Hamiltonian Monte Carlo

Dr. Jarad Niemi

Iowa State University

November 9, 2023

Adapted from Radford Neal's MCMC Using Hamltonian Dynamics in Handbook of Markov Chain Monte Carlo (2011).

Hamiltonian system

Considering a body in a frictionless 1-dimensional environment with

- mass m,
- location θ, and
- momentum $\omega = mv$ where v is velocity.

The mass has

- potential energy $U(\theta)$, proportional to its height, and
- kinetic energy $K(\omega) = \frac{1}{2}mv^2 = \frac{1\omega^2}{2m}$.

Hamilton's equations

Extending this to d dimensions, we have

- position vector θ and
- momentum vector ω.

The Hamiltonian $H(\theta, \omega)$ describes the time evolution of the system through

$$\begin{aligned} \frac{d\theta_i}{dt} &= \frac{\partial H}{\partial \omega_i} \\ \frac{d\omega_i}{dt} &= -\frac{\partial H}{\partial \theta_i} \end{aligned}$$

for $i = 1, \ldots, d$.

Potential and kinetic energy

For Hamiltonian Monte Carlo, we usually use Hamiltonian functions that can be written as follows:

$$H(\theta, \omega) = U(\theta) + K(\omega)$$

where

- $U(\theta)$ is called the potential energy and will be defined to be minus the log probability density of the distribution for θ (plus any constant that is convenient) and
- $K(\omega)$ is called the kinetic energy and is usually defined as

  $$K(\omega) = \omega^\top M^{-1} \omega / 2$$

  where M is a symmetric, positive-definite "mass matrix", which is typically diagonal, and is often a scalar multiple of the identity matrix. This form for $K(\omega)$ corresponds to minus the log probability density (plus a constant) of the zero-mean Gaussian distribution with covariance matrix M.

The resulting Hamilton's equations are

$$\frac{d\theta_i}{dt} = [M^{-1}\omega]_i, \qquad \frac{d\omega_i}{dt} = -\frac{\partial U}{\partial \theta_i}.$$

One-dimensional example

Suppose

$$H(\theta,\omega) = U(\theta) + K(\omega), \quad U(\theta) = \theta^2/2, \quad K(\omega) = \omega^2/2$$

The dynamics resulting from this Hamiltonian are

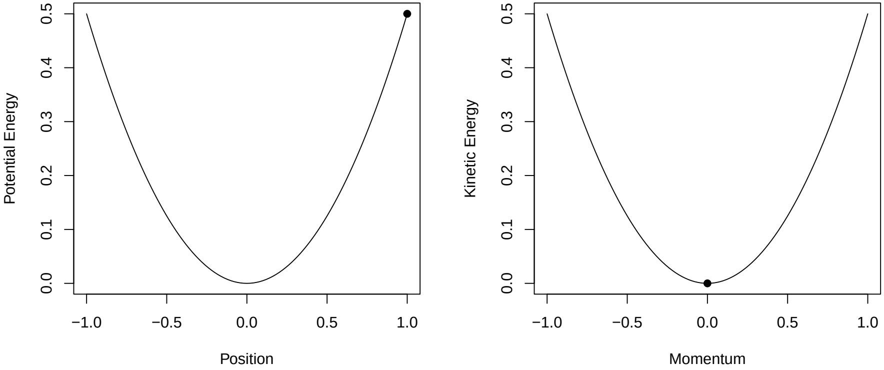
$$\frac{d\theta}{dt} = \omega, \quad \frac{d\omega}{dt} = -\theta.$$

Solutions are of the form

$$\theta(t) = r\, cos(a+t), \quad \omega(t) = -r\, sin(a+t)$$

for some constants r and a that depend on initial conditions.

One-dimensional example simulation

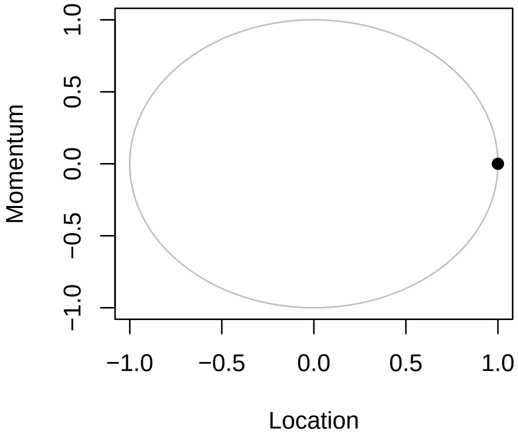


Conservation of the Hamiltonian

This simple model clearly conserves the Hamiltonian:

$$\begin{aligned}
H(\theta,\omega) &= U(\theta) + K(\omega) \\
&= \frac{\theta^2}{2} + \frac{\omega^2}{2} \\
&= \frac{[r\, cos(a+t)]^2}{2} + \frac{[-r\, sin(a+t)]^2}{2} \\
&= \frac{r^2}{2}\left[cos^2(a+t) + sin^2(a+t)\right] \\
&= r^2/2
\end{aligned}$$

