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

Michael H. Zhu¹ Chang Liu² Jun Zhu³

Algorithm A.1 SPIDER for ParVIs

Require: Initial particles \{x_{0}^{(j)}\}_{j=1}^{M}, target distribution \prod_{n=1}^{N} p_n(x), update period \( T_s \), learning rate \( \varepsilon \).

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

1: Initialize \( \tilde{x}^{(i)} \leftarrow x_{0}^{(i)} \) for \( i=1,\ldots,M \).
2: for \( s=1,2,3,\ldots \) do
3: \hspace{1em} Let \( x_{0}^{(i)} \leftarrow \tilde{x}^{(i)} \) for \( i=1,\ldots,M \).
4: \hspace{1em} for \( k=0,\ldots,T_s-1 \) do
5: \hspace{2em} if \( k=0 \) then
6: \hspace{3em} Let \( W_0^{(i)} \leftarrow \hat{U}(\{x^{(j)}\}^{(i)}_j) + \sum_{n=1}^{N} \hat{V}_n(\{x^{(j)}\}^{(i)}_j) \) for \( i=1,\ldots,M \).
7: \hspace{2em} else
8: \hspace{3em} Sample a data point \( n_k \in \{1,\ldots,N\} \) uniformly at random.
9: \hspace{3em} Let \( W_k^{(i)} \leftarrow \hat{U}(\{x^{(j)}\}^{(i)}_j) + N\hat{V}_n(\{x^{(j)}\}^{(i)}_j) - \hat{\Gamma}^{(x^{(j)}_k)}_{\{x^{(j)}_{k-1}\}^{(i)}_j} \left( \hat{U}(\{x^{(j)}_{k-1}\}^{(i)}_j) + N\hat{V}_n(\{x^{(j)}_{k-1}\}^{(i)}_j) \right) \) for \( i=1,\ldots,M \).
10: \hspace{2em} end if
11: \hspace{1em} Let \( x_{k+1}^{(i)} \leftarrow x_{k}^{(i)} + \varepsilon W_k^{(i)}/\|W_k\| \), where \( \|W_k\|^2 = \frac{1}{M} \sum_{j=1}^{M} \|W_k^{(j)}\|^2 \) for \( i=1,\ldots,M \).
12: end for
13: Let \( \tilde{x}^{(i)} \leftarrow x_{T_s}^{(i)} \) for \( i=1,\ldots,M \).
14: end for
15: return \( \tilde{x}^{(i)}_{i=1}^{M} \).

¹Department of Computer Science, Stanford University, Stanford, CA, USA ²Microsoft Research Asia, Beijing, 100080, China ³Dept. 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>, Michael H. Zhu <mhzhu@cs.stanford.edu>.

Algorithm A.2 L-BFGS two-loop recursion in SQN-VR for ParVIs

Require: Memory size $\tau$, L-BFGS pairs $\left\{ ([S_{u}^{(j)}]_{j=1}^{M}, [Y_{u}^{(j)}]_{j=1}^{M}) \right\}_{u=s-\tau+1}^{s}$, gradient $[W^{(j)}]_{j=1}^{M}$.

1: Let $Q^{(i)} \leftarrow W^{(i)}$ for $i = 1, \ldots, M$.
2: for $u = s, s-1, \ldots, s-\tau+1$ do
3: Let $\rho_{u} \leftarrow 1/\sum_{j=1}^{M} (S_{u}^{(j)})^{T} Y_{u}^{(j)}$.
4: Let $\alpha_{u} \leftarrow \rho_{u} \sum_{j=1}^{M} (S_{u}^{(j)})^{T} Q^{(j)}$.
5: Let $Q^{(i)} \leftarrow Q^{(i)} - \alpha_{u} Y_{u}^{(i)}$ for $i = 1, \ldots, M$.
6: end for
7: Let $Z^{(i)} \leftarrow \sum_{i=1}^{M} \frac{(S_{u}^{(j)})^{T} Y_{u}^{(j)}}{\sum_{j=1}^{M} (Y_{u}^{(j)})^{T} Y_{u}^{(i)}} Q^{(i)}$ for $i = 1, \ldots, M$.
8: for $u = s-\tau+1, s-\tau+2, \ldots, s$ do
9: Let $\beta \leftarrow \rho_{u} \sum_{j=1}^{M} (Y_{u}^{(j)})^{T} Z^{(j)}$.
10: Let $Z^{(i)} \leftarrow Z^{(i)} + (\alpha_{u} - \beta) S_{u}^{(j)}$ for $i = 1, \ldots, M$.
11: end for
12: return $\{ Z^{(i)} \}_{i=1}^{M}$.
Algorithm A.3 Stochastic Quasi-Newton with Variance Reduction (SQN-VR) for ParVIs

