12: Computationally Efficient Markov chain Simulation

Taylor

University of Virginia

Introduction

We mention:

1. a few tuning tips for Random-Walk Metropolis-Hastings
2. Metropolis-adjusted Langevin Algorithm (MALA)
3. Hamiltonian Monte Carlo (HMC)
4. Pseudo-Marginal Metropolis-Hastings (PMMH).

Random Walk M-H: Some Tricks

Last section, when we were using the Metropolis-Hastings algorithm, we struggled with tuning our proposal's covariance matrix:

$$q(\theta^* \mid \theta^{t-1}) = \text{Normal}(\theta^{t-1}, \Sigma).$$

The book recommends setting

$$\Sigma \approx \frac{2.4^2}{d} \text{Var}(\theta \mid y)$$

after transforming θ be roughly normal. Here d is the dimension of θ. A rough approximation of the posterior covariance matrix is required.

The book also recommends aiming for an acceptance rate of about 22% for problems where $d > 5$.

MALA

The **Metropolis-adjusted Langevin Algorithm** is a Metropolis-Hastings algorithm with a special proposal distribution:

$$q(\theta^* \mid \theta^{t-1}) = \text{Normal}(\theta^{t-1} + \frac{\sigma_1^2}{2}\nabla \log p(\theta \mid y), \sigma_1^2 D).$$

It jumps more often in the direction of the mode, and is usually more efficient than regular Random Walk Metropolis-Hastings.

Tune $\sigma_1^2 > 0, D$, hopefully have an acceptance rate around (.574) (source)

HMC

Hamiltonian Monte Carlo can be quite effective at sampling from a high-dimensional posterior. It makes use of the derivative of the log-likelihood as well.

We will describe it in three steps:

1. Describing Hamiltonian dynamics in continuous time
2. Describing how to discretize Hamiltonian dynamics
3. Describing how to use these in a proposal distribution in the Metropolis-Hastings algorithm.

HMC

Hamiltonian Monte Carlo can be quite effective at sampling from a high-dimensional posterior. It makes use of the derivative of the log-likelihood as well.

We will describe it in three steps:

1. Describing Hamiltonian dynamics in continuous time
2. Describing how to discretize Hamiltonian dynamics
3. Describing how to use these in a proposal distribution in the Metropolis-Hastings algorithm.

Video time!

1. `https://www.youtube.com/watch?v=mzjErXqBXw4`
2. `https://www.youtube.com/watch?v=87E0DKs5bok`

HMC

Say you have $\theta_1, \ldots, \theta_d$. You add d auxiliary variables: $\phi_1, \ldots, \phi_d$.

It's customary to use the notation $q_1, \ldots, q_d$ (the positions), and $p_1, \ldots, p_d$ (the momenta).

HMC

Say you have $\theta_1, \ldots, \theta_d$. You add d auxiliary variables: $\phi_1, \ldots, \phi_d$.

It's customary to use the notation $q_1, \ldots, q_d$ (the positions), and $p_1, \ldots, p_d$ (the momenta).

A HMC proposal follows a two-step procedure:

1. sample a random momentum vector
2. transform the momentum and position nonrandomly using Hamilton's equations

Both steps are transition kernels that preserve the stationary distribution.

HMC: a 1-d example

Start with one-dimensional q (position) and one-dimensional p (momentum). Also, m is mass (a tuning parameter).

1. potential energy: $U(q)$ is negative the logarithm of the unnormalized posterior.
2. kinetic energy: $K(p) = \frac{p^2}{2m}$

$p = m \times$ velocity

Two good resources:

1. `https://arxiv.org/pdf/1206.1901.pdf` (primary reference),
2. `https://arxiv.org/pdf/1701.02434.pdf`

HMC

When the particle goes up the hill, it loses kinetic energy, and gains potential energy.

