1st Symposium on Advances in Approximate Bayesian Inference, 2018 1–5

NeuTra-lizing Bad Geometry in Hamiltonian Monte Carlo Using Neural Transport

Matthew D. Hoffman MHOFFMAN@GOOGLE.COM
Pavel Sountsov SIEGE@GOOGLE.COM
Joshua Dillon JVDILLON@GOOGLE.COM
Ian Langmore LANGMORE@GOOGLE.COM
Dustin Tran TRANDUSTIN@GOOGLE.COM
Srinivas Vasudevan SRVASUDE@GOOGLE.COM
Google AI

1. Introduction

Hamiltonian Monte Carlo (HMC; Duane et al., 1987; Neal, 2011) is an Markov chain Monte Carlo (MCMC) algorithm that is well suited to sampling from high-dimensional distributions. It introduces a set of auxiliary variables that let one generate Metropolis-Hastings proposals (Metropolis et al., 1953; Hastings, 1970) by simulating the dynamics of a fictional Hamiltonian physical system. However, HMC is not a silver bullet. When the geometry of the target distribution is unfavorable, it may take many evaluations of the log-probability of the target distribution and its gradient for the chain to mix between faraway states (Betancourt, 2017).

Parno and Marzouk (2014) proposed a way to fix such unfavorable geometry by applying a reversible transformation (or "transport map") based on a series of polynomial regressions that warps the space in which the MCMC chain is simulated. This transport map parameterization and MCMC are both inefficient in high dimensions, requiring introduction of independence assumptions that limit the transport map flexibility.

In this work, we propose using a series of inverse autoregressive flows (IAF; Kingma et al., 2016) parameterized by neural networks and fit using variational inference as the transport map, combined with HMC which can exploit the gradient information of the target distribution and the transport map. We evaluate our neural-transport HMC (NeuTra HMC for short) approach on a variety of synthetic and real problems, and find that it can consistently outperform HMC, often by an order of magnitude. We also adapt this strategy to train variational autoencoders (Rezende et al., 2014; Kingma and Welling, 2014).

2. Neural Transport MCMC

Normalizing-flow variational inference proceeds by minimizing the KL divergence between the transformed distribution obtained by passing standard-normal random variables z through an invertible flow f_ϕ (which has density $q(\theta) = \mathcal{N}(f_\phi^{-1}(\theta); \mathbf{0}, \mathrm{I})|\frac{\partial f^{-1}}{\partial \theta}|$) and the target distribution $p(\theta)$.

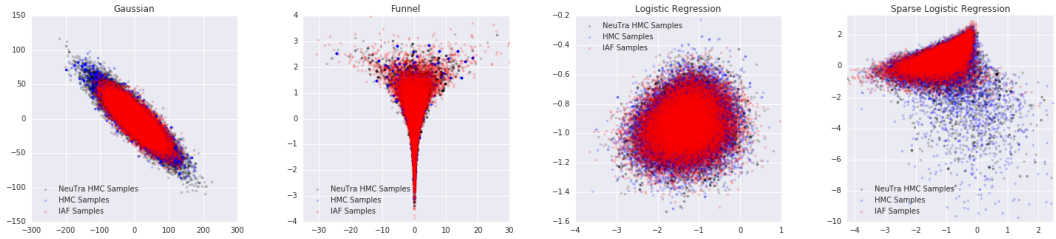


Figure 1: 2-dimensional projections of samples from an IAF variational distribution, HMC, and NeuTra HMC on the four unsupervised target distributions.

Marzouk et al. (2016) note that the process of fitting a transport map by variational inference can be interpreted in terms of the inverse map. KL divergence is invariant to changes of variables, so minimizing $\mathrm{KL}(q(\theta) \mid\mid p(\theta))$, is equivalent to minimizing $\mathrm{KL}(q(z) \mid\mid p(z))$. That is, in z-space variational inference is trying to warp the pulled-back target distribution $p(z)$ to look as much as possible like the fixed distribution $q(z)$.

If we have tuned the parameters ϕ of the map f_ϕ so that $p(z) = p(\theta = f_\phi(z))|\frac{\partial f}{\partial z}| \approx q(z)$, and $q(z)$ is relatively easy to sample from by MCMC, then we can efficiently sample from $p(\theta)$ by running a Markov chain whose target distribution is $p(z)$.

