Markov Chain Monte Carlo and Variational Inference: Bridging the Gap

Tim Salimans TIM@ALGORITMICA.NL
Algoritmica

Diederik P. Kingma and Max Welling [D.P.KINGMA,M.WELLING]@UVA.NL
University of Amsterdam

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. By doing so we obtain a rich class of inference algorithms bridging the gap between variational methods and MCMC, and offering the best of both worlds: fast posterior approximation through the maximization of an explicit objective, with the option of trading off additional computation for additional accuracy. We describe the theoretical foundations that make this possible and show some promising first results.

1. MCMC and Variational Inference

Bayesian analysis gives us a very simple recipe for learning from data: given a set of unknown parameters or latent variables z that are of interest, we specify a prior distribution $p(z)$ quantifying what we know about z before observing any data. Then we quantify how the observed data x relates to z by specifying a likelihood function $p(x|z)$. Finally, we apply Bayes' rule $p(z|x) = p(z)p(x|z)/\int p(z)p(x|z)dz$ to give the posterior distribution, which quantifies what we know about z after seeing the data.

Although this recipe is very simple conceptually, the implied computation is often intractable. We therefore need to resort to approximation methods in order to perform Bayesian inference in practice. The two most popular approximation methods for this purpose are variational inference and MCMC . The former has the advantage of maximizing an explicit objective , and being faster in most cases . The latter has the advantage of being nonparametric and asymptotically exact . Here , we show how both methods can be combined in order to get the best of both worlds.

1.1. Variational Inference

Variational inference casts Bayesian inference as an optimization problem where we introduce a parameterized posterior approximation $q_\theta(z|x)$ which is fit to the posterior distribution by choosing its parameters θ to maximize a lower bound $\mathcal{L}$

1.2. MCMC and Auxiliary Variables

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

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

idea 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 $y = z_0, z_1, \dots, z_{t-1}$ to be a set of *auxiliary rvs*.

$$\begin{aligned}
&\mathcal{L}_{\text{aux}} && (3)\\
&= \mathbb{E}_{q(y,z_T|x)}[\log[p(x,z_T)r(y|x,z_T)] - \log q(y,z_T|x)]\\
&= \mathcal{L} - \boxed{\mathbb{E}_{q(z_T|x)}\{D_{KL}[q(y|z_T,x)||r(y|z_T,x)]\}}\\
&\leq \mathcal{L} \ \leq \ \log[p(x)],
\end{aligned}$$

where $r(y|x, z_T)$ is an auxiliary inference distribution which we are free to choose, and our marginal posterior approximation is given by $q(z_T|x) = \int q(y, z_T|x)\mathrm{d}y$. The marginal approximation $q(z_T|x)$ is now a mixture of distributions of the form $q(z_T|x, y)$. Since this is a very rich class of distributions, auxiliary variables may be used to obtain a closer fit to the exact posterior (Salimans & Knowles, 2013). The choice $r(y|x, z_T) = q(y|x, z_T)$ would be optimal, but again often intractable to compute; in practice, good results can be obtained by specifying a $r(y|x, z_T)$ that can approximate $q(y|x, z_T)$ to a reasonable degree. One way this can be achieved is by specifying $r(y|x, z_T)$ to be of some flexible parametric form, and optimizing the lower bound over the parameters of this distribution. In this paper we consider the special case where the auxiliary inference distribution also has a Markov structure just like the posterior approximation: $r(z_0, \dots, z_{t-1}|x, z_T) = \prod_{t=1}^{T} r_t(z_{t-1}|x, z_t)$,

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

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. By specifying these q_t and r_t in some flexible parametric form, we can then optimize the value of (4) in order to get a good approximation to the true posterior distribution.