Define the **Hamiltonian** as

$$H(q, p) = U(q) + K(p).$$

and define **Hamilton's Equations** as

$$\frac{dq}{dt} = \frac{\partial H(q, p)}{\partial p} \tag{1}$$

$$\frac{dp}{dt} = -\frac{\partial H(q, p)}{\partial q} \tag{2}$$

HMC: a first example

Assume the posterior is a Gaussian with mean $y = 0$ and variance 1. Negative log of the posterior is proportional to

$$U(q) = \frac{q^2}{2}.$$

Also, assume kinetic energy is of the form

$$K(p) = \frac{p^2}{2m}.$$

HMC: a first example

Assume the posterior is a Gaussian with mean $y = 0$ and variance 1. Negative log of the posterior is proportional to

$$U(q) = \frac{q^2}{2}.$$

Also, assume kinetic energy is of the form

$$K(p) = \frac{p^2}{2m}.$$

so

$$\frac{dq}{dt} = \frac{\partial H(q,p)}{\partial p} = \frac{dK(p)}{dp} = p(t)/m \tag{3}$$

$$\frac{dp}{dt} = -\frac{\partial H(q,p)}{\partial q} = -\frac{dU(q)}{dq} = -q(t) \tag{4}$$

HMC: a first example

If $m = 1$ integrating

$$\frac{dq}{dt} = \frac{\partial H(q,p)}{\partial p} = \frac{dK(p)}{dp} = p(t) \tag{5}$$

$$\frac{dp}{dt} = -\frac{\partial H(q,p)}{\partial q} = -\frac{dU(q)}{dq} = -q(t) \tag{6}$$

with respect to time yields

$$q(t) = r\cos(a+t) \tag{7}$$

$$p(t) = -r\sin(a+t) \tag{8}$$

HMC: a first example

For this particular model, $(q, p)'$ rotates clockwise in phase-space.

$$q(t) = r\cos(a + t) \tag{9}$$

$$p(t) = -r\sin(a + t) \tag{10}$$

Can be written as

$$\begin{bmatrix} r\cos(a+t) \\ -r\sin(a+t) \end{bmatrix} = \underbrace{\begin{bmatrix} \cos([t-s]) & \sin([t-s]) \\ \sin([t-s]) & \cos([t-s]) \end{bmatrix}}_{T_{t-s}} \begin{bmatrix} r\cos(a+s) \\ -r\sin(a+t) \end{bmatrix}$$

(use Angle-Sum trig identity)

HMC: a first example

When $m \neq 1$, T_s might look like this.

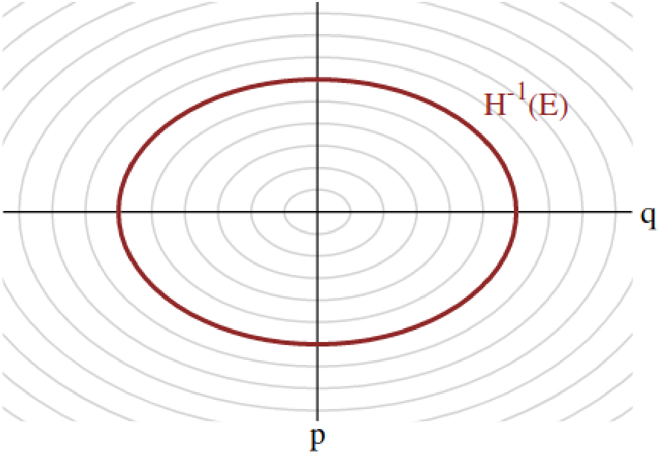


image source: `https://arxiv.org/pdf/1701.02434.pdf`

HMC

Now assume d-dimensional posterior $\mathbf{q} = (q_1, \ldots, q_d)$ and $\mathbf{p} = (p_1, \ldots, p_d)$.

