MCMC and Variational Inference: Bridging the Gap

Tim Salimans
salimans.tim@gmail.com

Diederik P. Kingma
dpkingma@gmail.com

Max Welling
welling.max@gmail.com

July 8, 2015
MCMC vs. Variational Inference

MCMC approximates the posterior distribution $p(z|x)$ with

$$q(z_T|x) = \int q(z_0|x) \prod_{t=1}^{T} q(z_t|z_{t-1}, x) dz_0, \ldots, z_{T-1}$$
MCMC vs. Variational Inference

MCMC approximates the posterior distribution $p(z|x)$ with

$$q(z_T|x) = \int q(z_0|x) \prod_{t=1}^{T} q(z_t|z_{t-1}, x) dz_0, \ldots, z_{T-1}$$

- automatically adapts to true posterior
- asymptotically exact
MCMC vs. Variational Inference

MCMC approximates the posterior distribution \( p(z|x) \) with

\[
q(z_T|x) = \int q(z_0|x) \prod_{t=1}^{T} q(z_t|z_{t-1}, x) \, dz_0, \ldots, z_{T-1}
\]

- automatically adapts to true posterior
- asymptotically exact
- slow convergence, hard to assess quality
- tuning

⇒ Combine MCMC & VI using stochastic approximation
MCMC vs. Variational Inference

MCMC approximates the posterior distribution $p(z|x)$ with

$$q(z_T|x) = \int q(z_0|x) \prod_{t=1}^{T} q(z_t|z_{t-1}, x) dz_0, \ldots, z_{T-1}$$

- automatically adapts to true posterior
- asymptotically exact
- slow convergence, hard to assess quality
- tuning

⇒ Combine MCMC & VI using stochastic approximation
MCMC as Variational Inference

\[ q(z_T|x) = \int q(z_0|x) \prod_{t=1}^{T} q(z_t|z_{t-1}, x) \, dz_0, \ldots, z_{T-1} \]

with auxiliary variables \( y = (z_0, \ldots, z_{T-1}) \).
MCMC as Variational Inference

\[
q(z_T | x) = \int q(z_0 | x) \prod_{t=1}^{T} q(z_t | z_{t-1}, x) dz_0, \ldots, z_{T-1}
\]

with auxiliary variables \( y = (z_0, \ldots, z_{T-1}) \).

Variational lower bound

\[
\log p(x) \geq \mathbb{E}_{q(z_T | x)} \{ \log[p(x, z_T)] - \log[q(z_T | x)] \} = L
\]
MCMC as Variational Inference

\[ q(z_T|x) = \int q(z_0|x) \prod_{t=1}^{T} q(z_t|z_{t-1}, x) dz_0, \ldots, z_{T-1} \]

with auxiliary variables \( y = (z_0, \ldots, z_{T-1}) \).

Variational lower bound

\[ \log p(x) \geq \mathbb{E}_{q(z_T|x)} \{ \log[p(x, z_T)] - \log[q(z_T|x)] \} = L \]

Auxiliary variational lower bound

\[ \log p(x) \geq \mathbb{E}_{q(y,z_T|x)} \{ \log[p(x, z_T)r(y|z_T, x)] - \log[q(y, z_T|x)] \} \]
\[ = L - \mathbb{E}_{q(z_T|x)} \{ D_{KL}[q(y|z_T, x) || r(y|z_T, x)] \} \]

with \( r(y|z_T, x) \) an arbitrary auxiliary target distribution, e.g. \( r(y|z_T, x) = \prod_{t=1}^{T} r_t(z_{t-1}|z_t, x) \) a Markov model.
Sampling approximation of MCMC lower bound

Evaluate variational lower bound by sampling from $q(y, z_T|x)$. 
Sampling approximation of MCMC lower bound

Evaluate variational lower bound by sampling from $q(y, z_T|x)$.

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

**Require:** Reverse model(s) $r_t(z_{t-1}|z_t, x)$
Sampling approximation of MCMC lower bound

Evaluate variational lower bound by sampling from \( q(y, z_T | x) \).

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

**Require:** Reverse model(s) \( r_t(z_{t-1} | z_t, x) \)

Draw initial r.v. \( z_0 \sim q(z_0 | x) \)
Sampling approximation of MCMC lower bound

Evaluate variational lower bound by sampling from $q(y, z_T | x)$.