Require: Initial particles \{x_0^{(i)}\}_{i=1}^M, target distribution \Var{p_0(x) \prod_{n=1}^N p_n(x)}\, number of epochs \mathcal{S}\, update period \mathcal{T}\, learning rate \varepsilon_1, \varepsilon_2\, L-BFGS memory size, cautious update threshold \mathcal{C}.

Require: Vector field estimators \Hat{U}((\{x_0^{(j)}\}_j^{(i)})) and \Hat{V}_n((\{x_0^{(j)}\}_j^{(i)}), parallel transport estimator \Hat{\Gamma}((\{x_0^{(j)}\}_j^{(i)})^{(i)}) \text{, inverse exponential map estimator } \Exp((\{x_0^{(j)}\}_j^{(i)}))^{(i)}.

1: Initialize \Hat{x}_0^{(i)} \leftarrow x_0^{(i)} \text{ for } i = 1, \ldots, M.
2: Let \Hat{V}_1^{(i)} \leftarrow \sum_{n=1}^N \Hat{V}_n((\Hat{x}_0^{(j)}))^{(i)} \text{ for } i = 1, \ldots, M.
3: for s = 1, 2, 3, \ldots, \mathcal{S} do
4:  Let \Hat{x}_s^{(i)} \leftarrow \Hat{x}_s^{(i)} \text{ for } i = 1, \ldots, M.
5:  for k = 0, \ldots, \mathcal{T} - 1 do
6:     Sample a data point \eta_k \in \{1, \ldots, N\}.
7:     if s > 2 then
8:         Let \Hat{W}_k^{(i)} = \left(\Hat{\Gamma}((\Hat{x}_{k+1}^{(j)}))^{(i)} - \left(\Hat{U}((\Hat{x}_{k+1}^{(j)}))^{(i)} + \Hat{V}_n((\Hat{x}_{k+1}^{(j)}))^{(i)} - \Hat{\Gamma}((\Hat{x}_s^{(j)}))^{(i)} \right) \right) \text{ for } i = 1, \ldots, M.
9:     Compute the quasi-Newton update \Hat{Z}_k^{(j)} \leftarrow \Hat{W}_k^{(i)} \text{ by L-BFGS two-loop recursion.}
10:     Let \Hat{x}_{k+1}^{(i)} \leftarrow \Hat{x}_s^{(i)} - \varepsilon_1 \Hat{\Gamma}((\Hat{x}_s^{(j)})) \text{ for } i = 1, \ldots, M.
11:  else
12:     Let \Hat{W}_k^{(i)} = \Hat{U}((\Hat{x}_s^{(j)}))^{(i)} + \Hat{V}_n((\Hat{x}_s^{(j)}))^{(i)} - \Hat{\Gamma}((\Hat{x}_s^{(j)}))^{(i)} \text{ for } i = 1, \ldots, M.
13:     Let \Hat{x}_{k+1}^{(i)} \leftarrow \Hat{x}_s^{(i)} + \varepsilon_1 \Hat{W}_k^{(i)} \text{ for } i = 1, \ldots, M.
14:  end if
15: end for
16: Let \Hat{x}_{s+1}^{(i)} \leftarrow \Hat{x}_s^{(i)} \text{ for } i = 1, \ldots, M.
17: Let \Hat{V}_{s+1}^{(i)} \leftarrow \sum_{n=1}^N \Hat{V}_n((\Hat{x}_{s+1}^{(j)}))^{(i)} \text{ for } i = 1, \ldots, M.
18: Calculate the tangent vector \Hat{\eta}^{(i)} \text{ from } \Hat{x}_s^{(i)} \text{ to } \Hat{x}_{s+1}^{(i)} \text{ by } \Hat{\eta}^{(i)} = \Exp((\Hat{x}_{s+1}^{(j)}))^{(i)} \text{ for } i = 1, \ldots, M.
19: Let \Hat{S}_{s+1}^{(j)} = \left(\Hat{\eta}^{(j)}\right)^{\top} \left(\Hat{x}_{s+1}^{(j)}\right)^{(i)} \text{ for } i = 1, \ldots, M.
20: Let \Hat{Y}_{s+1}^{(j)} = \Hat{U}((\Hat{S}_{s+1}^{(j)}))^{(i)} + \Hat{V}_{s+1}^{(i)} - \Hat{\Gamma}((\Hat{x}_{s+1}^{(j)}))^{(i)} \text{ for } i = 1, \ldots, M.
21: if \sum_{j=1}^M \left(\Hat{S}_{s+1}^{(j)}\right)^{\top} \Hat{Y}_{s}^{(j)} + \varepsilon \sum_{j=1}^M \left(\Hat{S}_{s+1}^{(j)}\right)^{\top} \Hat{Y}_{s}^{(j)} \geq c \sum_{j=1}^M \left(\Hat{S}_{s+1}^{(j)}\right)^{\top} \Hat{Y}_{s}^{(j)} \right)^{\top} \Hat{Y}_{s}^{(j)} \text{ then}
22: Store the L-BFGS pair \{\Hat{S}_{s+1}^{(j)}\}_{j=1}^M, \{\Hat{Y}_{s+1}^{(j)}\}_{j=1}^M, \text{ and discard the oldest pair if the memory size is exceeded.}
23: end if
24: Transport the L-BFGS pairs \{\Hat{S}_{s+1}^{(j)}\}_{j=1}^M, \{\Hat{Y}_{s+1}^{(j)}\}_{j=1}^M \text{ from } \Hat{x}_s^{(j)} \text{ to } \Hat{x}_{s+1}^{(j)} \text{ using } \Hat{\Gamma}((\Hat{x}_{s+1}^{(j)}))^{(i)}.
25: end for
26: return \{\Hat{x}_{s+1}^{(i)}\}_{i=1}^M.
Experimental results on all datasets

