quasi-Newton updates and a memory size of 10. For all of the variance reduction methods, we update the full gradient over the entire dataset after each epoch, and we first run 10 epochs of SGD. The grid search for each optimizer consists of all combinations of learning rates and any additional optimizer-specific hyperparameters. To ensure a fair comparison in all of our results, the x-axis in our figures is the number of passes over the dataset, specifically the number of data point gradient evaluations for all of the particles divided by the dataset size N. For each dataset, we ensure that every algorithm is initialized with the same starting positions for the particles and uses the same sequence of training examples throughout.

To evaluate how well the particles approximate the posterior, we consider several metrics related to sample quality. For each of the Bayesian linear regression and logistic regression problems, we first obtain a ground truth set of 40,000 MCMC samples from a long run of No U-Turn Sampler (NUTS) (Hoffman & Gelman, 2014). Specifically, we use the implementation of NUTS in PyStan (Carpenter et al., 2017) with a dense mass matrix, and we run 16 chains of NUTS with 500 burn-in iterations and 2,500 estimation iterations each. Given this reference set of samples, our first metric is Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) between the 100 ParVI particles and the 40,000 MCMC samples. We use an RBF kernel for MMD with the kernel bandwidth equal to the median of the pairwise distances between the MCMC samples. From the ground-truth MCMC samples, we can calculate a ground truth posterior mean vector μ and covariance matrix Σ. For a ParVI algorithm, let $\hat{\mu}_t$ be the sample mean and $\hat{\Sigma}_t$ be the sample covariance of the set of particles at iteration t. We define the mean squared error (MSE) for a ParVI with respect to μ as $\frac{1}{d}\|\hat{\mu}_t - \mu\|_2^2$ and with respect to Σ as $\frac{1}{d^2}\|\hat{\Sigma}_t - \Sigma\|_F^2$. Finally, our last metric is kernel Stein discrepancy (KSD) (Chwialkowski et al., 2016; Liu et al., 2016) for the 100 ParVI particles with respect to the posterior distribution p specified by $\nabla \log p$. We evaluate KSD using the IMQ kernel proposed by Gorham & Mackey (2017), which has been proven to detect convergence and non-convergence of a sequence of samples for certain target distributions.

For each optimizer, we run a grid search over all of the optimizer hyperparameters and choose the hyperparameters that achieve the minimum MMD at the end of the run as the best-performing hyperparameters to show in our results.

For each of the datasets, we report the number of data points N, the dimensionality D, and the condition number of the posterior covariance matrix Σ, cond(Σ). Note that cond(Σ) is a computable, heuristic approximation of the Hessian of the KL on the Wasserstein space. Equation (1) gives an explicit relationship between the Hessian of the KL on the Wasserstein space and the Hessian of the log-density of the target posterior. For Bayesian linear regression, the posterior is Gaussian, so the Hessian of the log-posterior is the negative inverse posterior covariance matrix, which has the same condition number as the posterior covariance matrix that we report. For Bayesian logistic regression, the posterior can be approximated well by a Gaussian.

Bayesian linear regression We first consider a Bayesian linear regression model where the prior on the regression coefficients is standard Gaussian. We run experiments on 8 UCI regression datasets (Dua & Graff, 2019). For space reasons, we show results for 3 of the datasets in Fig. 1 and present all of the results in the supplement.

In Fig. 1(a), we show results for the noise dataset, which is a small dataset with 1,503 examples, a low dimensionality of 6, and a low posterior covariance matrix condition number of 12. After running 100 epochs of each optimizer with extensive tuning, we see that the best-performing Adagrad and SGD optimizers achieve a MMD of around $10^{-0.85}$. In contrast, all of the variance reduction algorithms achieve a MMD of $10^{-1.38}$ to $10^{-1.63}$. The variance reduction al-