Conservation of the Hamiltonian



Conservation of the Hamiltonian

The dynamics conserve the Hamiltonian since

$$\begin{aligned} \frac{dH}{dt} &= \sum_{i=1}^{d} \left[\frac{d\theta_i}{dt}\frac{\partial H}{\partial \theta_i} + \frac{d\omega_i}{dt}\frac{\partial H}{\partial \omega_i} \right] \\ &= \sum_{i=1}^{d} \left[\frac{\partial H}{\partial \omega_i}\frac{\partial H}{\partial \theta_i} - \frac{\partial H}{\partial \theta_i}\frac{\partial H}{\partial \omega_i} \right] \end{aligned}$$

If H is conserved, then the acceptance probability based on Hamiltonian dynamics is 1. Simulating most Hamiltonian systems, we can only make H approximately conserved.

Reversibility

Hamiltonian dynamics is reversible, i.e. the mapping T_s from the state at time t, $(\theta(t), \omega(t))$, to the state at time $t+s$, $(\theta(t+s), p(t+s))$, is one-to-one, and hence as an inverse, T_{-s}. Under our usual assumptions for HMC, the inverse mapping can be obtained by negating ω, applying T_s, and then negating ω again. The reversibility of Hamiltonian dynamics is important for showing convergence of HMC.

Volume preservation

If we apply the mapping T_s to points in some region R in (θ, ω) space with volume V, the image of R under T_s will also have volume V. This feature simplifies calculation of the acceptance probability for Metropolis updates.

Euler's method

For simplicity, assume

$$H(\theta,\omega) = U(\theta) + K(\omega), \qquad K(\omega) = \sum_{i=1}^{d} \frac{\omega_i^2}{2m_i}.$$

One way to simulate Hamiltonian dynamics is to discretize time into increments of e, i.e.

$$\begin{array}{lll} \omega_i(t+e) & = \omega_i(t) + e\frac{d\omega_i}{dt}(t) & = \omega_i(t) - e\frac{\partial U}{\partial \theta_i}(\theta(t)) \\ \theta_i(t+e) & = \theta_i(t) + e\frac{d\theta_i}{dt}(t) & = \theta_i(t) + e\frac{\omega_i(t)}{m_i} \end{array}$$

Leapfrog method

An improved approach is the leapfrog method which has the following updates:

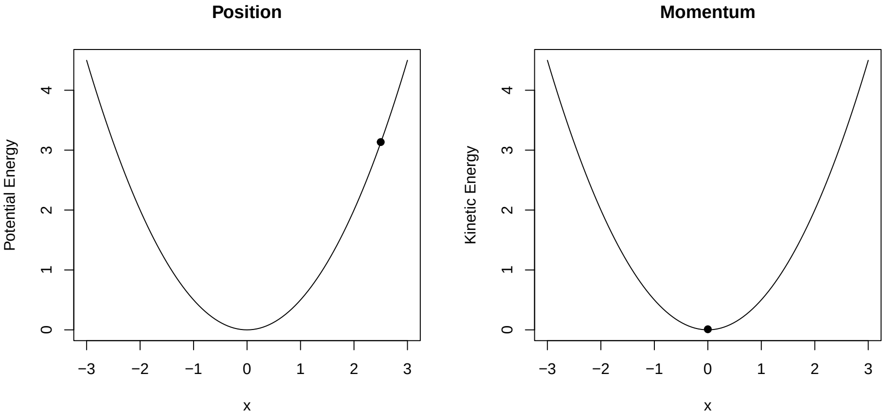
$$\begin{aligned} \omega_i(t+e/2) &= \omega_i(t) - (e/2)\frac{\partial U}{\partial \theta_i}(\theta(t)) \\ \theta_i(t+e) &= \theta_i(t) + e\frac{\omega_i(t+e/2)}{m_i} \\ \omega_i(t+e) &= \omega_i(t+e/2) - (e/2)\frac{\partial U}{\partial \theta_i}(\theta(t+e)) \end{aligned}$$

The leapfrog method is reversible and preserves volume exactly.