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

Draw initial r.v. $z_0 \sim q(z_0 | x)$
Init lower bound est. $\hat{L} = \log[p(x, z_0)/q(z_0 | x)]$
Sampling approximation of MCMC lower bound

Evaluate variational lower bound by sampling from $q(y, z_T | x)$.

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

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

1. Draw initial r.v. $z_0 \sim q(z_0 | x)$
2. Init lower bound est. $\hat{L} = \log[p(x, z_0) / q(z_0 | x)]$

   for $t = 1 : T$

   1. Random transition $z_t \sim q_t(z_t | z_{t-1}, x)$
   2. Calculate ratio $\alpha_t = \frac{p(x,z_t)r_t(z_{t-1}|z_t,x)}{p(x,z_{t-1})q_t(z_t|z_{t-1},x)}$
   3. Update lower bound $\hat{L} \leftarrow \hat{L} + \log[\alpha_t]$

end for
Sampling approximation of MCMC lower bound

Evaluate variational lower bound by sampling from $q(y, z_T | x)$.

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

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

1. Draw initial r.v. $z_0 \sim q(z_0 | x)$
2. Init lower bound est. $\hat{L} = \log[p(x, z_0)/q(z_0 | x)]$

for $t = 1 : T$ do

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

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

   Update lower bound $\hat{L} \leftarrow \hat{L} + \log[\alpha_t]$  

end for

**return** lower bound estimate $\hat{L}$
MCMC with detailed balance always improves the LB
MCMC with detailed balance always improves the LB

For $q_t(z_t|x, z_{t-1})$ satisfying *detailed balance*,

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

where $\tilde{q}_t(z_{t-1}|x, z_t)$ is the reversed transition.
MCMC with detailed balance always improves the LB

For \( q_t(z_t|x, z_{t-1}) \) satisfying \textit{detailed balance},

\[
\frac{p(x, z_t)\hat{q}_t(z_{t-1}|x, z_t)}{p(x, z_{t-1})q_t(z_t|x, z_t)} = 1,
\]

where \( \hat{q}_t(z_{t-1}|x, z_t) \) is the reversed transition. Then

\[
\log[\alpha_t] = \log r_t(z_{t-1}|x, z_t) - \log \hat{q}_t(z_{t-1}|x, z_t).
\]
MCMC with detailed balance always improves the LB

For \( q_t(z_t|x, z_{t-1}) \) satisfying \textit{detailed balance},

\[
\frac{p(x, z_t) \tilde{q}_t(z_{t-1}|x, z_t)}{p(x, z_{t-1}) q_t(z_t|x, z_{t-1})} = 1,
\]

where \( \tilde{q}_t(z_{t-1}|x, z_t) \) is the reversed transition. Then

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

For optimal \( r_t(z_{t-1}|x, z_t) = q(z_{t-1}|x, z_t) \),

\[
\mathbb{E}_q \log[\alpha_t] = \mathbb{E}_q \log q(z_{t-1}|x, z_t) - \log \tilde{q}_t(z_{t-1}|x, z_t) \geq 0.
\]
MCMC with detailed balance always improves the LB

For $q_t(z_t|x, z_{t-1})$ satisfying detailed balance,

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

where $\tilde{q}_t(z_{t-1}|x, z_t)$ is the reversed transition. Then

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

For optimal $r_t(z_{t-1}|x, z_t) = q(z_{t-1}|x, z_t),$

$$\mathbb{E}_q \log[\alpha_t] = \mathbb{E}_q \log q(z_{t-1}|x, z_t) - \log \tilde{q}_t(z_{t-1}|x, z_t) \geq 0.$$

▶ MCMC always improves approximation unless already perfect!
MCMC with detailed balance always improves the LB

For \( q_t(z_t|x, z_{t-1}) \) satisfying *detailed balance*,

\[
\frac{p(x, z_t) \tilde{q}_t(z_{t-1}|x, z_t)}{p(x, z_{t-1})q_t(z_t|x, z_{t-1})} = 1,
\]

where \( \tilde{q}_t(z_{t-1}|x, z_t) \) is the reversed transition. Then

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

For optimal \( r_t(z_{t-1}|x, z_t) = q(z_{t-1}|x, z_t) \),

\[
\mathbb{E}_q \log[\alpha_t] = \mathbb{E}_q \log q(z_{t-1}|x, z_t) - \log \tilde{q}_t(z_{t-1}|x, z_t) \geq 0.
\]