2. Optimizing the lower bound

For most choices of the transition operators q_t and inverse models r_t, the auxiliary vlb (4) cannot be calculated analytically. However, if we can at least sample from the transitions q_t, and evaluate the inverse models r_t at those samples, we can still approximate the vlb without bias using the following algorithm:

Algorithm 1 MCMC lower bound estimate

```
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 rv $z_0 \sim q(z_0|x)$
  Initialize the lower bound estimate 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 $\alpha_t = \frac{p(x,z_t)r_t(z_{t-1}|x,z_t)}{p(x,z_{t-1})q_t(z_t|x,z_{t-1})}$
    Update the lower bound $L = L + \log[\alpha_t]$
  end for
  return the unbiased lower bound estimate $L$
```

The key insight behind the recent work in *stochastic gradient variational inference* . if all the individual steps of an algorithm like this are differentiable in the parameters of q and r, which we denote by θ, then so is the algorithm 's output L. Since L is an unbiased estimate of the vlb, its derivative is then an unbiased estimate of the derivative of the lower bound , which can be used in a stochastic optimization algorithm.

Obtaining gradients of the Monte Carlo estimate of Algorithm 1 requires the application of the chain rule through the random sampling of the transition operators $q_t(z_t|x, z_{t-1})$. This can in many cases be realized by drawing from these operators in two steps: In the first step we draw a set of *primitive random variables* u_t from a fixed distribution $p(u_t)$, and we then transform those as $z_t = g_\theta(u_t, x)$ with a transformation $g_\theta()$ chosen in such a way that z_t follows the distribution $q_t(z_t|x, z_{t-1})$. If this is the case we can apply BP , differentiating through the sampling function to obtain unbiased stochastic estimates of the gradient of the lb objective wrt θ (Salimans & Knowles , 2013 ; Kingma & Welling , 2014 ; Rezende et al., 2014). An alternative solution, which we do not consider here, would be to approximate the gradient of the lb using Monte Carlo directly (Paisley et al., 2012; Ranganath et al., 2014; Mnih & Gregor, 2014).

Once we have obtained a stochastic estimate of the gradient , we can use this estimate in a stochastic gradient -based optimization algorithm for fitting our approximation to the true posterior *p*(*z*|*x*).

Algorithm 2 Markov Chain Variational Inference (MCVI)

```
Require: Forward Markov model $q_\theta(z)$ and backward Markov model $r_\theta(z_0, \ldots, z_{t-1}|z_T)$
Require: Parameters $\theta$
Require: Stochastic estimate $L(\theta)$ of the variational lower bound $\mathcal{L}_{\text{aux}}(\theta)$ from Algorithm 1
  while not converged do
    Obtain unbiased stochastic estimate $\hat{g}$ with $E_q[\hat{g}] = \nabla_\theta \mathcal{L}_{\text{aux}}(\theta)$ by differentiating $L(\theta)$
    Update the parameters $\theta$ using gradient $\hat{g}$ and a stochastic optimization algorithm
  end while
  return final optimized variational parameters $\theta$
```

2.1. Example: bivariate Gaussian

$$p(z^1, z^2) \propto \exp\left[-\frac{1}{2\sigma_1^2}(z^1 - z^2)^2 - \frac{1}{2\sigma_2^2}(z^1 + z^2)^2\right].$$

We consider two MCMC methods that update the univariate z^1, z^2 in turn. The first method is Gibbs sampling, which samples from the Gaussian full conditional distributions $p(z^i|z^{-i}) = N(\mu_i, \sigma_i^2)$. The second method is the over-relaxation method of (Adler, 1981), which instead updates the univariate z^i using $q(z_t^i|z_{t-1}) = N[\mu_i + \alpha(z_{t-1}^i - \mu_i), \sigma_i^2(1 - \alpha^2)]$. For $\alpha = 0$ the two methods are equivalent, but for other values of α the over-relaxation method may mix more quickly than Gibbs sampling. To test this we calculate the vlb for this MCMC algorithm, and maximize wrt α to find the most effective transition operator.

For the inverse model $r(z_{t-1}|z_t)$ we use Gaussians with mean parameter linear in z_t and variance independent of z_{t-1} . For this particular case this specification allows us to recover the $q(z_{t-1}|z_t)$ distribution exactly. We use $\sigma_1 = 1, \sigma_2 = 10$ in our exact posterior, and we initialize the Markov chain at $(-10, -10)$, with addition of infinitesimal noise (variance of 10^{-10}). Figure 1 shows the lower bound for both MCMC methods: over-relaxation with an optimal α of -0.76 clearly recovers the exact posterior much more quickly than plain Gibbs sampling. The fact that optimization of the vlb allows us to improve upon standard methods like GS is promising for more challenging applications.