When

$$K(\mathbf{p}) = \frac{\mathbf{p}' M^{-1} \mathbf{p}}{2}$$

Hamilton's equations become

$$\frac{dq_i}{dt} = \frac{\partial H(\mathbf{q}, \mathbf{p})}{\partial p_i} = \frac{dK(\mathbf{p})}{dp_i} = [M^{-1}\mathbf{p}]_i \tag{11}$$

$$\frac{dp_i}{dt} = -\frac{\partial H(\mathbf{q}, \mathbf{p})}{\partial q_i} = -\frac{dU(\mathbf{q})}{dq_i} \tag{12}$$

Recall that $U(\mathbf{q})$ is the negative log of the posterior you're interested in.

HMC Property 1: Reversibility of T_s

Recall $T_s : [\mathbf{q}(0), \mathbf{p}(0)] \mapsto [\mathbf{q}(s), \mathbf{p}(s)]$. It is always has an easy-to-find inverse.

Proof: just take the negative of the derivatives.

HMC Property 2: Conservation of the Hamiltonian

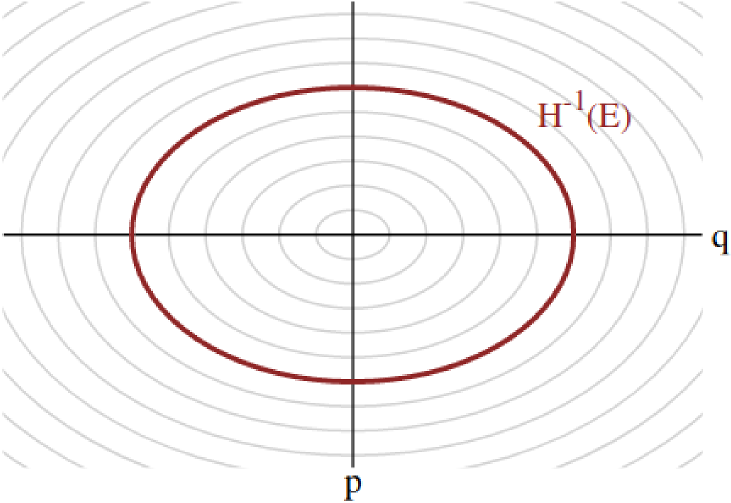
Using the chain rule:

$$\begin{aligned}
\frac{dH}{dt} &= \sum_{i=1}^{d} \frac{\partial H}{\partial p_i}\frac{dp_i}{dt} + \sum_{i=1}^{d} \frac{\partial H}{\partial q_i}\frac{dq_i}{dt} \\
&= \sum_{i=1}^{d} \frac{dK}{dp_i}\frac{dp_i}{dt} + \sum_{i=1}^{d} \frac{dU}{dq_i}\frac{dq_i}{dt} \\
&= \sum_{i=1}^{d} \frac{dK}{dp_i}\left(-\frac{dU}{dq_i}\right) + \sum_{i=1}^{d} \frac{dU}{dq_i}\frac{dK}{dp_i} \\
&= 0
\end{aligned}$$

Moving through time keeps you on the same contour or level-set in the phase space.

HMC

T_s keeps you on a level-set/contour:



HMC Property 3 and 4: Volume Preservation and Symplecticness

TODO

HMC: looking back at the big picture

Again, HMC will work as follows: given that we are currently at position $\mathbf{q}(t)$, we are going to sample a momentum vector (which puts us on one of the level-sets), and then we are going to follow T_s for a deterministic amount of time.