- MCMC always improves approximation unless already perfect!
- In practice we need \( r_t(z_{t-1}|x, z_t) \approx q(z_{t-1}|x, z_t) \).
Optimizing the Markov chain

- Specify parameterized Markov chain $q_{\theta}(z) = q_{\theta}(z_0|x) \prod_{t=1}^T q_{\theta}(z_t|z_{t-1}, x)$
- Specify parameterized auxiliary distribution $r_{\theta}(y|z_T)$
- Sample the variational lower bound $\hat{L}(\theta)$
- Do SGD using $\nabla_{\theta} \hat{L}(\theta)$ (reparameterization trick)

Require:
- Forward model $q_{\theta}(z)$ and backward model $r_{\theta}(y|z_T)$
- Parameters $\theta$, step size $\lambda$

while not converged do
  Get stochastic variational lower bound estimate $\hat{L}(\theta)$
  Update parameters using SGD: $\theta \leftarrow \theta + \lambda \nabla_{\theta} \hat{L}(\theta)$
end while

return final optimized variational parameters $\theta$
Optimizing the Markov chain

- Specify parameterized Markov chain
  \[ q_\theta(z) = q_\theta(z_0|x) \prod_{t=1}^{T} q_\theta(z_t|z_{t-1}, x) \]
Optimizing the Markov chain

- Specify parameterized Markov chain
  \[ q_\theta(z) = q_\theta(z_0|x) \prod_{t=1}^{T} q_\theta(z_t|z_{t-1}, x) \]
- Specify parameterized auxiliary distribution \( r_\theta(y|z_T) \)
Optimizing the Markov chain

- Specify parameterized Markov chain
  \[ q_\theta(z) = q_\theta(z_0|x) \prod_{t=1}^{T} q_\theta(z_t|z_{t-1}, x) \]
- Specify parameterized auxiliary distribution \( r_\theta(y|z_T) \)
- Sample the variational lower bound \( \hat{L}(\theta) \)
Optimizing the Markov chain

- Specify parameterized Markov chain
  \[ q_\theta(z) = q_\theta(z_0|x) \prod_{t=1}^T q_\theta(z_t|z_{t-1}, x) \]
- Specify parameterized auxiliary distribution \( r_\theta(y|z_T) \)
- Sample the variational lower bound \( \hat{L}(\theta) \)
- Do SGD using \( \nabla_\theta \hat{L}(\theta) \) (reparameterization trick)
Optimizing the Markov chain

- Specify parameterized Markov chain
  \[ q_{\theta}(z) = q_{\theta}(z_0|x) \prod_{t=1}^{T} q_{\theta}(z_t|z_{t-1}, x) \]
- Specify parameterized auxiliary distribution \( r_{\theta}(y|z_T) \)
- Sample the variational lower bound \( \hat{L}(\theta) \)
- Do SGD using \( \nabla_{\theta} \hat{L}(\theta) \) (reparameterization trick)

**Require:** Forward model \( q_{\theta}(z) \) and backward model \( r_{\theta}(y|z_T) \)

**Require:** Parameters \( \theta \), step size \( \lambda \)

```latex
\textbf{while} not converged \textbf{do}
  \text{Get stochastic variational lower bound estimate } \hat{L}(\theta) \\
  \text{Update parameters using SGD: } \theta \leftarrow \theta + \lambda \nabla_{\theta} \hat{L}(\theta)
\textbf{end while}
```

**return** final optimized variational parameters \( \theta \)
Gaussian example

Bivariate Gaussian target distribution:

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

Bivariate Gaussian target distribution:

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

Sample \(z^1, z^2\) in turn using

\[
q(z^i_t | z_{t-1}) = p(z^i_t | z^{-i}) = N(\mu_i, \sigma_i^2) \quad \text{(Gibbs sampling)}
\]
Gaussian example

Bivariate Gaussian target distribution:

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

Sample \( z^1, z^2 \) in turn using

\[ q(z^i_t|z_{t-1}) = p(z^i|z^{-i}) = N(\mu_i, \sigma_i^2) \quad \text{(Gibbs sampling)} \]

or

\[ q(z^i_t|z_{t-1}) = N[\mu_i + \alpha(z^i_{t-1} - \mu_i), \sigma_i^2(1 - \alpha^2)] \quad \text{(over-relaxation)} \]
Gaussian example

Bivariate Gaussian target distribution:

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

Sample \( z^1, z^2 \) in turn using

\[
q(z^i_t | z_{t-1}) = p(z^i_t | z^{-i}) = N(\mu_i, \sigma_i^2) \quad \text{(Gibbs sampling)}
\]

or

\[
q(z^i_t | z_{t-1}) = N[\mu_i + \alpha (z^i_{t-1} - \mu_i), \sigma_i^2 (1 - \alpha^2)] \quad \text{(over-relaxation)}
\]

▶ Gaussian reverse model \( r_t(z_{t-1} | z_t) \), linear dependence on \( z_t \)
Gaussian example

Bivariate Gaussian target distribution:

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

Sample \( z^1, z^2 \) in turn using

\[ q(z^i_t|z_{t-1}) = p(z^i_t|z^{-i}) = N(\mu_i, \sigma_i^2) \quad \text{(Gibbs sampling)} \]

or

\[ q(z^i_t|z_{t-1}) = N[\mu_i + \alpha(z^i_{t-1} - \mu_i), \sigma_i^2(1-\alpha^2)] \quad \text{(over-relaxation)} \]

- Gaussian reverse model \( r_t(z_{t-1}|z_t) \), linear dependence on \( z_t \)
- Maximize variational lower bound w.r.t. \( \alpha \)
Gaussian example

Gibbs sampling versus over-relaxation [1] for a bivariate Gaussian.

The improved mixing of over-relaxation results in an improved variational lower bound.
Hamiltonian Dynamics

\[ H(u, z_t) = 0 \]

\[ u_t^{\text{halfway}} = u_t^{\text{init}} + \gamma \frac{1}{2} \nabla_{z_t} \log p(x, z_t) \]

\[ z_t^{+1} = z_t + \gamma M^{-1} u_t^{\text{halfway}} \]

\[ u_t^{\text{final}} = u_t^{\text{halfway}} + \gamma \frac{1}{2} \nabla_{z_t^{+1}} \log p(x, z_t^{+1}) \]
Hamiltonian Dynamics

Construct $q_t(z_t|z_{t-1}, x)$ by simulating Hamiltonian dynamics, using Hamiltonian

$$H(u, z) = 0.5u^T M^{-1} u - \log p(x, z)$$

with momentum variables $u$. 
Hamiltonian Dynamics

Construct \( q_t(z_t|z_{t-1}, x) \) by simulating Hamiltonian dynamics, using Hamiltonian

\[
H(u, z) = 0.5 u^T M^{-1} u - \log p(x, z)
\]

with momentum variables \( u \).
Initialize momentum \( u_{t,\text{init}} \sim q_t(u_{t,\text{init}}|z_{t-1}, x) \)
Hamiltonian Dynamics

Construct $q_t(z_t|z_{t-1}, x)$ by simulating Hamiltonian dynamics, using Hamiltonian

$$H(u, z) = 0.5 u^T M^{-1} u - \log p(x, z)$$

with momentum variables $u$.

Initialize momentum $u_{t, \text{init}} \sim q_t(u_{t, \text{init}}|z_{t-1}, x)$

Simulate Hamiltonian using the *leapfrog integrator*:

$$u_{t, \text{halfway}} = u_{t, \text{init}} + \frac{\gamma}{2} \nabla z_t \log[p(x, z_t)]$$

$$z_{t+1} = z_t + \gamma M^{-1} u_{t, \text{halfway}}$$

$$u_{t, \text{final}} = u_{t, \text{halfway}} + \frac{\gamma}{2} \nabla z_{t+1} \log[p(x, z_{t+1})]$$
Hamiltonian Variational Inference

Require:
- Momentum initialization distribution \( q_t(u_t, \text{init} | z_{t-1}, x) \), and reverse model \( r_t(u_t, \text{final} | z_t, x) \).
- HMC stepsize and mass matrix \( \gamma, M \).

1. Draw initial random variable \( z_0 \sim q(z_0 | x) \).
2. Initialize lower bound \( \hat{L} = \log[p(x, z_0)] - \log[q(z_0 | x)] \).
3. For \( t = 1 : T \):
   - Draw initial momentum \( u_t, \text{init} \sim q_t(u_t, \text{init} | x, z_{t-1}) \).
   - Set \( z_t, u_t, \text{final} = \text{Hamiltonian Dynamics}(z_{t-1}, u_t, \text{init}) \).
   - Calculate ratio \( \alpha_t = \frac{p(x, z_t) r_t(u_t, \text{final} | x, z_t)}{p(x, z_{t-1}) q_t(u_t, \text{init} | x, z_{t-1})} \).
   - Update lower bound \( \hat{L} \leftarrow \hat{L} + \log[\alpha_t] \).
4. Return lower bound estimate \( \hat{L} \).