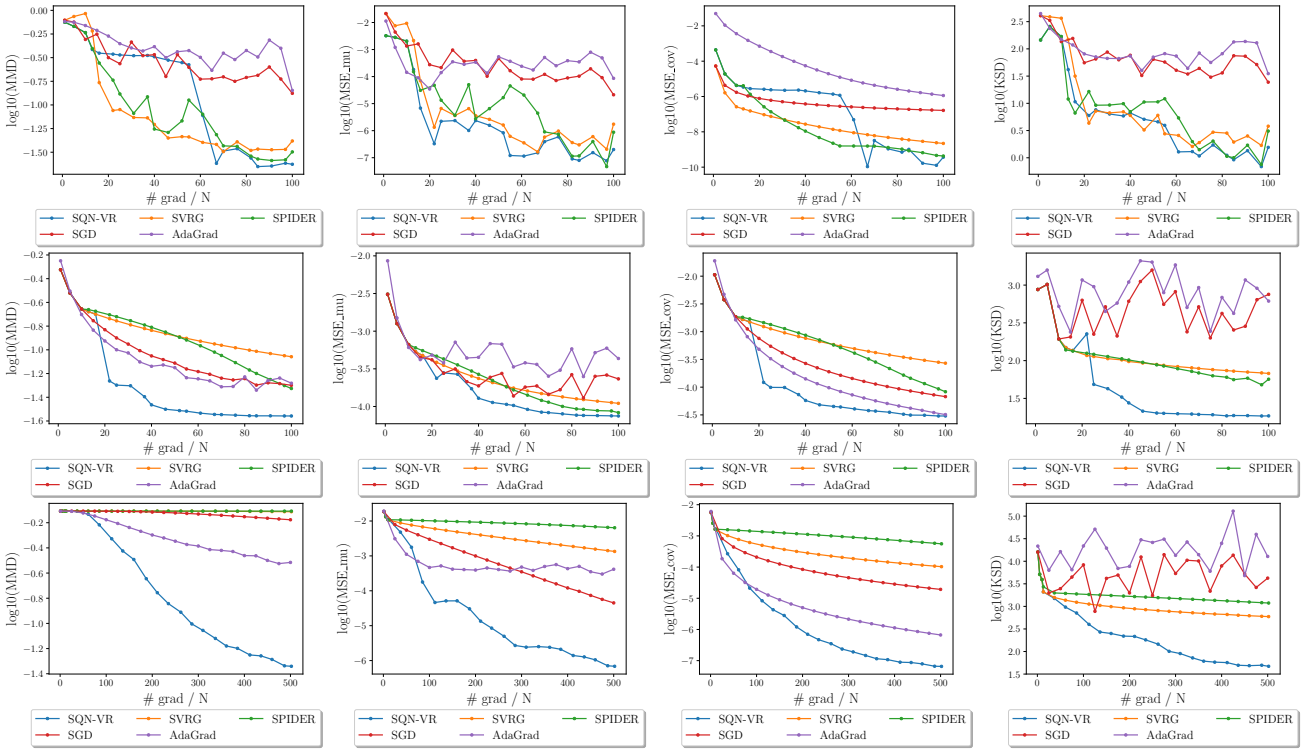

Figure 1. Experimental results for Bayesian linear regression, from top to bottom: (a) noise (b) parkinson (c) toms.

gorithms also achieve a much lower MSE than AdaGrad and SGD for estimating μ ($10^{-5.76}$ to $10^{-6.70}$ compared to $10^{-4.06}$ to $10^{-4.67}$) and Σ ($10^{-8.66}$ to $10^{-9.43}$ compared to $10^{-5.95}$ to $10^{-6.79}$), resulting in much more accurate posterior mean and covariance estimates. The KSD metric provides further evidence that the variance reduction algorithms produce particles with much higher sample quality.

