



UvA-DARE (Digital Academic Repository)

Markov Chain Monte Carlo and Variational Inference: Bridging the Gap

Salimans, T.; Kingma, D.P.; Welling, M.

Publication date

2014

Document Version

Submitted manuscript

Published in

Accepted papers: Advances in Variational Inference: NIPS 2014 Workshop: 13 December 2014, Convention and Exhibition Center, Montreal, Canada

[Link to publication](#)

Citation for published version (APA):

Salimans, T., Kingma, D. P., & Welling, M. (2014). Markov Chain Monte Carlo and Variational Inference: Bridging the Gap. In *Accepted papers: Advances in Variational Inference: NIPS 2014 Workshop: 13 December 2014, Convention and Exhibition Center, Montreal, Canada* NIPS Foundation. https://drive.google.com/file/d/0BwY-r_90KHY4d3ZTNDJpY3FYRS1rVEVVb3lUQzMzdk01Q2VV/view?pli=1

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

Markov Chain Monte Carlo and Variational Inference: Bridging the Gap

Tim Salimans
Algoritmica
tim@algoritmica.nl

Diederik P. Kingma
University of Amsterdam
dpkingma@gmail.com

Max Welling
University of Amsterdam
welling.max@gmail.com

Abstract

Recent advances in stochastic gradient variational inference have made it possible to perform variational Bayesian inference with posterior approximations containing auxiliary random variables. This enables us to explore a new synthesis of variational inference and Monte Carlo methods where we incorporate one or more steps of MCMC into our variational approximation. We describe the theoretical foundations that make this possible and show some promising first results.

1 Stochastic gradient variational inference

At the center of Bayesian analysis is the posterior distribution $p(z|x)$, where z is a set of unknown parameters or latent variables and x is the observed data. If the prior $p(z)$ and likelihood $p(x|z)$ have been specified, the posterior distribution can be computed using Bayes' rule. In practice this computation is often intractable and we have to resort to approximation methods. One such approximation method is *variational inference*, which casts inference as an optimization problem where we introduce a parameterized posterior approximation $q_\theta(z|x)$ (or $q_\theta(z)$) which is fit to the posterior distribution by choosing its parameters θ to maximize the lower bound of the marginal likelihood:

$$\begin{aligned}\log p(x) &= \mathbb{E}_{q_\theta(z|x)}[\log p(x, z) - \log q_\theta(z|x)] + D_{KL}(q_\theta(z|x)||p(z|x)) & (1) \\ &\geq \mathbb{E}_{q_\theta(z|x)}[\log p(x, z) - \log q_\theta(z|x)] = \mathcal{L}. & (2)\end{aligned}$$

Since $\log p(x)$ is independent of θ , maximizing the bound \mathcal{L} w.r.t. θ will minimize the KL-divergence $D_{KL}(q_\theta(z|x)||p(z|x))$, and the bound is tight when $D_{KL}(q_\theta(z|x)||p(z|x)) = 0$. The approximate posterior $q_\theta(z|x)$ is some appropriately chosen probabilistic model that is differentiable w.r.t. θ and from which we can sample. In order to maximize the objective \mathcal{L} we need to evaluate the expectation with respect to $q_\theta(z|x)$, which itself is often also intractable. Recent work in *stochastic gradient variational inference* therefore proposes to approximate this expectation using Monte Carlo, by replacing \mathcal{L} with a sample average using random draws from $q_\theta(z|x)$. Obtaining gradients of this Monte Carlo estimate requires the application of the chain rule through the random sampling from $q_\theta(z|x)$ [1, 2, 3]. This can in many cases be realised by drawing from $q_\theta(z|x)$ in two steps: In the first step we draw a set of *primitive random variables* u from a fixed distribution $p(u)$, and we then transform those as $z = g_\theta(u, x)$ with $g_\theta(\cdot)$ chosen in such a way that z has $q_\theta(z|x)$ as its distribution. If this is the case we can apply backpropagation, differentiating through the sampling function to obtain unbiased stochastic estimates of the gradient of the lower bound objective with respect to θ [1, 2, 3]. Alternatively, the gradient of the lower bound may be approximated using Monte Carlo directly [4, 5, 6]. Once we have obtained a stochastic estimate of the gradient of (2) with respect to θ , we can use this estimate in a stochastic gradient-based optimization algorithm for fitting our approximation to the true posterior $p(z|x)$.