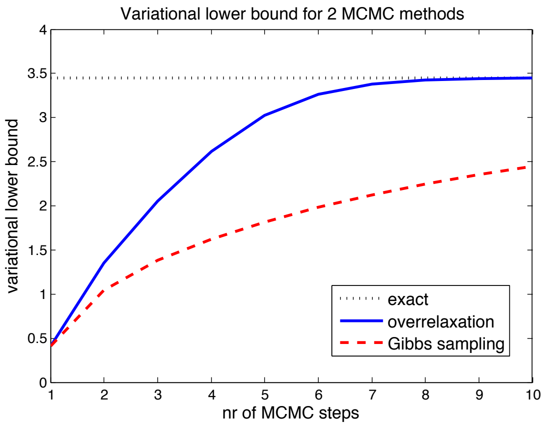

Figure 1. The log marginal likelihood lower bound for a bivariate Gaussian target and an MCMC variational approximation, using Gibbs sampling or Adler's overrelaxation.

3. Hamiltonian variational inference

One of the most efficient and widely applicable MCMC methods is *Hamiltonian Monte Carlo* (HMC) (Neal, 2011). 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 v with the same dimension as z. These auxiliary variables are initialized with a random draw from a distribution $v'_t \sim q(v'_t|x, z_{t-1})$, after which the method simulates the dynamics corresponding to the Hamiltonian $H(v, z) = 0.5v^T M^{-1} v - \log p(x, z)$, where z and v are iteratively updated using the *leapfrog integrator*, see (Neal, 2011).

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

$$\begin{aligned} q(v_t, z_t|z_{t-1}, x) &= q(v_t, z_t, z_{t-1}|x)/q(z_{t-1}|x) \\ &= q(v'_t, z_{t-1}|x)/q(z_{t-1}|x) = q(v'_t|z_{t-1}, x) \end{aligned}$$

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

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

Algorithm 3 Hamiltonian variational inference (HVI)

```
Require: Unnormalized log posterior $\log p(x, z)$
Require: Number of iterations $T$
Require: Momentum initialization distribution(s)
    $q_t(v'_t|z_{t-1}, x)$ and inverse model(s) $r_t(v_t|z_t, x)$
 Require: HMC stepsize and mass matrix $\epsilon, M$
                Draw an initial rv $z_0 \sim q(z_0|x)$
            Init. lb $L = \log[p(x, z_0)] - \log[q(z_0|x)]$
  for $t = 1 : T$ do
    Draw initial momentum $v'_t \sim q_t(v'_t|x, z_{t-1})$
    Set $z_t, v_t =$ Hamiltonian Dynamics$(z_{t-1}, v'_t)$
    Calculate the ratio $\alpha_t = \frac{p(x,z_t)r_t(v_t|x,z_t)}{p(x,z_{t-1})q_t(v'_t|x,z_{t-1})}$
          Update the lb $L = L + \log[\alpha_t]$
  end for
  return $L$, approx. posterior draw $z_T$
```

Here we omit the Metropolis-Hastings step that is typically used with Hamiltonian Monte Carlo. Section 4.1 discusses how such as step could be integrated into Algorithm 3.

We fit the 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* (HVI). After running the algorithm to convergence, we then 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 $r_t(v_t|x, z_t)$ is flexible enough.

In addition to improving the quality of our approximation, we find that adding HMC steps to a variational approximation often reduces the variance in our stochastic gradient estimates, 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 (Bastien et al., 2012). As a rule of thumb, using the Hamiltonian variational approximation with m MCMC steps and k leapfrog steps is about mk times as expensive per iteration as when using a fixed form approximation. 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 Hamiltonian variational approximation and fixed form variational approximation, and depends on the structure of $p(x, z)$.