(a) $N = 1030$, $D = 9$, $\text{cond}(\Sigma) = 73$, concrete

(b) $N = 1503$, $D = 6$, $\text{cond}(\Sigma) = 12$, noise

(c) $N = 5875$, $D = 21$, $\text{cond}(\Sigma) = 65697$, parkinson

(d) $N = 17379$, $D = 13$, $\text{cond}(\Sigma) = 218$, bike

Figure A.1. Experimental results for Bayesian linear regression (page 1)
Experimental results on all datasets

Figure A.2. Experimental results for Bayesian linear regression (page 2)
Experimental results on all datasets

(a) $N = 768$, $D = 9$, $\text{cond}(\Sigma) = 5$, pima

(b) $N = 1151$, $D = 20$, $\text{cond}(\Sigma) = 1106$, diabetic

(c) $N = 14980$, $D = 15$, $\text{cond}(\Sigma) = 4730$, eeg

(d) $N = 58000$, $D = 10$, $\text{cond}(\Sigma) = 3109$, space

*Figure A.3. Experimental results for Bayesian logistic regression (page 1)*
Supplementary Material for Variance Reduction and Quasi-Newton for Particle-Based Variational Inference

Experimental results on all datasets

(c) $N = 100000$, $D = 19$, $\text{cond}(\Sigma) = 2412$, susy

(f) $N = 12214$, $D = 51$, $\text{cond}(\Sigma) = 58$, mnist (with reduced dimension of 50 using PCA, same as in the main paper)

(g) $N = 464809$, $D = 55$, $\text{cond}(\Sigma) = 341266$, covtype

(h) $N = 12214$, $D = 785$, mnist with $28 \times 28$ image pixels as input (using 1,000 particles)

Figure A.4. Experimental results for Bayesian logistic regression (page 2)
Visualizations

(a) Mean of the particles as a function of number of iterations

(b) Standard deviation of the particles as a function of number of iterations

(c) Estimated marginal posterior distribution using kernel density estimation

Figure A.5. Visualizations for 3 selected dimensions out of 51 on the mnist dataset. The 3 columns represent the 3 selected dimensions. For each dimension and each algorithm, (a) the mean and (b) the standard deviation of the particles are plotted as a function of the number of iterations, and (c) the kernel density estimate of the final set of particles is shown. We see that the methods with variance reduction (SQN-VR, SVRG, SPIDER) achieve much better approximation of the posterior.
Comparison with Stein variational Newton

We compare with the Stein variational Newton (SVN) method by Detommaso et al. (2018). We implemented two versions of SVN, “full” and “block-diagonal” approximation. For SVN, we use the entire dataset for computing the gradient and Hessian since there is no extension of SVN to the mini-batch setting. We choose the number of iterations for SVN so that the total number of passes over the dataset for SVN is the same as the other methods. The experimental results are presented in Figures A.6 to A.9.