In case of a dataset with multiple datapoints x_i , it can be efficient to let the distribution $q(z|x)$ be an explicit function of x_i , since in that case there is often no necessity for 'local' variational parameters θ per individual datapoint x_i ; instead, q maps from global parameters θ and local observed value x_i

to a distribution over the local latent variable(s) z_i . We can then optimize over θ for all observations x_i jointly. The joint lower bound to be optimized in this case is given by

$$\sum_{i=1}^n \log p(x_i) \geq \sum_{i=1}^n \mathbb{E}_{q_\theta(z_i|x_i)} [\log p(z_i, x_i) - \log q_\theta(z_i|x_i)].$$

of which an unbiased estimator (and its gradients) can be constructed by sampling x_i from the empirical distribution and sampling from $q_\theta(z_i|x_i)$. We will often leave out the subscript i for brevity.

One convenient way of parameterizing such a conditional approximation $q_\theta(z|x)$ is by using an inference network as in Helmholtz machines [7] or the related variational auto-encoders [2, 3]. In this paper we instead propose to let x enter the approximation q via Markov transition kernels. In [8] the authors also explore the usage of Markov chains in q ; however their Markov chains for q have a restricted functional form consisting only of diffusion with small update steps, the chain does not sample from a posterior, and their $p(z)$ is not an arbitrary distribution, but rather chosen for simplicity.

1.1 Using auxiliary variables

One application of stochastic gradient variational inference shown in earlier work [1] is to fit a posterior approximation containing *auxiliary random variables*. We do this by introducing a new set of random variables y , and lower bounding (2) by

$$\mathbb{E}_{q(z|x)} [\log p(x, z) - \log q(z|x)] \geq \mathbb{E}_{q(y, z|x)} [\log p(x, z) - \log q(y, z|x) + \log r(y|x, z)], \quad (3)$$

where $r(y|x, z)$ is an auxiliary inference distribution which we are free to choose, and our marginal posterior approximation is given by $q(z|x) = \int q(y, z|x) dy$. The marginal approximation $q(z|x)$ is now a mixture of distributions of the form $q(z|x, y)$. Since this is a very rich class of distributions, this method may be used to obtain a closer fit to the exact posterior, see [1]. The choice $r(y|x, z) = q(y|x, z)$ (3) would be optimal, but again often intractable to compute; in practice, good results can be obtained by specifying a $r(y|x, z)$ that can approximate $q(y|x, z)$ to a reasonable degree. One way this can be achieved is by specifying $r(y|x, z)$ to be of some flexible parametric form, and optimizing the lower bound over the parameters of this distribution.

2 Integrating MCMC into the variational approximation

A popular alternative to variational inference is the method of *Markov Chain Monte Carlo* (MCMC). Like variational inference, MCMC starts by taking a random draw z_0 from some initial distribution $q(z_0)$ or $q(z_0|x)$. Rather than optimizing this distribution, however, MCMC methods subsequently apply a *stochastic transition operator* to the random draw z_0 :