Compared to regular Hamiltonian Monte Carlo, Algorithm 3 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 we may choose to omit the Metropolis-Hastings step so as not to reject any of the proposed transitions. Furthermore, we optimize a lower bound on the log marginal likelihood, and we can assess the approximation quality using the techniques discussed in (Salimans & Knowles, 2013). By finding a good initial distribution $q(z_0)$, we may also speed up convergence to the true posterior and get a good posterior approximation using only a very short Markov chain, rather than relying on asymptotic theory.

3.1. Example: A beta-binomial model for overdispersion

To demonstrate our Hamiltonian variational approximation algorithm we use an example from (Albert, 2009), 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 (Albert, 2009) 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 $q_\theta(z_0), q_\theta(v'_1|z_0), r_\theta(v_1|z_1)$ to all be multivariate Gaussian distributions with diagonal covariance matrix. The mass matrix M is also diagonal. The means of $q_\theta(v'_1|z_0)$ and $r_\theta(v_1|z_1)$ 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, and the approximation is run using different numbers of leapfrog steps in the Hamiltonian dynamics.

As can be seen from Figures 2 and 3, 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 $q_\theta(z_0), q_\theta(v'_1|z_0), r_\theta(v_1|z_1)$, and additional MCMC steps may also help. Adjusting only the means of $q_\theta(v'_1|z_0)$ and $r_\theta(v_1|z_1)$ 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 $q_\theta(v_1'|z_0)$ is larger than that of $r_\theta(v_1|z_1)$, and the variance of $q_\theta(z_0)$ is smaller than that of $p(v|z)$: The Hamiltonian dynamics then effectively transfers entropy from v to z, resulting in an improved lower bound.

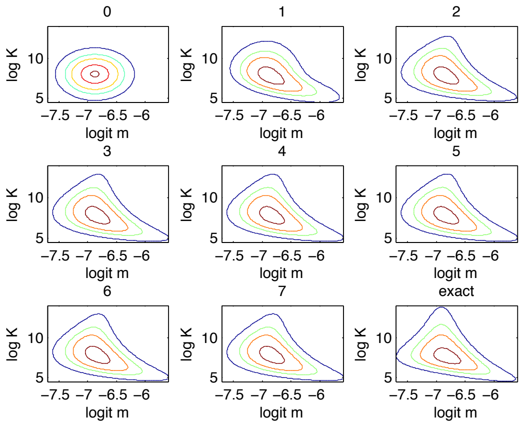

Figure 2. Approximate posteriors for a varying number of leapfrog steps. Exact posterior at bottom right.

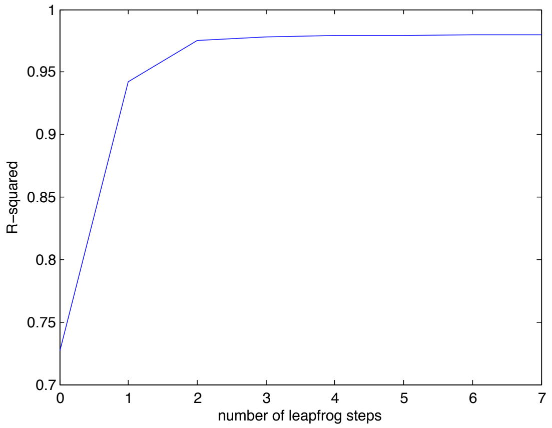

Figure 3. R-squared accuracy measure (Salimans & Knowles, 2013) for approximate posteriors using a varying number of leapfrog steps.

3.2. Example: Generative model for handwritten digits

Next, we demonstrate the effectiveness of our Hamiltonian variational inference approach for learning deep generative neural network models. These models are fitted to a binarized version of the MNIST dataset as e.g. used in (Uria et al., 2014). This dataset consists of 70000 data vectors x_i, each of which represents a black-and-white image of a handwritten digit. The task of modeling the distribution of these handwritten digit images is often used as a comparative benchmark for probability density and mass modeling approaches.

Our generative model $p(x_i, z_i)$ consists of a spherical Gaussian prior $p(z_i) = \mathcal{N}(0, \mathbf{I})$, and conditional likelihood (or *decoder*) $p_\theta(x_i|z_i)$ parameterized with either a fully connected neural network as in (Kingma & Welling, 2014; Rezende et al., 2014), or a convolutional network as in (Dosovitskiy et al., 2014). The network takes as input the latent variables z_i, and outputs the parameters of a conditionally independent (Bernoulli) distribution over the pixels.