SVN-full performs comparable to SQN-VR in many of the experiments. The experiments with the biggest differences in terms of $\log_{10}(\text{MMD})$ are SVN-full being better than SQN-VR by about 0.4 on toms and kegg but being worse than SQN-VR by about 1.3 on mnist and 0.9 on covtype. However, the running time of SVN-full can be much longer than SQN-VR. For example, our implementation of SVN-full took around 5.5 hours for toms (vs 15 minutes for SQN-VR) and 30 minutes for kegg (vs 10 minutes for SQN-VR).

SVN-block-diagonal, in terms of MMD, is comparable to SQN-VR on 2 of the 8 linear regression datasets but is worse than SQN-VR on all of the other datasets. In these cases, SVN-block-diagonal is often comparable to SGD and AdaGrad in terms of MMD.
Comparison with Stein variational Newton

(a) N = 1030, D = 9, cond(Σ) = 73, concrete

(b) N = 1503, D = 6, cond(Σ) = 12, noise

(c) N = 5875, D = 21, cond(Σ) = 65697, parkinson

(d) N = 17379, D = 13, cond(Σ) = 218, bike

Figure A.6. Comparison with Stein variational Newton for Bayesian linear regression (page 1)
Comparison with Stein variational Newton

Figure A.7. Comparison with Stein variational Newton for Bayesian linear regression (page 2)
Comparison with Stein variational Newton

(a) N = 768, D = 9, cond(Σ) = 5, pima

(b) N = 1151, D = 20, cond(Σ) = 1106, diabetic

(c) N = 14980, D = 15, cond(Σ) = 4730, eeg

(d) N = 58000, D = 10, cond(Σ) = 3109, space

Figure A.8. Comparison with Stein variational Newton for Bayesian logistic regression (page 1)
Comparison with Stein variational Newton

Figure A.9. Comparison with Stein variational Newton for Bayesian logistic regression (page 2)
Comparison with ADVI

We compare with Automatic Differentiation Variational Inference (ADVI) by Kucukelbir et al. (2017). Specifically, we run Gaussian parametric variational inference with a full-rank Gaussian using the implementation of ADVI in PyMC3 (https://docs.pymc.io/). We use a similar experimental setup as in our paper: a batch size of 10, 100 Monte Carlo samples from the variational Gaussian distribution to estimate the loss (corresponding to our 100 particles), and AdaGrad optimizer. We do a grid search over learning rates and choose the one that attains the lowest loss at the end of the run. The experimental results are shown in Figures A.10 to A.13.

We find that the ADVI solution has much higher MSE_mu and MSE_cov than SQN-VR on every dataset except toms, for which ADVI has lower MSE_cov by $10^1$ but higher MSE_mu by $10^3$, and was often comparable to AdaGrad. One possible explanation is the doubly stochastic (data sub-sampling and Monte Carlo sampling from the variational distribution) gradients in ADVI, so ADVI only learns an approximate solution.
Comparison with ADVI

(a) $N = 1030, D = 9, \text{cond}((\Sigma)) = 73$, concrete

(b) $N = 1503, D = 6, \text{cond}((\Sigma)) = 12$, noise

(c) $N = 5875, D = 21, \text{cond}((\Sigma)) = 65697$, parkinson

(d) $N = 17379, D = 13, \text{cond}((\Sigma)) = 218$, bike

Figure A.10. Comparison with ADVI for Bayesian linear regression (page 1)
Comparison with ADVI

Figure A.11. Comparison with ADVI for Bayesian linear regression (page 2)
Comparison with ADVI

(a) N = 768, D = 9, cond(Σ) = 5, pima

(b) N = 1151, D = 20, cond(Σ) = 1106, diabetic

(c) N = 14980, D = 15, cond(Σ) = 4730, eeg

(d) N = 58000, D = 10, cond(Σ) = 3109, space

Figure A.12. Comparison with ADVI for Bayesian logistic regression (page 1)
Comparison with ADVI

Figure A.13. Comparison with ADVI for Bayesian logistic regression (page 2)