$$z_t \sim q(z_t|z_{t-1}, x).$$

By judiciously choosing the transition operator $q(z_t|z_{t-1})$ and iteratively applying it many times, the outcome of this procedure, z_T , will be a random variable that converges in distribution to the exact posterior $p(z|x)$. The advantage of MCMC is that the samples it gives us can approximate the exact posterior arbitrarily well if we are willing to apply the stochastic transition operator a sufficient number of times. The downside of MCMC is that in practice we do not know how many times is sufficient, and getting a good approximation using MCMC can take a very long time. The central idea of this paper is that we can interpret the stochastic Markov chain $q(z|x) = q(z_0|x) \prod_{t=1}^T q(z_t|z_{t-1}, x)$ as a variational approximation in an expanded space by considering $z_0, z_1, \dots, z_{t-1} = y$ to be auxiliary random variables (see section 1.1), and optimize this approximation for improved results. Doing so gives us the following bound on the marginal log-likelihood:

$$\begin{aligned} \log p(x) &\geq \mathbb{E}_q [\log p(x, z_T) - \log q(z_0, \dots, z_T|x) + \log r(z_0, \dots, z_{t-1}|x, z_T)] \\ &= \mathbb{E}_q \left[\log [p(x, z_T)/q(z_0|x)] + \sum_{t=1}^T \log [r_t(z_{t-1}|x, z_t)/q_t(z_t|x, z_{t-1})] \right], \quad (4) \end{aligned}$$

where the subscript t in q_t and r_t highlights the possibility of using different transition operators q_t and inverse models r_t at different points in the Markov chain.

Algorithmically, we can obtain an unbiased approximation of the lower bound using the following iterative algorithm

Algorithm 1 MCMC lower bound

Require: Model with joint distribution $p(x, z)$ and a desired but intractable posterior $p(z|x)$

Require: Number of iterations T

Require: Transition operator(s) $q_t(z_t|x, z_{t-1})$

Require: Inverse model(s) $r_t(z_{t-1}|x, z_t)$

Draw an initial random variable $z_0 \sim q(z_0|x)$

Initialize the lower bound as $L = \log p(x, z_0) - \log q(z_0|x)$

for $t = 1 : T$ **do**

 Perform random transition $z_t \sim q_t(z_t|x, z_{t-1})$

 Calculate the ratio $\alpha_t = \frac{p(x, z_t)r_t(z_{t-1}|z_t)}{p(x, z_{t-1})q_t(z_t|z_{t-1})}$

 Update the lower bound $L = L + \log \alpha_t$

end for

return the approximate lower bound L , and approximate posterior draw z_T

The result of Algorithm 1 is an unbiased approximation of the variational lower bound. Under the assumptions of Section 1 it is also a smooth function of the parameters of our variational approximation $q_\theta(z|x)$ and our auxiliary inverse model $r_\theta(z_0, \dots, z_{t-1}|x, z_T)$. By differentiating the sampled lower bound with respect to parameters θ using stochastic backpropagation techniques [1, 2, 3] we obtain an unbiased estimate of the gradient of the bound w.r.t. the variational parameters; the variational parameters can then be optimized by stochastic gradient ascent.

Algorithm 2 Markov Chain Variational Inference (MCVI)

Require: Forward Markov model $q_\theta(z)$ and backward Markov model $r_\theta(z_0, \dots, z_{t-1}|z_T)$

Require: Parameters θ

Require: Stochastic estimate of the variational lower bound $\mathcal{L}(\theta)$ from Algorithm 1

while not converged **do**

 Obtain unbiased stochastic estimate \hat{g} with $E_q[\hat{g}] = \nabla_\theta \mathcal{L}(\theta)$ by differentiating the stochastic approximation $L(\theta)$ [1, 2, 3] or using a direct approximation [4, 5, 6]

 Update the parameters θ using gradient \hat{g} and a stochastic optimization algorithm

end while

return the final optimized variational parameters θ

In Algorithm 2 we suggest optimizing the bound over all MCMC steps jointly, which is expected to give the best results for a fixed number of MCMC steps. However, the iterative nature of the bound also suggests another approach: We can take any existing variational approximation and improve it by adding a single MCMC step and optimizing its local bound contribution $\mathbb{E}_q \log[\alpha_t]$. This way we can iteratively improve our posterior approximation, similar to how boosting algorithms are used to iteratively fit regression models. Note that this approach is not restricted to MCMC; it is not strictly necessary for the Markov chain to be an MCMC chain.