Since we now have a dataset consisting of multiple datapoints x_i, with separate latent variables z_i per datapoint, it is efficient to let the distribution $q(z|x)$ be an explicit function of the data 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 lb to be optimized is

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

of which an unbiased estimator (and its gradients) can be constructed by sampling minibatches of data x_i from the empirical distribution and $z_i \sim q_\theta(z_i|x_i)$.

One flexible way of parameterizing the posterior approximation $q_\theta(z_i|x_i)$ is by using an inference network as in Helmholtz machines (Hinton & Zemel, 1994) or the related variational auto-encoders (VAE) (Kingma & Welling, 2014; Rezende et al., 2014). We can augment or replace such inference networks with the MCMC variational approximations developed here, as the parameters θ of the Markov chain can also be shared over all data vectors x_i.

Specifically, we replace or augment inference networks as used in (Kingma & Welling, 2014; Rezende et al., 2014) with a Hamiltonian posterior approximation as described in Algorithm 3, with $T = 1$ and a varying number of leapfrog steps. The auxiliary inference model $r(v|x, z)$ is chosen to be a fully-connected neural network with one deterministic hidden layer with $n_h = 300$ hidden units with softplus $(\log(1 + \exp(x)))$ activations 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 $r(v|x, z)$, with two layers of n_h hidden units, softplus activations and Gaussian output with diagonal covariance structure.

In a third experiment, we replaced the fully-connected networks with convolutional networks in both the inference model and the generative model. The inference model consists of three convolutional layers with 5×5 filters, [16,32,32] feature maps, stride of 2 and softplus activations. The convolutional layers are followed by a single fully-connected layer with $n_h = 300$ units and softplus activations. The architecture of the generative model mirrors the inference model but with stride replaced by upsampling, similar to (Dosovitskiy et al., 2014). The number of leapfrog steps was varied from 0 to 16. After broader model search with a validation set, we trained a final model with 16 leapfrog steps and $n_h = 800$.

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.

Model	$\log p(x)$ $\leq -$	$\log p(x)$ $= -$
HVI + fully-connected VAE:		
Without inference network:		
5 leapfrog steps	90.86	87.16
10 leapfrog steps	87.60	85.56
With inference network:		
No leapfrog steps	94.18	88.95
1 leapfrog step	91.70	88.08
4 leapfrog steps	89.82	86.40
8 leapfrog steps	88.30	85.51
HVI + convolutional VAE:		
No leapfrog steps	86.66	83.20
1 leapfrog step	85.40	82.98
2 leapfrog steps	85.17	82.96
4 leapfrog steps	84.94	82.78
8 leapfrog steps	84.81	82.72
16 leapfrog steps	84.11	82.22
16 leapfrog steps, $n_h = 800$	83.49	81.94
From (Gregor et al., 2015):		
DBN 2hl		84.55
EoNADE		85.10
DARN 1hl	88.30	84.13
DARN 12hl	87.72	
DRAW	80.97	

Stochastic gradient-based optimization was performed using Adam (Kingma & Ba, 2014) with default hyper-parameters. Before fitting our models to the full training set, the model hyper-parameters and number of training epochs were determined based on performance on a validation set of about 15% of the available training data. The marginal likelihood of the test set was estimated with importance sampling by taking a Monte Carlo estimate of the expectation $p(x) = \mathbb{E}_{q(z|x)}[p(x, z)/q(z|x)]$ (Rezende et al., 2014) with over a thousand importance samples per test-set datapoint.

See table 1 for our numerical results and a comparison to reported results with 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. Our best result is 81.94 nats with convolutional networks for inference and generation, and HVI with 16 leapfrog steps. This is slightly worse than the best reported number with DRAW (Gregor et al., 2015), a VAE with recurrent neural networks for both inference and generation. Our approaches are not mutually exclusive, and could indeed be combined for even better results.