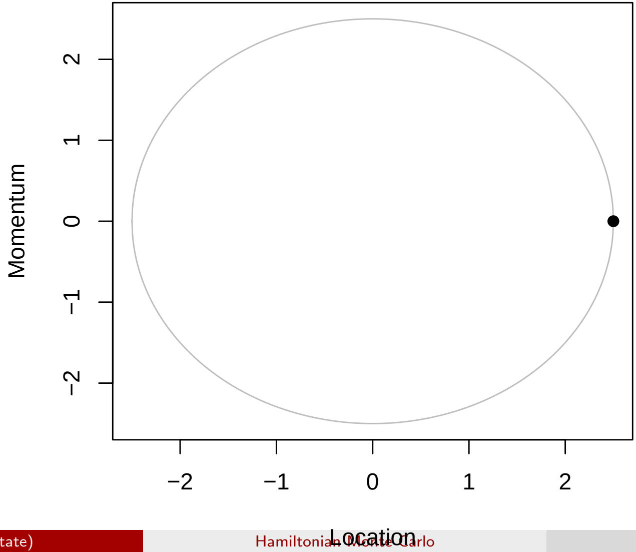
Leap-frog simulator

```
leap_frog = function(U, grad_U, e, L, theta, omega) {
  omega = omega - e/2 * grad_U(theta)

    for (l in 1:L) {
      theta = theta + e * omega
      if (l<L) omega = omega - e * grad_U(theta)
    }
    omega = omega - e/2 * grad_U(theta)
  return(list(theta=theta,omega=omega))
}
```

Leap-frog simulator



Conservation of the Hamiltonian



Probability distributions

The Hamiltonian is an energy function for the joint state of “position”, θ, and “momentum”, ω, and so defines a joint distribution for them, via

$$p(\theta, \omega) = \frac{1}{Z} \exp\left(-H(\theta, \omega)\right)$$

where Z is the normalizing constant.
If $H(\theta, \omega) = U(\theta) + K(\omega)$, the joint density is

$$p(\theta, \omega) = \frac{1}{Z} \exp\left(-U(\theta)\right) \exp\left(-K(\omega)\right).$$

If we are interested in a posterior distribution, we set

$$U(\theta) = -\log\left[p(y|\theta)p(\theta)\right].$$

Hamiltonian Monte Carlo algorithm

Set tuning parameters

- L: the number of steps
- e: stepsize
- $D = \{d_i\}$: covariance matrix for ω

Let $\theta^{(i)}$ be the current value of the parameter θ. The leap-frog Hamiltonian Monte Carlo algorithm is

1. Sample $\omega \sim N_d(0, D)$.
2. Simulate Hamiltonian dynamics on location $\theta^{(i)}$ and momentum ω via the leapfrog method (or any reversible method that preserves volume) for L steps with stepsize e. Call these updated values θ^* and $-\omega^*$.
3. Set $\theta^{(i+1)} = \theta^*$ with probability $\min\{1, \rho(\theta^{(i)}, \theta^*)\}$ where

$$\rho(\theta^{(i)}, \theta^*) = \frac{p(\theta^*|y)}{p(\theta^{(i)}|y)} \frac{p(\omega^*)}{p(\omega^{(i)})} = \frac{p(y|\theta^*)p(\theta^*)}{p(y|\theta^{(i)})p(\theta^{(i)})} \frac{N_d(\omega^*; 0, D)}{N_d(\omega^{(i)}; 0, D)}$$

   otherwise set $\theta^{(i+1)} = \theta^{(i)}$.

Reversibility

Reversibility for the leapfrog means that

- if you simulate from (θ, ω) to (θ^*, ω^*) for some step size e and number of steps L then
- if you simulate from $(\theta^*, -\omega^*)$ for the same e and L, you will end up at $(\theta, -\omega)$.

If we use q to denote our simulation “density”, then reversibility means

$$q(\theta^*, \omega^* | \theta, \omega) = q(\theta, -\omega | \theta^*, -\omega^*)$$

and thus in the Metropolis-Hastings calculation, the proposal is symmetric. In order to ensure reversibility of our proposal, we need to negate momentum after we complete the leap-frog simulation. So long as $q(\omega) = q(-\omega)$, which is true for a multivariate normal centered at 0, this will not affect our acceptance probability.