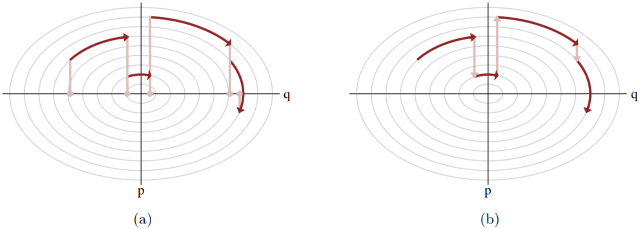


FIG 22. *(a) Each Hamiltonian Markov transition lifts the initial state onto a random level set of the Hamiltonian, which can then be explored with a Hamiltonian trajectory before projecting back down to the target parameter space. (b) If we consider the projection and random lift steps as a single momentum resampling step, then the Hamiltonian Markov chain alternates between deterministic trajectories along these level sets (dark red) and a random walk across the level sets (light red).*

HMC: looking back at the big picture

Again, HMC will work as follows: given that we are currently at position $\mathbf{q}(t)$, we are going to sample a momentum vector (which puts us on one of the level-sets), and then we are going to follow T_s for a deterministic amount of time.

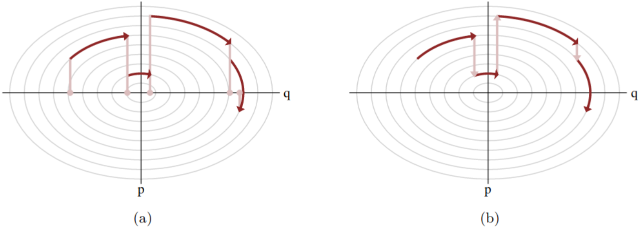


Fig 22. *(a) Each Hamiltonian Markov transition lifts the initial state onto a random level set of the Hamiltonian, which can then be explored with a Hamiltonian trajectory before projecting back down to the target parameter space. (b) If we consider the projection and random lift steps as a single momentum resampling step, then the Hamiltonian Markov chain alternates between deterministic trajectories along these level sets (dark red) and a random walk across the level sets (light red).*

Following a contour line is impossible in continuous time though...

Discretizing Hamilton's Equations: Version 1.0

We need to be able to approximate T_s using the derivatives. To do that, we pick a small change in time called ϵ. Then we take L steps of size ϵ.

Two procedures are described. The last one is the one that is most commonly used.

For simplicity, assume the mass matrix is diagonal, making

$$K(\mathbf{p}) = \mathbf{p}' M^{-1} \mathbf{p} = \sum_{i=1}^{d} \frac{p_i^2}{2m_i}.$$

Discretizing Hamilton's Equations: Version 1.0

When $K(\mathbf{p}) = \sum_{i=1}^{d} \frac{p_i^2}{2m_i}$, **Euler's method** approximates

$$\frac{dq_i}{dt} = \frac{\partial H(\mathbf{q}, \mathbf{p})}{\partial p_i} = \frac{dK(\mathbf{p})}{dp_i} = p_i / m_i \tag{13}$$

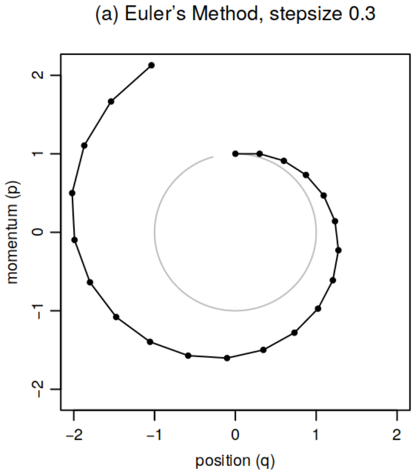
$$\frac{dp_i}{dt} = -\frac{\partial H(\mathbf{q}, \mathbf{p})}{\partial q_i} = -\frac{dU(\mathbf{q})}{dq_i} \tag{14}$$

with

$$q_i(t + \epsilon) = q_i(t) + \epsilon p_i(t) / m_i \tag{15}$$

$$p_i(t + \epsilon) = p_i(t) - \epsilon \frac{dU}{dq_i}(\mathbf{q}(t)) \tag{16}$$

Discretizing Hamilton's Equations: Version 1.0



Twenty steps when $H(q,p) = p^2/2 + q^2/2$, the initial state is $(q,p) = (0,1)$. Leap-frog is better because it preserves the volume!