We propose two main improvements to the approaches of Marzouk et al. (2016) that scale their transport-map MCMC idea to the higher-dimensional problems common in Bayesian statistics and probabilistic machine learning. First, we use Hamiltonian Monte Carlo (HMC; Duane et al., 1987; Neal, 2011), which, because it uses uses gradient information, is able to mix dramatically faster than competing MCMC methods in high dimensions (in $\Omega(D^{1/4})$ steps instead of $\Omega(D)$). Second, we use IAFs, which are more scalable (and likely more powerful) than polynomial maps. We call the resulting approach neural-transport HMC, or NeuTra HMC for short.

To summarize, given a target distribution $p(\theta)$, NeuTra HMC proceeds in three steps:

1. Fit an IAF to minimize the $KL(q(z)|\frac{\partial f}{\partial z}|^{-1}||p(\theta))$.
2. Run HMC with target distribution $p(z)$, initialized with a sample from $q(z)$.
3. Push the z-space samples forward through f to get samples from $p(\theta)$.

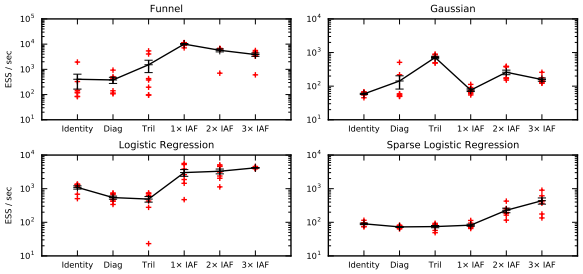
Note that we never need to compute the inverse $f^{-1}(\theta)$, which is expensive for IAFs.

3. Experiments

We evaluate NeuTra HMC's performance on a variety of target distributions. All experimental code is open-sourced at `github.com/hidden-for-submission`.

For each distribution we consider a family of IAFs with differing numbers of flows. We also considered two non-neural maps as baselines: a per-component scale vector ("Diag") and shift and a lower-triangular affine transformation ("Tril") and shift. When comparing to non-preconditioned HMC, we term that condition the "Identity" map.

(*a*)



(*b*)

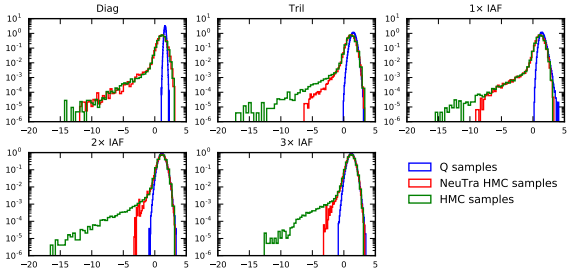


Figure 2: a) ESS per second for the unconditional distributions (higher is better). b) Histograms of one of the components of the per-element log scale parameter $\boldsymbol{\tau}$ in the sparse logistic regression problem for different transport maps.

In all cases, we trained the transport maps with a decaying learning rate using Adam (Kingma and Ba, 2015). Before evaluating the samplers, we tuned the HMC step size and number of leapfrog steps to fully explore the target distributions .

3.1. Unconditional Target Distributions

Neal's Funnel Distribution: We consider a $D = 100$ dimensional version of the funnel distribution described by Neal (2003). This distribution mimics the geometry of a hierarchical Bayesian prior with a centered parameterization, which is known to be problematic for HMC (Neal, 2011).

Ill-conditioned Gaussian: We take a $D = 200$ dimensional Gaussian distribution with the covariance matrix with eigenvalues sampled from $\text{Gam}(k = 0.5, \theta = 1)$. The covariance matrix is quenched (sampled once and shared among all the experiments). In practice, the eigenvalues range over 6 orders of magnitude.

Hierarchical logistic regression: We consider two types of hierarchical logistic regression applied to the German credit dataset. We assume a "soft-sparsity" logistic regression model,

$$p(\log \tau_0) = \mathcal{N}(0, 1) \qquad p(\boldsymbol{\beta}) = \mathcal{N}(\mathbf{0}, \mathrm{I})$$
$$p(\log \boldsymbol{\tau}) = \text{LogGam}(\alpha = 0.5, \beta = 0.5) \qquad p(\mathbf{y}|X, \boldsymbol{\beta}, \boldsymbol{\tau}, \tau_0) = \text{Bern}(\sigma(\tau_0 X(\boldsymbol{\beta} \odot \boldsymbol{\tau}))),$$

The sparse gamma prior on $\boldsymbol{\tau}$ imposes a soft sparsity prior on the centered coefficients, which could be used for variable selection. This parameterization uses D=125 dimensions.