- No rejection step, to keep everything differentiable.
- Optimize lower bound w.r.t. \( \gamma, M \) and all parameters in \( q, r \).
- Differentiate through the leapfrog integrator.
Hamiltonian Variational Inference

**Require:** Momentum initialization distribution \( q_t(u_{t,\text{init}}|z_{t-1}, x) \), and reverse model \( r_t(u_{t,\text{final}}|z_t, x) \)

**Require:** HMC stepsize and mass matrix \( \gamma, M \)
Hamiltonian Variational Inference

**Require:** Momentum initialization distribution \( q_t(u_{t,\text{init}}|z_{t-1}, x) \), and reverse model \( r_t(u_{t,\text{final}}|z_t, x) \)

**Require:** HMC stepsize and mass matrix \( \gamma, M \)

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

Init. lower bound \( \hat{L} = \log[p(x, z_0)] - \log[q(z_0|x)] \)
Hamiltonian Variational Inference

**Require:** Momentum initialization distribution \( q_t(u_t, \text{init} | z_{t-1}, x) \), and reverse model \( r_t(u_t, \text{final} | z_t, x) \)

**Require:** HMC stepsize and mass matrix \( \gamma, M \)

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

Init. lower bound \( \hat{L} = \log[p(x, z_0)] - \log[q(z_0 | x)] \)

**for** \( t = 1 : T \) **do**

Draw initial momentum \( u_{t, \text{init}} \sim q_t(u_{t, \text{init}} | x, z_{t-1}) \)
Hamiltonian Variational Inference

Require: Momentum initialization distribution $q_t(u_{t,\text{init}}|z_{t-1}, x)$, and reverse model $r_t(u_{t,\text{final}}|z_t, x)$

Require: HMC stepsize and mass matrix $\gamma, M$

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

Init. lower bound $\hat{L} = \log[p(x, z_0)] - \log[q(z_0|x)]$

for $t = 1 : T$ do
  Draw initial momentum $u_{t,\text{init}} \sim q_t(u_{t,\text{init}}|x, z_{t-1})$
  Set $z_t, u_{t,\text{final}} = \text{Hamiltonian Dynamics}(z_{t-1}, u_{t,\text{init}})$
end for

• No rejection step, to keep everything differentiable
• Optimize lower bound w.r.t. $\gamma, M$ and all parameters in $q, r$
Hamiltonian Variational Inference

**Require:** Momentum initialization distribution \( q_t(u_{t,\text{init}}|z_{t-1}, x) \), and reverse model \( r_t(u_{t,\text{final}}|z_t, x) \)

**Require:** HMC stepsize and mass matrix \( \gamma, M \)

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

Init. lower bound \( \hat{L} = \log[p(x, z_0)] - \log[q(z_0|x)] \)

for \( t = 1 : T \) do

Draw initial momentum \( u_{t,\text{init}} \sim q_t(u_{t,\text{init}}|x, z_{t-1}) \)

Set \( z_t, u_{t,\text{final}} = \text{Hamiltonian\_Dynamics}(z_{t-1}, u_{t,\text{init}}) \)

Calculate ratio \( \alpha_t = \frac{p(x, z_t)r_t(u_{t,\text{final}}|x, z_t)}{p(x, z_{t-1})q_t(u_{t,\text{init}}|x, z_{t-1})} \)
Hamiltonian Variational Inference

**Require:** Momentum initialization distribution $q_t(u_{t,\text{init}}|z_{t-1}, x)$, and reverse model $r_t(u_{t,\text{final}}|z_t, x)$

**Require:** HMC stepsize and mass matrix $\gamma, M$

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

Init. lower bound $\hat{L} = \log[p(x, z_0)] - \log[q(z_0|x)]$

for $t = 1 : T$ do

Draw initial momentum $u_{t,\text{init}} \sim q_t(u_{t,\text{init}}|x, z_{t-1})$

Set $z_t, u_{t,\text{final}} = \text{Hamiltonian\_Dynamics}(z_{t-1}, u_{t,\text{init}})$

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

Update lower bound $\hat{L} \leftarrow \hat{L} + \log[\alpha_t]$

end for
Hamiltonian Variational Inference

Require: Momentum initialization distribution $q_t(u_{t, \text{init}}|z_{t-1}, x)$, and reverse model $r_t(u_{t, \text{final}}|z_t, x)$