Algorithm 3 Boosted MCVI

Require: Unnormalized log posterior $\log p(x, z)$

Require: Variational approximation $q(z_0|x)$

for $t = 1 : T$ **do**

 Add transition operator $q_t(z_t|x, z_{t-1})$ and backward model $r_t(z_{t-1}|x, z_t)$, optimize their parameters to maximize the local contribution to the lower bound $\mathbb{E}_{q(z_t, z_{t-1})} \log[\alpha_t]$

 Set the new posterior approximation equal to $q(z_t) = \int q_t(z_t|z_{t-1})q(z_{t-1})dx_{t-1}$

end for

return the final posterior approximation $q(z_T)$

This approach may also be interesting for *online* Bayesian inference, where we could add an MCMC step every time a new data point x is obtained. Another potential application are dynamic Bayesian models, where we could track a time-varying parameter vector z using a variational Markov chain.

2.1 Detailed balance

For practical MCMC inference we almost always use a transition operator that satisfies *detailed balance*, i.e. a transition operator $q_t(z_t|x, z_{t-1})$ for which we have

$$\frac{p(x, z_t)\bar{q}_t(z_{t-1}|x, z_t)}{p(x, z_{t-1})q_t(z_t|x, z_{t-1})} = 1,$$

where $\bar{q}_t(z_{t-1}|x, z_t)$ denotes $q_t(z_t|x, z_{t-1})$ with its z arguments reversed (not $q(z_{t-1}|x, z_t)$: the conditional pdf of z_{t-1} given z_t under q). If our transition operator satisfies detailed balance, we can divide α_t in Algorithm 1 by the ratio above (i.e. 1) to give

$$\log[\alpha_t] = \log r_t(z_{t-1}|x, z_t) - \log \bar{q}_t(z_{t-1}|x, z_t).$$

By optimally choosing $p_t(z_{t-1}|x, z_t)$ in this expression, we can make the expectation $\mathbb{E}_q \log[\alpha_t]$ non-negative: if the iterate z_{t-1} has converged to the posterior distribution $p(z|x)$, then it follows from detailed balance that $\bar{q}_t(z_{t-1}|x, z_t) = q(z_{t-1}|x, z_t)$. In that case choosing $p_t(z_{t-1}|x, z_t) = \bar{q}_t(z_{t-1}|x, z_t)$ is optimal, and the lower bound is unaffected by the transition. If, on the other hand, the chain has not fully *mixed* yet, then $\bar{q}_t(z_{t-1}|x, z_t) \neq q(z_{t-1}|x, z_t)$: the last iterate z_{t-1} will then have a predictable dependence on the initial conditions which allows us to choose $r_t(z_{t-1}|x, z_t)$ in such a way that $E_q \log[\alpha_t]$ is positive and improves our lower bound. Hence a stochastic transition respecting detailed balance always improves our variational posterior approximation unless it is already perfect! In practice, the extent to which we can capitalize on this to improve our lower bound depends on an adequately powerful model $r_t(z_{t-1}|x, z_t)$.

A practical transition operator that satisfies detailed balance is Gibbs sampling. Another popular way of ensuring our transitions satisfy detailed balance is by correcting them using Metropolis-Hastings acceptance/rejection. In the latter case, the stochastic transition operator $q_t(z_t|x, z_{t-1})$ is constructed in two steps: First a stochastic *proposal* z'_t is generated from a distribution $\phi(z'_t|z_{t-1})$. Next, the *acceptance probability* is calculated as

$$\rho(z_{t-1}, z'_t) = \min \left[\frac{p(x, z'_t)\phi(z_{t-1}|z'_t)}{p(x, z_{t-1})\phi(z'_t|z_{t-1})}, 1 \right].$$