Discretizing Hamilton's Equations: Version 2.0

When $K(\mathbf{p}) = \sum_{i=1}^{d} \frac{p_i^2}{2m_i}$, **the leap-frog method** approximates

$$\frac{dq_i}{dt} = \frac{\partial H(\mathbf{q},\mathbf{p})}{\partial p_i} = \frac{dK(\mathbf{p})}{dp_i} = p_i/m_i \tag{17}$$

$$\frac{dp_i}{dt} = -\frac{\partial H(\mathbf{q},\mathbf{p})}{\partial q_i} = -\frac{dU(\mathbf{q})}{dq_i} \tag{18}$$

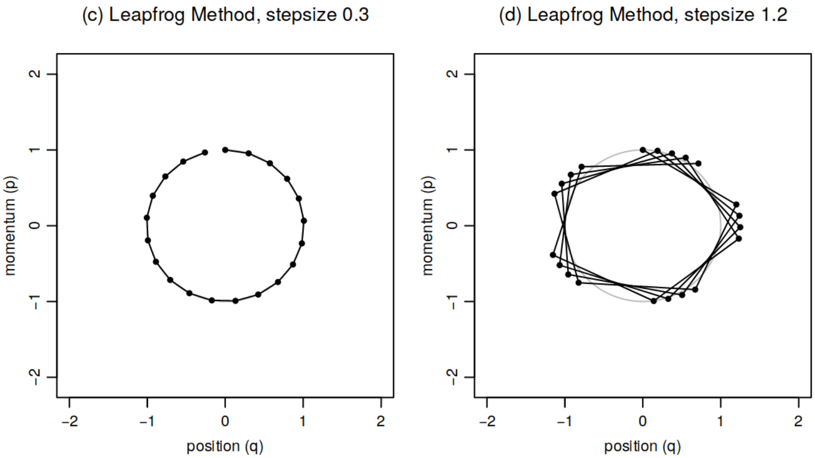
with

$$p_i(t+\epsilon/2) = p_i(t) - (\epsilon/2)\frac{dU}{dq_i}(\mathbf{q}(t)) \tag{19}$$

$$q_i(t+\epsilon) = q_i(t) + \epsilon\frac{p_i(t+\epsilon/2)}{m_i} \tag{20}$$

$$p_i(t+\epsilon) = p_i(t+\epsilon/2) - (\epsilon/2)\frac{dU}{dq_i}(\mathbf{q}(t+\epsilon)) \tag{21}$$

Discretizing Hamilton's Equations: Version 2.0



Twenty steps when $H(q,p) = p^2/2 + q^2/2$, the initial state is $(q,p) = (0,1)$.

Describing the HMC algorithm

The algorithm targets the distribution for $(\mathbf{q}, \mathbf{p})$:

$$
\begin{aligned}
\frac{1}{Z} \exp\left[-\frac{H(\mathbf{q}, \mathbf{p})}{T}\right] &= \frac{1}{Z} \exp\left[-\frac{K(\mathbf{p}) + U(\mathbf{q})}{T}\right] \\
&= \frac{1}{Z} \exp\left[-\frac{K(\mathbf{p})}{T}\right] \exp\left[-\frac{U(\mathbf{q})}{T}\right] \\
&= \frac{1}{Z} \exp\left[-\frac{K(\mathbf{p})}{T}\right] \times \\
&\qquad \exp\left[-\frac{-\log\{\text{prior}(\mathbf{q}) \times \text{likelihood}(\mathbf{q})\}}{T}\right]
\end{aligned}
$$

Describing the HMC algorithm

Step 1:

Sample p from the conditional target distribution

$$\frac{1}{Z}\exp\left[-\frac{K(\mathbf{p})}{T}\right].$$

In our case, this is the same as the marginal, due to independence.

Notice how this is a Gibbs-like step! It preserves the stationary distribution, and it has 100% chance of being accepted.