Conservation of Hamiltonian results in perfect acceptance

The Hamiltonian is conserved if $H(\theta, \omega) = H(\theta^*, \omega^*)$ which implies

$$\begin{aligned} p(\theta^*|y)p(\omega^*) &= \exp\left(-H(\theta^*, \omega^*)\right) \\ &= \exp\left(-H(\theta, \omega)\right) \\ &= p(\theta|y)p(\omega) \end{aligned}$$

and thus the Metropolis-Hastings acceptance probability is

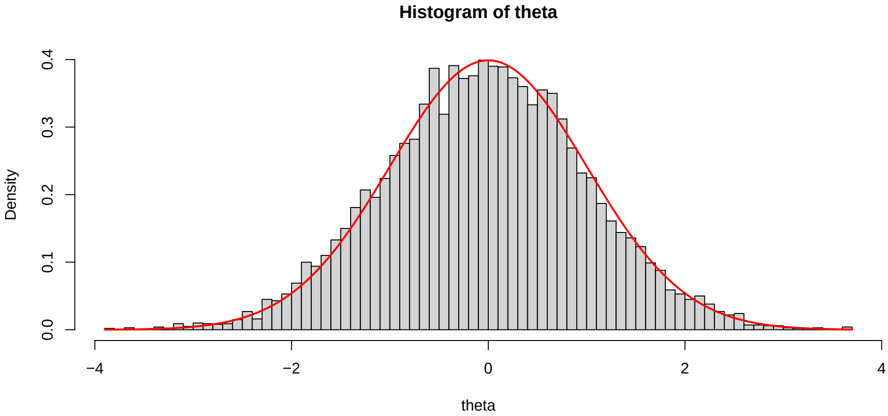
$$\rho(\theta^{(i)}, \theta^*) = \frac{p(\theta^*|y)p(\omega^*)}{p(\theta^{(i)}|y)p(\omega^{(i)})} = 1.$$

This will only be the case if the simulation is perfect! But we have discretization error. The acceptance probability accounts for this error.

```
HMC_neal = function(U, grad_U, e, L, current_theta) {
  theta = current_theta
  omega = rnorm(length(theta),0,1)
  current_omega = omega

  omega = omega - e * grad_U(theta) / 2

  for (i in 1:L) {
    theta = theta + e * omega
    if (i!=L) omega = omega - e * grad_U(theta)
  }
  omega = omega - e * grad_U(theta) / 2

  omega = -omega

  current_U  = U(current_theta)
  current_K  = sum(current_omega^2)/2
  proposed_U = U(theta)
  proposed_K = sum(omega^2)/2

  # cat(paste(e,L,i,theta,omega,"\n"))

  if (runif(1) < exp(current_U-proposed_U+current_K-proposed_K))
  {
    return(theta)
  }
  else {
    return(current_theta)
  }
}
```

```
HMC = function(n_reps, log_density, grad_log_density, tuning, initial) {
  theta = rep(0, n_reps)
  theta[1] = initial$theta

  for (i in 2:n_reps) theta[i] = HMC_neal(U = function(x) -log_density(x),
                                          grad_U = function(x) -grad_log_density(x),
                                          e = tuning$e,
                                          L = tuning$L,
                                          theta[i-1])
  theta
}
```

```
theta = HMC(1e4, function(x) -x^2/2, function(x) -x, list(e=1,L=1), list(theta=0))
hist(theta, freq=F, 100)
curve(dnorm, add=TRUE, col='red', lwd=2)
```



Tuning parameters

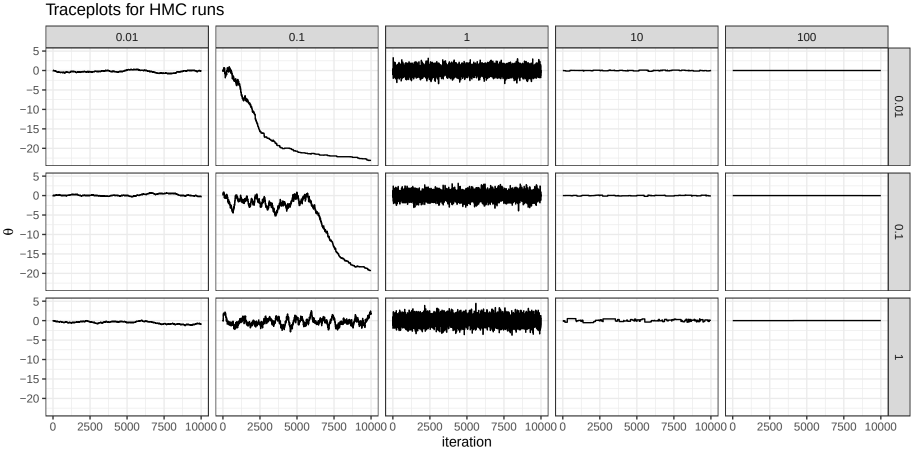
There are three tuning parameters:

- e: step size
- L: number of steps
- D: covariance matrix for momentum

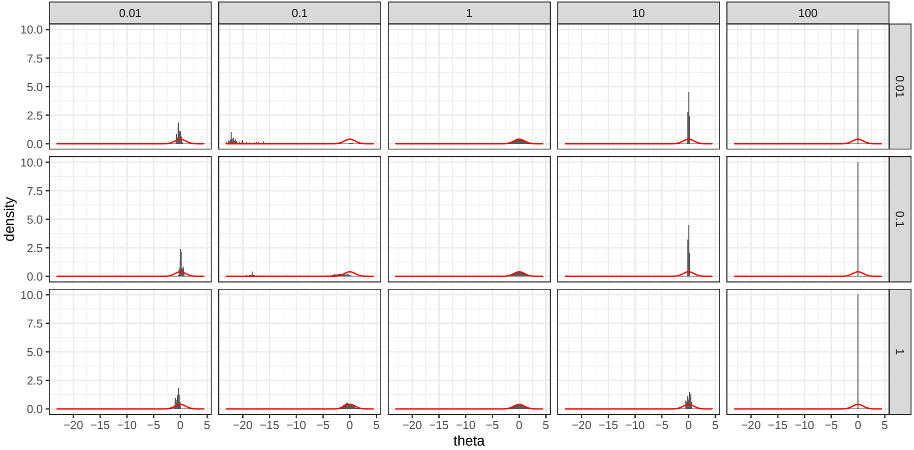
Let $\Sigma = V(\theta|y)$, then an optimal normal distribution for ω is $N(0, \Sigma^{-1})$. Typically, we do not know Σ, but we can estimate it using posterior samples. We can update this estimate throughout burn-in (or warm-up).

Effect of e and L

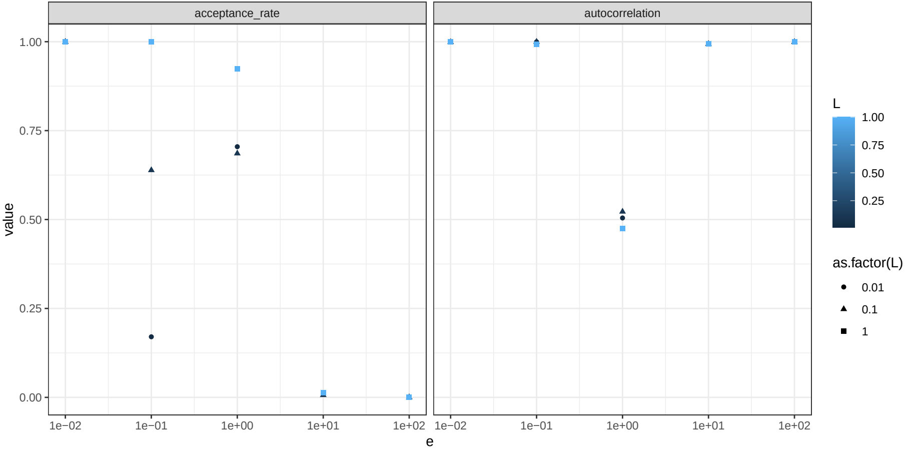
```
n_reps = 1e4
d = expand.grid(e=10^c(-2:2), L=10^c(-2:0))
r = ddply(d, .(e,L), function(xx) {
  data.frame(
    iteration = 1:n_reps,
    theta = HMC(n_reps, function(x) -x^2/2, function(x) -x, list(e=xx$e,L=xx$L), list(theta=0)))
})
```


Traceplots for HMC runs
0.01
0.1
1
10
100
0.01
0.1
1
5
0
−5
−10
−15
−20
θ
0
2500
5000
7500
10000
iteration

```
## Warning:  The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```



```
## Warning:  Removed 1 rows containing missing values (‘geom_point()‘).
```



Random-walk vs HMC

https://www.youtube.com/watch?v=Vv3f0QNWvWQ

Summary

Hamiltonian Monte Carlo (HMC) is a Metropolis-Hastings method using parameter augmentation and a sophisticated proposal distribution based on Hamiltonian dynamics such that

- the acceptance probability can be kept near 1
- while still efficiently exploring the posterior.

HMC still requires us to set tuning parameters

- e: step size
- L: number of steps
- D: covariance matrix for momentum

and can only be run in models with continuous parameters in $\mathbb{R}^d$ (or transformed to $\mathbb{R}^d$).