In Figs. 1(b) and 1(c), we consider two more challenging datasets with significantly higher posterior covariance matrix condition numbers. In Fig. 1(b), for the parkinson dataset with N = 5875, D = 21, cond(Σ)=65697, we see that SQN-VR performs the best after 100 epochs with a MMD of $10^{-1.56}$; Adagrad, SGD, and SPIDER achieve a MMD of around $10^{-1.3}$, and SVRG achieves a MMD of around $10^{-1.06}$. Thus, we see that variance reduction alone might not improve over well-tuned SGD for ill-conditioned problems. If we compare Adagrad, SGD, and SPIDER in terms of MSE for μ and Σ, we notice that SPIDER performs the best for estimating μ and the worst for estimating Σ, Adagrad performs the best at estimating Σ and the worst for estimating μ, and SGD is in between. While these 3 methods produce particles with similar MMD, the distributions of the particles are very different, reflected in the differing estimates for μ and Σ. Interestingly, the KSD metric suggests that the particles from SGD and Adagrad have worse quality than the particles from SPIDER and SVRG; this might be due to the less stable optimization procedure. In Fig. 1(c), we present results for the toms dataset which has 28,179 data points, a high dimensionality of 97, and a high posterior covariance matrix condition number of 45,923. After 500 epochs, the best-performing SQN-VR achieves a MMD of $10^{-1.34}$, Adagrad achieves a MMD of $10^{-0.52}$, and the other optimizers achieve a MMD no better than $10^{-0.18}$. For this ill-conditioned problem, we see that SQN-VR is essential for fast convergence and accurate posterior inference.

Bayesian logistic regression We consider a Bayesian logistic regression model for binary classification where the prior on the regression coefficients is standard Gaussian. We run 8 Bayesian logistic regression experiments. For space reasons, we show results for MNIST and covtype in Fig. 2 and present all of the results in the supplement.

Our MNIST (LeCun et al., 1998) binary classification problem is classifying digits 7 vs. 9 after applying PCA to reduce the dimension of the image to 50, similar to Korattikara et al. (2014). The MNIST dataset has 12,214 training examples and a low posterior covariance matrix condition number of 58. In Fig. 2(a), we see that all of the variance reduction algorithms perform well, achieving a MMD of around $10^{-1.85}$. In contrast, the best-performing AdaGrad achieves a MMD of $10^{-0.81}$ and SGD achieves a MMD of $10^{-1.01}$.

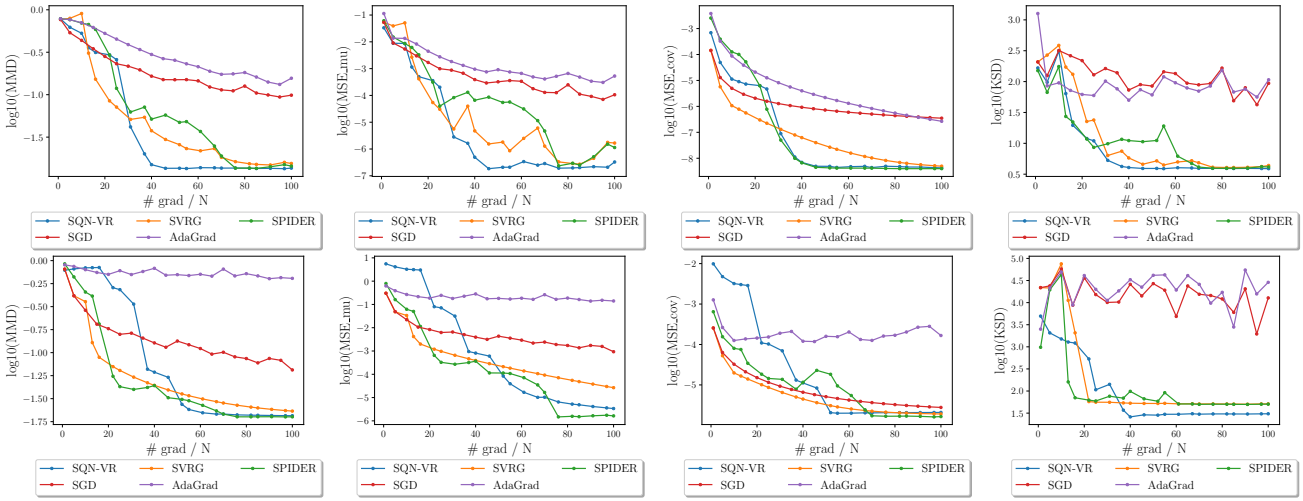

Figure 2. Experimental results for Bayesian logistic regression, from top to bottom: (a) mnist (b) covtype.