Describing the HMC algorithm

Step 2:

If we could integrate Hamilton's equations exactly, then our proposal would be deterministic, and we would accept with probability 1. However, because we are using numerical leap-frog integration, there will be some deviation from the ideal point you end up at. We think of the L leap-frog steps as a proposal distribution. This is a deterministic proposal, and it's symmetrical. So what we end up with is a Metropolis-like acceptance probability:

$$\min\left[1, \frac{\exp\left[-H(\mathbf{q}^*, \mathbf{p}^*)\right]}{\exp\left[-H(\mathbf{q}, \mathbf{p})\right]}\right]$$

Recall that the proposal distributions cancel in this expression because the proposals are symmetric. Also, if we were integrating exactly, the Hamiltonian wouldn't change, and this would simplify to 1.

Describing the HMC algorithm

Code from `https://arxiv.org/pdf/1206.1901.pdf` that performs one iteration of HMC can be found in the file `hmc.r`.

Here's a visualization:
`https://chi-feng.github.io/mcmc-demo/`

Pseudo-Marginal Metropolis-Hastings

Sometimes you are not able to evaluate the likelihood at all! For example, with models with a lot of hidden data, the likelihood is a high-dimensional integral or sum that may be either impossible or computationally difficult to evaluate.

Pseudo-Marginal Metropolis-Hastings

Sometimes you are not able to evaluate the likelihood at all! For example, with models with a lot of hidden data, the likelihood is a high-dimensional integral or sum that may be either impossible or computationally difficult to evaluate.

The marginal Metropolis-Hastings' acceptance ratio is

$$\frac{p(y_{1:T} \mid \theta')p(\theta')}{p(y_{1:T} \mid \theta)p(\theta)} \frac{q(\theta \mid \theta')}{q(\theta' \mid \theta)},$$

and the **pseudo-marginal Metropolis-Hastings** acceptance ratio is

$$\frac{\hat{p}(y_{1:T} \mid \theta')p(\theta')}{\hat{p}(y_{1:T} \mid \theta)p(\theta)} \frac{q(\theta \mid \theta')}{q(\theta' \mid \theta)}.$$

Pseudo-Marginal Metropolis-Hastings

Sometimes you are not able to evaluate the likelihood at all.

The marginal Metropolis-Hastings' acceptance ratio is

$$\frac{p(y_{1:T} \mid \theta')p(\theta')}{p(y_{1:T} \mid \theta)p(\theta)} \frac{q(\theta \mid \theta')}{q(\theta' \mid \theta)},$$

and the **pseudo-marginal Metropolis-Hastings** acceptance ratio is

$$\frac{\hat{p}(y_{1:T} \mid \theta')p(\theta')}{\hat{p}(y_{1:T} \mid \theta)p(\theta)} \frac{q(\theta \mid \theta')}{q(\theta' \mid \theta)}.$$

Pseudo-Marginal Metropolis-Hastings

Rewrite it a little differently:

$$\frac{\overbrace{\hat{p}(y_{1:T} \mid u', \theta')p(\theta')\psi(u'|y_{1:T}, \theta')}^{\text{unnormalized target}}}{\hat{p}(y_{1:T}|u, \theta)p(\theta)\psi(u|y_{1:T}, \theta)} \underbrace{\frac{\psi(u|y_{1:T}, \theta)q(\theta|\theta')}{\psi(u'|y_{1:T}, \theta')q(\theta'|\theta)}}_{\text{proposal distribution}}$$

- The full normalized target is $p(\theta, u \mid y_{1:T}) = \frac{\hat{p}(y_{1:T}|u,\theta)\psi(u|y_{1:T},\theta)p(\theta|y_{1:T})}{p(y_{1:T}|\theta)}$
- When likelihood is unbiased, the marginal is $p(\theta \mid y_{1:T})$
- u is the collection of all auxiliary random variables (e.g. importance sampling output, a particle filter's samples and ancestor indices, etc.)