To simplify the model, we also consider a non-sparse version where we drop the per-element scale terms, yielding a distribution with $D = 63$ dimensions and a simpler correlation structure.

Figure 1 shows samples from the three-flow IAF variational distribution, vanilla HMC, and NeuTra HMC. IAF is not able to perfectly match the Gaussian and sparse logistic regression target distributions, but NeuTra HMC generates good samples.

We next consider the gross chain statistics, focusing on ESS/sec Figure 2a, ESS normalized ESS by the total runtime of the chain. In most cases NeuTra HMC outperformed both the non-preconditioned HMC and affine transport maps. While the neural transport maps require a non-trivial amount of computation, the neutralization of the target distribution geometry allows HMC to take many fewer leapfrog steps while attaining acceptable convergence.

For the Gaussian, Tril outperforms NeuTra HMC due to simplicity of the target distribution, but falls short in distributions where the local geometry around the mode is not reflective of the tails.

Despite the good ESS performance, we were concerned whether NeuTra was fully exploring the distributions tails. In Figure 2b we plot histograms of one of components of the per-element log scale parameter $\boldsymbol{\tau}$ in the sparse logistic regression problem for different transport maps. While HMC uses $q(\theta)$ as the initial distribution, it quickly forgets the initial state and explores the tails independently of it. NeuTra, on the other hand, utilizes the geometry of the transport map, tying its performance to its quality. Since $q(\theta)$ is a poor approximation to the posterior for all maps considered, NeuTra performs the best (in terms of bias) when the map makes fewer assumptions about the global geometry.

3.2. Conditional Target Distributions

To incorporate NeuTra HMC into these models we build upon the interleaved training procedure of Hoffman (2017). The parameters of the approximate posterior and the transport map are trained using the standard ELBO.

We use the convolutional architecture from Kingma et al. (2016) with the IAF map and train it on dynamically binarized MNIST. (Wu et al., 2017). For NeuTra + IAF we obtain test log-likelihood of 79.35 ± 0.01 nats compared to a regular IAF VAE which gets 79.76 ± 0.03 nats. Using an independent Gaussian $q(\theta)$ yields 80.84 ± 0.02 nats.

4. Discussion

We described Neural-Transport (NeuTra) HMC, a method for accelerating Hamiltonian Monte Carlo sampling by nonlinearly warping the geometry of the target distribution using inverse autoregressive flows trained using variational inference. Using nonlinear IAFs instead of affine flows often dramatically improves mixing speed, especially on posteriors from hierarchical Bayesian models.

One remaining concern is that in some cases, when the maps fail to adequately capture the geometry of the target distribution, NeuTra may actually slow mixing in the tails. It would be interesting to explore architectures and regularization strategies that could safeguard against this.

References

Michael Betancourt. A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*, 2017.

Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

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

Matthew D. Hoffman. Learning deep latent Gaussian models with Markov chain Monte Carlo. In *International Conference on Machine Learning*, pages 1510–1519, 2017.

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

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

Diederik P Kingma, Tim Salimans, and Max Welling. Improving variational inference with inverse autoregressive flow. *Advances in Neural Information Processing Systems*, (2011):1–8, 2016.

Youssef Marzouk, Tarek Moselhy, Matthew Parno, and Alessio Spantini. An introduction to sampling via measure transport. *arXiv preprint arXiv:1602.05023*, 2016.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

Radford M Neal. Slice sampling. *Annals of Statistics*, pages 705–741, 2003.

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

Matthew Parno and Youssef Marzouk. Transport map accelerated markov chain monte carlo. *arXiv preprint arXiv:1412.5492*, 2014.

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

Yuhuai Wu, Yuri Burda, Ruslan Salakhutdinov, and Roger Grosse. On the quantitative analysis of decoder-based generative models. In *International Conference on Learning Representations*, 2017.