The covtype dataset has 464,809 examples (using a 80% training split), 55 dimensions, and a high posterior covariance matrix condition number of 341,266. In Fig. 2(b), we see that SQN-VR and SPIDER perform the best, achieving a MMD of around $10^{-1.7}$, with SVRG following close behind with a MMD of around $10^{-1.64}$. Without variance reduction, the best-performing SGD achieves a MMD of $10^{-1.19}$ and AdaGrad achieves a MMD of $10^{-0.19}$. Looking at the other metrics, we observe that the particles from SGD approximate Σ well but approximate μ poorly and are also worse in terms of KSD. Thus, we see that variance reduction techniques can greatly accelerate the convergence of ParVIs for real-world datasets of varying size.

6. Discussion

Our experimental results on Bayesian linear regression and logistic regression demonstrate that existing ParVI approaches using stochastic gradients converge insufficiently fast and that variance reduction and quasi-Newton methods can greatly accelerate the convergence of ParVIs for accurate posterior inference in large-scale and ill-conditioned problems. While using variance reduction techniques alone sped up convergence in many large-scale problems, combining variance reduction and quasi-Newton techniques led to significantly faster convergence in several cases and the best performance on every dataset we considered. Our algorithms are applicable to general ParVIs and are based on principled Riemannian optimization algorithms.

From the perspective of posterior inference, our new methods produced a set of particles with significantly better sample quality, as measured by MMD and KSD, and better estimates of posterior expectations, such as mean and covariance. Accurate posterior inference requires solving the ParVI optimization problem to a high degree of accuracy, so leveraging Riemannian optimization methods with fast convergence and high accuracy is very important. While we did a large grid search to tune the hyperparameters, additional tuning of the hyperparameters and other techniques, such as adaptive learning rates and mini-batch sizes, could further improve the performance of the optimization algorithms.

In our experiments, we assumed the pairwise close condition, which we observed works well empirically. Under this assumption, our methods are simple, easy to use, fast in terms of running time, and work well in practice. In our experiments, we observed that the running times of our methods are generally comparable to or slightly faster than SGD and AdaGrad given the same number of gradient evaluations. The relative order of running times was generally SVRG $\leq$ SPIDER $\leq$ SQN-VR $\leq$ SGD $\leq$ AdaGrad. For example, on an Intel Xeon E5-2640v3, 100 epochs on the covtype dataset took 20 minutes for SVRG and SPIDER, 22 for SQN-VR, 24 for SGD, and 28 for AdaGrad.

We focused our experiments on Bayesian linear regression and logistic regression, running SVGD with a linear kernel, which works well for Gaussian-like posteriors. In this setting, we observed that ParVI methods can be highly accurate for estimating posterior expectations and producing a small set of particles which represent the posterior while being fast in terms of running time. As an example, on the challenging covtype dataset, our ParVI implementation took 22 minutes while 500 burn-in iterations of NUTS took 4.5 hours. Using a subset of 100 NUTS samples also gives a very poor representation of the posterior. Future work involves studying how well various ParVI methods approximate various posterior distributions under sample quality metrics to further improve ParVI methods for real-world Bayesian inference problems.

Acknowledgement

J.Z was supported by the National Key Research and Development Program of China (No. 2017YFA0700904), NSFC Projects (Nos. 61620106010), Beijing Academy of Artificial Intelligence (BAAI), Tsinghua-Huawei Joint Research Program, and a grant from Tsinghua Institute for Guo Qiang.

References

Abraham, R., Marsden, J. E., and Ratiu, T. *Manifolds, tensor analysis, and applications*, volume 75. Springer Science & Business Media, New York, 2012.

Ambrosio, L., Gigli, N., and Savaré, G. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008.