Require: HMC stepsize and mass matrix $\gamma, M$

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

Init. lower bound $\hat{L} = \log[p(x, z_0)] - \log[q(z_0|x)]$

for $t = 1 : T$ do

Draw initial momentum $u_{t, \text{init}} \sim q_t(u_{t, \text{init}}|x, z_{t-1})$

Set $z_t, u_{t, \text{final}} = \text{Hamiltonian\_Dynamics}(z_{t-1}, u_{t, \text{init}})$

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

Update lower bound $\hat{L} \leftarrow \hat{L} + \log[\alpha_t]$

end for

return lower bound estimate $\hat{L}$
Hamiltonian Variational Inference

**Require:** Momentum initialization distribution $q_t(u_{t,\text{init}}|z_{t-1}, x)$, and reverse model $r_t(u_{t,\text{final}}|z_t, x)$

**Require:** HMC stepsize and mass matrix $\gamma, M$

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

Init. lower bound $\hat{L} = \log[p(x, z_0)] - \log[q(z_0|x)]$

for $t = 1 : T$ do

- Draw initial momentum $u_{t,\text{init}} \sim q_t(u_{t,\text{init}}|x, z_{t-1})$

- Set $z_t, u_{t,\text{final}} =$ Hamiltonian Dynamics$(z_{t-1}, u_{t,\text{init}})$

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

- Update lower bound $\hat{L} \leftarrow \hat{L} + \log[\alpha_t]$

end for

return lower bound estimate $\hat{L}$

- No rejection step, to keep everything differentiable
Hamiltonian Variational Inference

**Require:** Momentum initialization distribution $q_t(u_{t,\text{init}}|z_{t-1}, x)$, and reverse model $r_t(u_{t,\text{final}}|z_t, x)$

**Require:** HMC stepsize and mass matrix $\gamma, M$

Draw initial random variable $z_0 \sim q(z_0|x)$
Init. lower bound $\hat{L} = \log[p(x, z_0)] - \log[q(z_0|x)]$

for $t = 1 : T$

Draw initial momentum $u_{t,\text{init}} \sim q_t(u_{t,\text{init}}|x, z_{t-1})$
Set $z_t, u_{t,\text{final}} = \text{Hamiltonian\_Dynamics}(z_{t-1}, u_{t,\text{init}})$
Calculate ratio $\alpha_t = \frac{p(x, z_t)r_t(u_{t,\text{final}}|x, z_t)}{p(x, z_{t-1})q_t(u_{t,\text{init}}|x, z_{t-1})}$
Update lower bound $\hat{L} \leftarrow \hat{L} + \log[\alpha_t]$

end for

**Return** lower bound estimate $\hat{L}$

- No rejection step, to keep everything differentiable
- Optimize lower bound w.r.t. $\gamma, M$ and all parameters in $q, r$
Hamiltonian Variational Inference

**Require:** Momentum initialization distribution \( q_t(u_{t,\text{init}}|z_{t-1}, x) \), and reverse model \( r_t(u_{t,\text{final}}|z_t, x) \)

**Require:** HMC stepsize and mass matrix \( \gamma, M \)

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

Init. lower bound \( \hat{L} = \log[p(x, z_0)] - \log[q(z_0|x)] \)

**for** \( t = 1 : T \) **do**

- Draw initial momentum \( u_{t,\text{init}} \sim q_t(u_{t,\text{init}}|x, z_{t-1}) \)
- Set \( z_t, u_{t,\text{final}} = \text{Hamiltonian\_Dynamics}(z_{t-1}, u_{t,\text{init}}) \)
- Calculate ratio \( \alpha_t = \frac{p(x, z_t)r_t(u_{t,\text{final}}|x, z_t)}{p(x, z_{t-1})q_t(u_{t,\text{init}}|x, z_{t-1})} \)
- Update lower bound \( \hat{L} \leftarrow \hat{L} + \log[\alpha_t] \)

**end for**

**return** lower bound estimate \( \hat{L} \)

- No rejection step, to keep everything differentiable
- Optimize lower bound w.r.t. \( \gamma, M \) and all parameters in \( q, r \)
- Differentiate through the leapfrog integrator
Low dimensional example: Overdispersed Counts

Simple **2-dimensional** model [2], simple specification $q, r$. One step of Hamiltonian dynamics, **varying number of leapfrog steps**.