The model can also be trained with a two-dimensional latent space to obtain a low-dimensional visualization of data. See figure 4 for a visualization of the latent space of such a model trained on the MNIST digits.

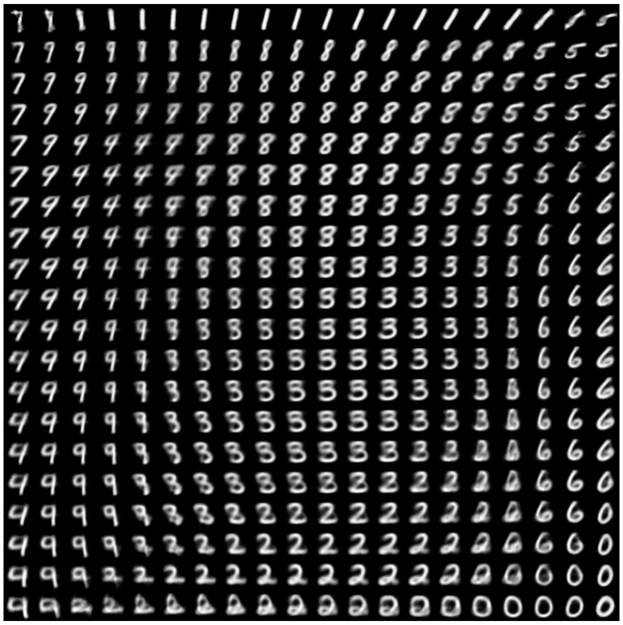

Figure 4. Visualization of the two-dimensional latent space of a generative model trained with our proposed Hamiltonian variational posterior approximation; shown here are the mean images $p(x|z)$ corresponding to different points z in latent space. Our proposed method results in better samples than what could be obtained when just using an inference network (without fine-tuning by Hamiltonian dynamics) as in (Kingma & Welling, 2014).

4. Specification of the Markov chain

In addition to the core contributions presented above, we now present a more detailed analysis of some possible specifications of the Markov chain used in the variational approximation. We discuss the impact of different specification choices on the theoretical and practical performance of the algorithm.

4.1. 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

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

where $\overleftarrow{q}_t(z_{t-1}|x, z_t)$ 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

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

By optimally choosing $r_t(z_{t-1}|x, z_t)$ in this expression, we can make the expectation $\mathbb{E}_q \log[\alpha_t]$ non-negative: what is required is that $r_t()$ is a predictor of the reverse dynamics that is equal or better than $\overleftarrow{q}_t()$. If the iterate z_{t-1} has converged to the posterior distribution $p(z|x)$ by running the Markov chain for a sufficient number of steps, then it follows from detailed balance that $\overleftarrow{q}_t(z_{t-1}|x, z_t) = q(z_{t-1}|x, z_t)$. In that case choosing $r_t(z_{t-1}|x, z_t) = \overleftarrow{q}_t(z_{t-1}|x, z_t)$ is optimal, and the lower bound is unaffected by the transition. If, on the other hand, the chain has not fully *mixed* yet, then $\overleftarrow{q}_t(z_{t-1}|x, z_t) \neq q(z_{t-1}|x, z_t)$: 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, we can only use this to improve our auxiliary lower bound if we also have an adequately powerful model $r_t(z_{t-1}|x, z_t)$ that can be made sufficiently close to $q(z_{t-1}|x, z_t)$.

A practical transition operator that satisfies detailed balance is Gibbs sampling, which can be trivially integrated into our framework as shown in Section 2.1. Another popular way of ensuring our transitions satisfy detailed balance is by correcting them using Metropolis-Hastings 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 $1-\rho(z_{t-1}, z'_t)$. The density of the resulting stochastic transition operator $q_t(z_t|x, z_{t-1})$ cannot be calculated analytically since it involves an intractable integral over $\rho(z_{t-1}, z'_t)$. To incorporate a Metropolis-Hastings step into our variational objective we will thus need to explicitly represent the acceptance decision as an additional auxiliary binary rv a. The Metropolis-Hastings step can then be interpreted as taking a reversible variable transformation with unit Jacobian:

$$\begin{aligned} z_{t-1} &\to \mathbb{I}[a = 1]z'_t + \mathbb{I}[a = 0]z_{t-1} \\ z'_t &\to \mathbb{I}[a = 1]z_{t-1} + \mathbb{I}[a = 0]z'_t \\ a &\to a. \end{aligned}$$