Finally, z_t is set to z'_t with probability $\rho(z_{t-1}, z'_t)$, and to z_{t-1} with probability $1 - \rho(z_{t-1}, z'_t)$. Taking these two steps together, the combined stochastic transition operator is found to be

$$q_t(z_t|x, z_{t-1}) = \rho(z_{t-1}, z_t)\phi(z_t|z_{t-1}) + s(z_{t-1})\delta_{z_{t-1}}(z_t)$$

with $s(z_{t-1}) = \int [1 - \rho(z_{t-1}, z_t)]\phi(z_t|z_{t-1})dz_t$, and $\delta_{z_{t-1}}(\cdot)$ a delta distribution centered at z_{t-1} .

Optimizing a variational approximation with Metropolis-Hastings corrected updates is difficult in practice, as the log reverse rejection probability $\log[s(z_{t-1})]$ cannot be stochastically approximated without bias. We therefore use Algorithm 1 without Metropolis-Hastings correction in the remainder of the paper.

3 Hamiltonian variational approximation

One of the most efficient and widely applicable MCMC methods is *Hamiltonian Monte Carlo* (HMC) [9]. HMC is an MCMC method for approximating continuous distributions $p(z|x)$ where the space of unknown variables is expanded to include a set of auxiliary variables y with the same dimension as z . These auxiliary variables are initialized with a random draw from a distribution $y'_t \sim q(y'_t|x, z_{t-1})$, after which the method simulates the dynamics corresponding to the Hamiltonian $H(y, z) = 0.5y^T M^{-1}y - \log p(x, z)$, where z and y are iteratively updated using the *leapfrog integrator*, see [9].

Hamiltonian dynamics of this form is a very effective way of exploring the posterior distribution $p(z|x)$ because the dynamics is guided by the gradient of the exact log posterior, and random walks

are suppressed by the auxiliary variables y , which are also called *momentum variables*. Furthermore, the transition from y'_t, z_{t-1} to y_t, z_t in HMC is deterministic, invertible and volume preserving, which means that we have

$$q(y_t, z_t | z_{t-1}, x) = q(y_t, z_t, z_{t-1} | x) / q(z_{t-1} | x) = q(y'_t, z_{t-1} | x) / q(z_{t-1} | x) = q(y'_t | z_{t-1}, x)$$

and similarly $r(y'_t, z_{t-1} | z_t, x) = r(y_t | z_t, x)$, with z_t, y_t the output of the Hamiltonian dynamics.

Using this choice of transition operator $q_t(y_t, z_t | z_{t-1}, x)$ and inverse model $r_t(y'_t, z_{t-1} | z_t, x)$ we obtain the following algorithm for stochastically approximating the log marginal likelihood lower bound:

Algorithm 4 Hamiltonian variational approximation of the lower bound

Require: Unnormalized log posterior distribution $\log p(x, z)$

Require: Number of iterations T

Require: Momentum initialization distribution(s) $q_t(y'_t | z_{t-1}, x)$ and inverse model(s) $r_t(y_t | z_t, x)$

Require: HMC stepsize and mass matrix ϵ, M

Draw an initial random variable $z_0 \sim q(z_0 | x)$

Initialize the lower bound as $L = \log[p(x, z_0)] - \log[q(z_0 | x)]$

for $t = 1 : T$ **do**

Draw initial momentum variables $y'_t \sim q_t(y'_t | x, z_{t-1})$

Simulate Hamiltonian dynamics $z_t, y_t = \text{Hamiltonian_Dynamics}(z_{t-1}, y'_t)$

Calculate the ratio $\alpha_t = \frac{p(x, z_t) r_t(y_t | x, z_t)}{p(x, z_{t-1}) q_t(y'_t | x, z_{t-1})}$

Update the lower bound $L = L + \log[\alpha_t]$

end for

return the lower bound L , and approximate posterior draw z_T

We can fit our variational approximation to the true posterior distribution by stochastically maximizing the lower bound with respect to q, r and the parameters (stepsize and mass matrix) of the Hamiltonian dynamics using Algorithm 2. We call this version of the algorithm *Hamiltonian Variational Inference*.