Contour plots approximate posterior. Exact posterior at bottom-right.  

R-squared accuracy of approximation [3]
Generative model for MNIST

Variational autoencoder for binarized MNIST [4], spherical Gaussian prior \( p(z) = \mathcal{N}(0, \mathbf{I}) \), MLP conditional likelihood \( p_{\theta}(x|z) \).

<table>
<thead>
<tr>
<th>Model</th>
<th>(-L)</th>
<th>(-\log p(x))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results with ( q(z_0</td>
<td>x) = \mathcal{N}(\mu, \sigma^2 \mathbf{I}) ):</td>
<td></td>
</tr>
<tr>
<td>5 leapfrog steps</td>
<td>90.86</td>
<td>87.16</td>
</tr>
<tr>
<td>10 leapfrog steps</td>
<td>87.60</td>
<td>85.56</td>
</tr>
<tr>
<td>With ( q(z_0</td>
<td>x) = ) inference network:</td>
<td></td>
</tr>
<tr>
<td>No leapfrog steps</td>
<td>94.18</td>
<td>88.95</td>
</tr>
<tr>
<td>1 leapfrog step</td>
<td>91.70</td>
<td>88.08</td>
</tr>
<tr>
<td>4 leapfrog steps</td>
<td>89.82</td>
<td>86.40</td>
</tr>
<tr>
<td>8 leapfrog steps</td>
<td>88.30</td>
<td>85.51</td>
</tr>
</tbody>
</table>

- MCMC makes bound tighter, gives better marginal likelihood
- MCMC also works with simple initialization
Generative model for MNIST

Variational autoencoder for binarized MNIST [4], spherical Gaussian prior $p(z) = \mathcal{N}(0, I)$, MLP conditional likelihood $p_\theta(x|z)$.

<table>
<thead>
<tr>
<th>Model</th>
<th>$-L$</th>
<th>$-\log p(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results with $q(z_0</td>
<td>x) = \mathcal{N}(\mu, \sigma^2 I)$:</td>
<td></td>
</tr>
<tr>
<td>5 leapfrog steps</td>
<td>90.86</td>
<td>87.16</td>
</tr>
<tr>
<td>10 leapfrog steps</td>
<td>87.60</td>
<td>85.56</td>
</tr>
<tr>
<td>With $q(z_0</td>
<td>x) = \text{inference network}$:</td>
<td></td>
</tr>
<tr>
<td>No leapfrog steps</td>
<td><strong>94.18</strong></td>
<td><strong>88.95</strong></td>
</tr>
<tr>
<td>1 leapfrog step</td>
<td>91.70</td>
<td>88.08</td>
</tr>
<tr>
<td>4 leapfrog steps</td>
<td>89.82</td>
<td>86.40</td>
</tr>
<tr>
<td>8 leapfrog steps</td>
<td><strong>88.30</strong></td>
<td><strong>85.51</strong></td>
</tr>
</tbody>
</table>

- MCMC makes bound tighter, gives better marginal likelihood.
Generative model for MNIST

Variational autoencoder for binarized MNIST [4], spherical Gaussian prior $p(z) = \mathcal{N}(0, I)$, MLP conditional likelihood $p_\theta(x|z)$.

<table>
<thead>
<tr>
<th>Model</th>
<th>$-L$</th>
<th>$-\log p(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results with $q(z_0</td>
<td>x) = \mathcal{N} (\mu, \sigma^2 I)$:</td>
<td></td>
</tr>
<tr>
<td>5 leapfrog steps</td>
<td>90.86</td>
<td>87.16</td>
</tr>
<tr>
<td>10 leapfrog steps</td>
<td>87.60</td>
<td>85.56</td>
</tr>
<tr>
<td>With $q(z_0</td>
<td>x) = \text{inference network}$:</td>
<td></td>
</tr>
<tr>
<td>No leapfrog steps</td>
<td>94.18</td>
<td>88.95</td>
</tr>
<tr>
<td>1 leapfrog step</td>
<td>91.70</td>
<td>88.08</td>
</tr>
<tr>
<td>4 leapfrog steps</td>
<td>89.82</td>
<td>86.40</td>
</tr>
<tr>
<td>8 leapfrog steps</td>
<td>88.30</td>
<td>85.51</td>
</tr>
</tbody>
</table>

- MCMC makes bound tighter, gives better marginal likelihood
- MCMC also works with simple initialization
Generative model for MNIST

Hamiltonian variational inference also works well with convolutional autoencoders.