Benamou, J.-D. and Brenier, Y. A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. *Handbook of Markov Chain Monte Carlo*. CRC Press, 2011.

Byrd, R. H., Hansen, S. L., Nocedal, J., and Singer, Y. A stochastic quasi-Newton method for large-scale optimization. *SIAM Journal on Optimization*, 26(2):1008–1031, 2016.

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.

Chatterji, N., Flammarion, N., Ma, Y., Bartlett, P., and Jordan, M. On the theory of variance reduction for stochastic gradient Monte Carlo. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 764–773, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018.

Chen, C., Zhang, R., Wang, W., Li, B., and Chen, L. A unified particle-optimization framework for scalable Bayesian sampling. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI 2018)*, Monterey, California USA, 2018a. Association for Uncertainty in Artificial Intelligence.

Chen, W. Y., Mackey, L., Gorham, J., Briol, F.-X., and Oates, C. J. Stein points. *arXiv preprint arXiv:1803.10161*, 2018b.

Chwialkowski, K., Strathmann, H., and Gretton, A. A kernel test of goodness of fit. In *Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)*, pp. 2606–2615, New York, New York USA, 2016. IMLS.

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, pp. 2292–2300, Lake Tahoe, Nevada USA, 2013. NIPS Foundation.

Defazio, A., Bach, F., and Lacoste-Julien, S. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pp. 1646–1654, 2014.

Detommaso, G., Cui, T., Marzouk, Y., Spantini, A., and Scheichl, R. A Stein variational Newton method. In *Advances in Neural Information Processing Systems*, pp. 9187–9197, Montréal, Canada, 2018. NIPS Foundation.

Do Carmo, M. P. *Riemannian Geometry*. Birkhäuser, 1992.

Dua, D. and Graff, C. UCI Machine Learning Repository, 2019. URL http://archive.ics.uci.edu/ml.

Dubey, K. A., Reddi, S. J., Williamson, S. A., Poczos, B., Smola, A. J., and Xing, E. P. Variance reduction in stochastic gradient Langevin dynamics. In *Advances in neural information processing systems*, pp. 1154–1162, 2016.

Ehlers, J., Pirani, F., and Schild, A. The geometry of free fall and light propagation, in the book "General Relativity" (papers in honour of JL Synge), 63–84, 1972.

Erbar, M. et al. The heat equation on manifolds as a gradient flow in the Wasserstein space. In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, volume 46, pp. 1–23. Institut Henri Poincaré, 2010.

Feng, Y., Wang, D., and Liu, Q. Learning to draw samples with amortized Stein variational gradient descent. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI 2017)*, Sydney, Australia, 2017. Association for Uncertainty in Artificial Intelligence.

Gorham, J. and Mackey, L. Measuring sample quality with kernels. *arXiv preprint arXiv:1703.01717*, 2017.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

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

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pp. 315–323, 2013.

Kasai, H., Sato, H., and Mishra, B. Riemannian stochastic quasi-Newton algorithm with variance reduction and its convergence analysis. In *International Conference on Artificial Intelligence and Statistics*, pp. 269–278, 2018.

Kheyfets, A., Miller, W. A., and Newton, G. A. Schild's ladder parallel transport procedure for an arbitrary connection. *International Journal of Theoretical Physics*, 39 (12):2891–2898, 2000.

Korattikara, A., Chen, Y., and Welling, M. Austerity in MCMC land: Cutting the Metropolis-Hastings budget. In *International Conference on Machine Learning*, pp. 181–189, 2014.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, Z., Zhang, T., Cheng, S., Zhu, J., and Li, J. Stochastic gradient Hamiltonian Monte Carlo with variance reduction for Bayesian inference. *Machine Learning*, 108(8-9): 1701–1727, 2019.

Liu, C. and Zhu, J. Riemannian Stein variational gradient descent for Bayesian inference. In *The 32nd AAAI Conference on Artificial Intelligence*, pp. 3627–3634, New Orleans, Louisiana USA, 2018. AAAI press.