Evaluating our target density at the transformed variables, we get the following addition to the lower bound:

$$\begin{aligned} \log[\alpha_t] = \log[p(x, z_t)/p(x, z_{t-1})] + \log[r_t(a|x, z_t)] \\ + \mathbb{I}[a = 1]\log[r_t(z_{t-1}|x, z_t)] \\ + \mathbb{I}[a = 0]\log[r_t(z'_t|x, z_t)] \\ - \log[q_t(z'_t|x, z_{t-1})q(a|z'_t, z_{t-1}, x)]. \end{aligned}$$

Assuming we are working with a continuous variable z, the addition of the binary variable a has the unfortunate effect that our Monte Carlo estimator of the lower bound is no longer a continuously differentiable function of the variational parameters θ, which means we cannot use the gradient of the exact log posterior to form our gradient estimates. Estimators that do not use this gradient are available (Salimans & Knowles, 2013; Paisley et al., 2012; Ranganath et al., 2014; Mnih & Gregor, 2014) but these typically have much higher variance. We can regain continuous differentiability with respect to θ by Rao-Blackwellizing our Monte Carlo lower bound approximation L and calculating the expectation with respect to $q(a|z'_t, z_{t-1}, x)$ analytically. For short Markov chains this is indeed an attractive solution. For longer chains this strategy becomes computationally demanding as we need to do this for every step in the chain, thereby exploring all 2^T different paths created by the T accept/reject decisions. Another good alternative is to simply omit the Metropolis-Hastings acceptance step from our transition operators and to rely on a flexible specification for $q()$ and $r()$ to sufficiently reduce any resulting bias.

4.2. Annealed variational inference

Annealed importance sampling is an MCMC strategy where the Markov chain consists of stochastic transitions $q_t(z_t|z_{t-1})$ that each satisfy detailed balance with respect to an unnormalized target distribution $\log[p_t(z)] = (1 - \beta_t)\log[q_0(z)] + \beta_t \log[p(x, z)]$, for β_t gradually increasing from 0 to 1. The reverse model for annealed importance sampling is then constructed using transitions $r(z_{t-1}|z_t) = q_t(z_t|z_{t-1})p_t(z_{t-1})/p_t(z_t)$, which are guaranteed to be normalized densities because of detailed balance. For this choice of posterior approximation and reverse model, the marginal likelihood lower bound is then given by

$$\log p(x) \geq \mathbb{E}_q \sum_{t=1}^{T} (\beta_t - \beta_{t-1}) \log[p(x, z_t)/q_0(z_t)].$$

With $\beta_0 = 0, \beta_T = 1$ this looks like the bound we have at $t = 0$, but notice that the expectation is now taken with respect to a different distribution than q_0. Since this new approximation is strictly closer to $p(z|x)$ than the old approximation, its expectation of the log-ratio $\log[p(x, z_t)/q_0(z_t)]$ is strictly higher, and the lower bound will thus be improved.

The main advantage of annealed variational inference over other variational MCMC strategies is that it does not require explicit specification of the reverse model r, and that the addition of the Markov transitions to our base approximation $q_0(z)$ is guaranteed to improve the variational lower bound. A downside of using this scheme for variational inference is the requirement that the transitions $q(z_t|z_{t-1})$ satisfy detailed balance, which can be impractical for optimizing q.