<table>
<thead>
<tr>
<th>Model</th>
<th>(-L)</th>
<th>(-\log p(x))</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Convolutional decoder and inference network:</em></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No leapfrog steps</td>
<td>86.66</td>
<td>83.20</td>
</tr>
<tr>
<td>1 leapfrog step</td>
<td>85.40</td>
<td>82.98</td>
</tr>
<tr>
<td>2 leapfrog steps</td>
<td>85.17</td>
<td>82.96</td>
</tr>
<tr>
<td>4 leapfrog steps</td>
<td>84.94</td>
<td>82.78</td>
</tr>
<tr>
<td>8 leapfrog steps</td>
<td>84.81</td>
<td>82.72</td>
</tr>
<tr>
<td>16 leapfrog steps</td>
<td>84.11</td>
<td>82.22</td>
</tr>
<tr>
<td>16 leapfrog steps, (n_h = 800)</td>
<td>83.49</td>
<td>81.94</td>
</tr>
</tbody>
</table>
Conclusion

- MCMC improves variational approximation
- MCMC kernels automatically adapt to target $p(z|x)$
- Approximation not restricted to standard exponential family distributions
- More MCMC steps = slower iterations, but fewer iterations needed for convergence
- Optimizing variational bound improves MCMC
- Automatic tuning, convergence assessment, independent sampling, no rejections
- Learn MCMC transitions $q_t(z_t|z_{t-1}, x)$
- Optimize initialization $q(z_0|x)$
- Many possibilities left to explore
Conclusion

- MCMC improves variational approximation
Conclusion

- MCMC improves variational approximation
  - MCMC kernels automatically adapt to target $p(z|x)$
Conclusion

- MCMC improves variational approximation
  - MCMC kernels automatically adapt to target $p(z|x)$
  - Approximation not restricted to standard exponential family distributions
Conclusion

- MCMC improves variational approximation
  - MCMC kernels automatically adapt to target $p(z|x)$
  - Approximation not restricted to standard exponential family distributions
  - More MCMC steps = slower iterations, but fewer iterations needed for convergence
Conclusion

- MCMC improves variational approximation
  - MCMC kernels automatically adapt to target $p(z|x)$
  - Approximation not restricted to standard exponential family distributions
  - More MCMC steps = slower iterations, but fewer iterations needed for convergence
- Optimizing variational bound improves MCMC
Conclusion

- MCMC improves variational approximation
  - MCMC kernels automatically adapt to target $p(z|x)$
  - Approximation not restricted to standard exponential family distributions
  - More MCMC steps $=$ slower iterations, but fewer iterations needed for convergence
- Optimizing variational bound improves MCMC
  - Automatic tuning, convergence assessment, independent sampling, no rejections
Conclusion

- MCMC improves variational approximation
  - MCMC kernels automatically adapt to target $p(z|x)$
  - Approximation not restricted to standard exponential family distributions
  - More MCMC steps = slower iterations, but fewer iterations needed for convergence
- Optimizing variational bound improves MCMC
  - Automatic tuning, convergence assessment, independent sampling, no rejections
  - Learn MCMC transitions $q_t(z_t|z_{t-1}, x)$
Conclusion

- MCMC improves variational approximation
  - MCMC kernels automatically adapt to target $p(z|x)$
  - Approximation not restricted to standard exponential family distributions
  - More MCMC steps = slower iterations, but fewer iterations needed for convergence
- Optimizing variational bound improves MCMC
  - Automatic tuning, convergence assessment, independent sampling, no rejections
  - Learn MCMC transitions $q_t(z_t|z_{t-1}, x)$
  - Optimize initialization $q(z_0|x)$

Many possibilities left to explore
Conclusion

- MCMC improves variational approximation
  - MCMC kernels automatically adapt to target $p(z|x)$
  - Approximation not restricted to standard exponential family distributions
  - More MCMC steps = slower iterations, but fewer iterations needed for convergence
- Optimizing variational bound improves MCMC
  - Automatic tuning, convergence assessment, independent sampling, no rejections
  - Learn MCMC transitions $q_t(z_t|z_{t-1}, x)$
  - Optimize initialization $q(z_0|x)$
- Many possibilities left to explore
References

Stephen L Adler.

Jim Albert.

Tim Salimans and David A Knowles.

Diederik P Kingma and Max Welling.