After running Hamiltonian Variational Inference to convergence, we have an optimized approximation $q(z|x)$ of the posterior distribution. Because our approximation automatically adapts to the local shape of the exact posterior, this approximation will often be better than a variational approximation with a fixed functional form, provided our model for $r_t(y_t | x, z_t)$ is flexible enough. Since we do not use Metropolis-Hastings acceptance/rejection each HMC step is not guaranteed to improve the variational approximation for fixed parameters θ . However, when we have optimized the bound over θ each HMC step can never make us worse off.

In addition to improving the quality of our approximation, we find that the HMC steps often reduce the variance in our stochastic gradient estimates as a side effect, thereby speeding up the optimization. The downside of using this algorithm is that its computational cost per iteration is higher than when using an approximate $q(z|x)$ of a fixed form, mainly owing to the need of calculating additional derivatives of $\log p(x, z)$. These derivatives may also be difficult to derive by hand, so it is advisable to use an automatic differentiation package such as Theano [10]. As a rule of thumb, using Hamiltonian Variational Inference with m MCMC steps and k leapfrog steps is about mk times as expensive per iteration as when using a fixed approximation $q(z|x)$. This may be offset by reducing the number of iterations, and in practice we find that adding a single MCMC step to a fixed-form approximation often speeds up the convergence of the lower bound optimization in wallclock time. The scaling of the computational demands in the dimensionality of z is the same for both methods and depends on the structure of $p(x, z)$.

Compared to the regular Hamiltonian Monte Carlo method, Algorithm 4 has a number of advantages: The samples drawn from $q(z|x)$ are independent, the parameters of the Hamiltonian dynamics (M, ϵ) are automatically tuned, and there are no rejections of transitions. Furthermore, we optimize a lower bound on the log marginal likelihood, and we can assess the approximation quality using the techniques discussed in [1]. By finding a good initial distribution $q(z_0)$, we may also speed up convergence to the true posterior. A downside is that Algorithm 4 will in general never match the true posterior exactly with a finite number of steps, while MCMC is asymptotically exact.

3.1 Example: A beta-binomial model for overdispersion

To demonstrate Hamiltonian Variational Inference we use an example from [11], which considers the problem of estimating the rates of death from stomach cancer for the largest cities in Missouri. The data is available from the R package LearnBayes. It consists of 20 pairs (n_j, x_j) where n_j contains the number of individuals that were at risk for cancer in city j , and x_j is the number of cancer deaths that occurred in that city. The counts x_j are overdispersed compared to what one could expect under a binomial model with constant probability, so [11] assumes a beta-binomial model with a two dimensional parameter vector z . The low dimensionality of this problem allows us to easily visualize the results.

We use a variational approximation containing a single HMC step so that we can easily integrate out the 2 momentum variables numerically for calculating the exact KL-divergence of our approximation and to visualize our results. We choose $q_\theta(z_0), q_\theta(y'_1|z_0), r_\theta(y_1|z_1)$ to all be multivariate Gaussian distributions with diagonal covariance matrix. The mass matrix M is also diagonal. The means of $q_\theta(y'_1|z_0)$ and $r_\theta(y_1|z_1)$ are defined as linear functions in z and $\nabla_z \log p(x, z)$, with adjustable coefficients. The covariance matrices are not made to depend on z . The approximation is run using different numbers of leapfrog steps in the Hamiltonian dynamics.