Liu, C., Zhuo, J., Cheng, P., Zhang, R., Zhu, J., and Carin, L. Understanding and accelerating particle-based variational inference. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 4082–4092, Long Beach, California USA, 2019. IMLS.

Liu, Q. Stein variational gradient descent as gradient flow. In *Advances in Neural Information Processing Systems*, pp. 3118–3126, Long Beach, California USA, 2017. NIPS Foundation.

Liu, Q. and Wang, D. Stein variational gradient descent: A general purpose Bayesian inference algorithm. In *Advances in Neural Information Processing Systems*, pp. 2370–2378, Barcelona, Spain, 2016. NIPS Foundation.

Liu, Q. and Wang, D. Stein variational gradient descent as moment matching. In *Advances in Neural Information Processing Systems 31*, pp. 8854–8863. 2018.

Liu, Q., Lee, J. D., and Jordan, M. I. A kernelized Stein discrepancy for goodness-of-fit tests. In *Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)*, New York, New York USA, 2016. IMLS.

Liu, Y., Ramachandran, P., Liu, Q., and Peng, J. Stein variational policy gradient. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI 2017)*, Sydney, Australia, 2017. Association for Uncertainty in Artificial Intelligence.

Moritz, P., Nishihara, R., and Jordan, M. A linearly-convergent stochastic L-BFGS algorithm. In *Artificial Intelligence and Statistics*, pp. 249–258, 2016.

Nocedal, J. and Wright, S. *Numerical optimization*. Springer Science & Business Media, 2006.

Otto, F. The geometry of dissipative evolution equations: the porous medium equation. 2001.

Pele, O. and Werman, M. Fast and robust earth mover's distances. In *Proceedings of the 12th International Conference on Computer Vision (ICCV-09)*, volume 9, pp. 460–467, Kyoto, Japan, 2009. IEEE.

Pu, Y., Gan, Z., Henao, R., Li, C., Han, S., and Carin, L. VAE learning via Stein variational gradient descent. In *Advances in Neural Information Processing Systems*, pp. 4239–4248, Long Beach, California USA, 2017. NIPS Foundation.

Roux, N. L., Schmidt, M., and Bach, F. R. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in neural information processing systems*, pp. 2663–2671, 2012.

Roychowdhury, A. and Parthasarathy, S. Accelerated stochastic quasi-Newton optimization on Riemann manifolds. *arXiv preprint arXiv:1704.01700*, 2017.

Villani, C. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

Wainwright, M. J., Jordan, M. I., et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

Wang, D., Tang, Z., Bajaj, C., and Liu, Q. Stein variational gradient descent with matrix-valued kernels. In *Advances in neural information processing systems*, pp. 7834–7844, 2019.

Xie, Y., Wang, X., Wang, R., and Zha, H. A fast proximal point method for computing Wasserstein distance. *arXiv preprint arXiv:1802.04307*, 2018.

Yoon, J., Kim, T., Dia, O., Kim, S., Bengio, Y., and Ahn, S. Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pp. 7343–7353, Montréal, Canada, 2018. NIPS Foundation.

Zhang, H., Reddi, S. J., and Sra, S. Riemannian SVRG: Fast stochastic optimization on Riemannian manifolds. In *Advances in Neural Information Processing Systems*, pp. 4592–4600, Barcelona, Spain, 2016. NIPS Foundation.

Zhang, J., Zhang, R., and Chen, C. Stochastic particle-optimization sampling and the non-asymptotic convergence theory. *arXiv preprint arXiv:1809.01293*, 2018a.

Zhang, J., Zhao, Y., and Chen, C. Variance reduction in stochastic particle-optimization sampling. *arXiv preprint arXiv:1811.08052*, 2018b.

Zhou, P., Yuan, X., Yan, S., and Feng, J. Faster first-order methods for stochastic non-convex optimization on Riemannian manifolds. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

Zhuo, J., Liu, C., Shi, J., Zhu, J., Chen, N., and Zhang, B. Message passing Stein variational gradient descent. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 6018–6027, 2018.