4.3. Using multiple iterates

So far we have defined our variational approximation as the marginal of the last iterate in the Markov chain, i.e. $q(z_T|x)$. This is wasteful if our Markov chain consists of many steps, and practical MCMC algorithms therefore always use multiple samples $z_{T+1-K}, \ldots, z_T$ from the Markov chain, with K the number of samples. When using multiple samples obtained at different points in the Markov chain, our variational approximation effectively becomes a discrete *mixture* over the marginals of the iterates that are used:

$$\begin{aligned} q(z|x) &= \frac{1}{K} \sum_{t=T+1-K}^{T} q(z_t|x) \\ &= \sum_{t=T+1-K}^{T} \mathbb{I}(w = t) q(z_t|x), \\ &\text{with } w \sim \text{Categorical}(T + 1 - K, \ldots, T). \end{aligned}$$

To use this mixture distribution to form our lower bound, we need to explicitly take into account the *mixture indicator variable* w. This variable has a categorical distribution $q(w = t), t \in [T + 1 - K, \ldots, T]$ that puts equal probability on each of the K last iterates of the Markov chain, the log of which is subtracted from our variational lower bound (3). This term is then offset by adding the corresponding log probability of that iterate under the inverse model $r(w = t|x, z)$. The simplest specification for the inverse model is to set it equal to $q(w = t)$: In that case both terms cancel, and we're effectively just taking the average of the last K lower bounds L computed by Algorithm 1. Although suboptimal, we find this to be an effective method of reducing variance when working with longer Markov chains. An alternative, potentially more optimal approach would be to also specify the inverse model for w using a flexible parametric function such as a neural network, taking x and the sampled z as inputs.

4.4. Sequential MCVI

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. Another approach is to optimize the MCMC steps sequentially, by maximizing the local bound contributions $\mathbb{E}_q \log[\alpha_t]$. Using this approach, we can take any existing variational approximation and improve it by adding one or more MCMC steps, as outlined in Algorithm 4. Improving an existing approximation in this way gives us an easier optimization problem, and can be compared to how boosting algorithms are used to iteratively fit regression models.

Algorithm 4 Sequential 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 inverse
    model $r_t(z_{t-1}|x, z_t)$.
    Choose the new parameters by maximizing the local
    lower bound contribution $\mathbb{E}_{q(z_t, z_{t-1})} \log[\alpha_t]$
    Set the new posterior approximation equal to
    $q(z_t|x) = \int q_t(z_t|x, z_{t-1}) q(z_{t-1}|x) dz_{t-1}$
  end for
  return the final posterior approximation $q(z_T|x)$
```

5. 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

Adler, Stephen L. Over-relaxation method for the monte carlo evaluation of the partition function for multiquadratic actions. *Physical Review D*, 23(12):2901, 1981.

Albert, Jim. *Bayesian Computation with R*. Springer Science, New York. Second edition, 2009.

Bastien, Frédéric, Lamblin, Pascal, Pascanu, Razvan, Bergstra, James, Goodfellow, Ian, Bergeron, Arnaud, Bouchard, Nicolas, Warde-Farley, David, and Bengio, Yoshua. Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*, 2012.

Dosovitskiy, Alexey, Springenberg, Jost Tobias, and Brox, Thomas. Learning to generate chairs with convolutional neural networks. *arXiv preprint arXiv:1411.5928*, 2014.

Gregor, Karol, Danihelka, Ivo, Graves, Alex, and Wierstra, Daan. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.

Hinton, Geoffrey E and Zemel, Richard S. Autoencoders, minimum description length, and helmholtz free energy. *Advances in neural information processing systems*, pp. 3–3, 1994.

Kingma, Diederik and Ba, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, Diederik P and Welling, Max. Auto-Encoding Variational Bayes. *Proceedings of the 2nd International Conference on Learning Representations*, 2014.

Mnih, Andriy and Gregor, Karol. Neural variational inference and learning in belief networks. In *The 31st International Conference on Machine Learning (ICML)*, 2014.

Neal, Radford. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2, 2011.

Paisley, John, Blei, David, and Jordan, Michael. Variational bayesian inference with stochastic search. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pp. 1367–1374, 2012.

Ranganath, Rajesh, Gerrish, Sean, and Blei, David. Black box variational inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pp. 814–822, 2014.

Rezende, Danilo J, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1278–1286, 2014.

Salimans, Tim and Knowles, David A. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882, 2013.

Uria, Benigno, Murray, Iain, and Larochelle, Hugo. A deep and tractable density estimator. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 467–475, 2014. URL http://jmlr.org/proceedings/papers/v32/uria14.html.