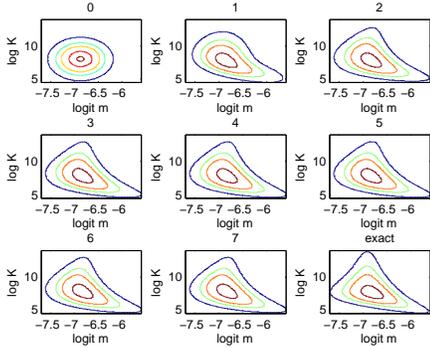


Figure 1: Approximate posteriors for a varying number of leapfrog steps. Exact posterior at bottom right.

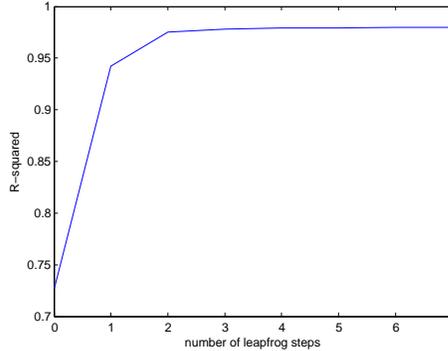


Figure 2: R-squared accuracy measure [1] for approximate posteriors using a varying number of leapfrog steps.

As can be seen from Figures 1 and 2, the Hamiltonian dynamics indeed helps us improve the posterior approximation. Most of the benefit is realized in the first two leapfrog iterations. Of course, more iterations may still prove useful for different problems and different specifications of $q_\theta(z_0), q_\theta(y'_1|z_0), r_\theta(y_1|z_1)$, and additional MCMC steps may also help. Adjusting only the means of $q_\theta(y'_1|z_0)$ and $r_\theta(y_1|z_1)$ based on the gradient of the log posterior is a simple specification that achieves good results. We find that even simpler parameterizations still do quite well, by finding a solution where the variance of $q_\theta(y'_1|z_0)$ is larger than that of $r_\theta(y_1|z_1)$, and the variance of $q_\theta(z_0)$ is smaller than that of $p(y|z)$: The Hamiltonian dynamics then effectively transfers entropy from y to z , resulting in an improved lower bound.

3.2 Example: Generative model for handwritten digits

Next, we demonstrate the effectiveness of our Hamiltonian variational approach for inference and learning with a deep generative neural network like described in [2]. The model will be fitted to a binarized version of the MNIST image dataset of handwritten digits as e.g. used in [12]; this dataset is often used as a comparative benchmark for probability density and mass modeling approaches.

Specifically, the generative model $p(x, z)$ consists of a spherical Gaussian prior $p(z) = \mathcal{N}(0, \mathbf{I})$, and conditional likelihood $p_\theta(x|z)$ parameterized as a multi-layer perceptron (MLP), i.e. a fully connected neural network. This MLP takes as input the latent variables z , and via two layers of each 300 hidden units with softplus ($\log(1 + e^v)$) nonlinearities, outputs a vector of conditionally independent Bernoulli scalars.

Model	$-\mathcal{L}$	$-\log p(x)$
<i>Results with $q(z_0 x) = \mathcal{N}(\mu, \sigma\mathbf{I})$:</i>		
5 leapfrog steps	90.86	87.16
10 leapfrog steps	87.60	85.56
<i>Results with $q(z_0 x)$ being an inference network:</i>		
No leapfrog steps	94.18	88.95
1 leapfrog step	91.70	88.08
4 leapfrog steps	89.82	86.40
8 leapfrog steps	89.03	85.79
<i>Results from [12]:</i>		
Mixture of Bernoullis K=10		168.95
Mixture of Bernoullis K=500		137.64
RBM (500 h, 25 CD steps) approx.		86.34
DBN 2hl approx.		84.55
NADE 1hl		88.33
Ensemble of NADE 1hl (2 orderings)		90.69
Ensemble of NADE 1hl (128 orderings)		87.71
Ensemble of NADE 2hl (2 orderings)		87.96
Ensemble of NADE 2hl (128 orderings)		85.10

Table 1: Comparison of our approach to other recent methods in the literature. we compare the average marginal log-likelihood measured in nats of the digits in the MNIST test set. See section 3.2 for details.

In [2] the authors describe an experiment where variational inference is performed with a doubly stochastic gradient ascent procedure, using a q distribution containing a neural network like the generative model. In each iteration, an unbiased estimate of the gradients for all variational and generative parameters is computed by differentiating the lower bound with a sample (minibatch) of 1000 datapoints and a single sample from q for each datapoint. We replace the inference network with the Hamiltonian posterior approximation as described in Algorithm 4 with $T = 1$ and a varying number of leapfrog steps. The auxiliary inference model $r(y|x, z)$ is chosen to be of a similar flexible parameteric form as the inference network in [2], namely a MLP with one deterministic hidden layer with 300 units and a Gaussian output variable with diagonal covariance. We tested two variants of the distribution $q(z_0|x)$. In one case, we let this distribution be a Gaussian with a mean and diagonal covariance structure that are learned, but independent of the datapoint x . In the second case, we let $q(z_0|x)$ be an inference network like $r(y|x, z)$, with two layers of each 300 hidden units and Gaussian output with diagonal covariance structure.

Stochastic gradient-based optimization was performed using a variant of RMSProp [13] with initialization bias correction, a learning rate of 0.001, first order decay of 0.1, and a second moment decay of 0.001. The optimization was run on 50,000 datapoints until convergence or divergence, which often took a few thousand epochs. The model iteration that produced the best likelihood lower bound on a validation set of 10,000 datapoints was selected.

The marginal likelihood of the test set was estimated with importance sampling by taking a Monte Carlo estimate of the expectation $p(x) = \mathbb{E}_{q(z|x)}[p(x, z)/q(z|x)]$ with over a thousand importance samples per test-set datapoint. See table 1 for our numerical results and a comparison to other methods. Without an inference network and with 10 leapfrog steps we were able to achieve a mean test-set lower bound of -87.6 , and an estimated mean marginal likelihood of -85.56 . When no Hamiltonian dynamics was included the gap is more than 5 nats; the smaller difference of 2 nats when 10 leapfrog steps were performed illustrates the bias-reduction effect of the MCMC chain.

4 Conclusion

By using auxiliary variables in combination with stochastic gradient variational inference we can construct posterior approximations that are much better than can be obtained using only simpler exponential family forms. One way of improving variational inference is by integrating one or more

MCMC steps into the approximation. By doing so we can bridge the accuracy/speed gap between MCMC and variational inference and get the best of both worlds.

References

- [1] Tim Salimans and David A Knowles. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882, 2013.
- [2] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *Proceedings of the 2nd International Conference on Learning Representations*, 2013.
- [3] 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 (ICML-14)*, pages 1278–1286, 2014.
- [4] John Paisley, David Blei, and Michael Jordan. Variational bayesian inference with stochastic search. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 1367–1374, 2012.
- [5] Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 814–822, 2014.
- [6] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In *The 31st International Conference on Machine Learning (ICML)*, 2014.
- [7] Geoffrey E Hinton and Richard S Zemel. Autoencoders, minimum description length, and helmholtz free energy. *Advances in neural information processing systems*, pages 3–3, 1994.
- [8] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Modeling arbitrary probability distributions using nonequilibrium thermodynamics. *Unpublished Manuscript*, 2014.
- [9] Radford Neal. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2, 2011.
- [10] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian Goodfellow, Arnaud Bergeron, Nicolas Bouchard, David Warde-Farley, and Yoshua Bengio. Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*, 2012.
- [11] Jim Albert. *Bayesian Computation with R*. Springer Science, New York. Second edition, 2009.
- [12] Benigno Uribe, Iain Murray, and Hugo Larochelle. A deep and tractable density estimator. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 467–475, 2014.
- [13] T. Tieleman and G. Hinton. Coursera: Neural networks for machine learning. 2012.