MCMC using Hamiltonian dynamics

by Radford Neal

Colin Cui

April 18, 2013

In this talk I will present...

In this talk I will present...

1. Background information of MCMC
2. Introduce an advanced method: Hamiltonian Monte Carlo (HMC)
3. Characteristics of HMC
   - Why it is faster than MCMC
   - How it avoids random walk
4. Graphs and a toy example I simulated
5. Other conditions
6. Further research and paper

I will try to explain things in its simplest way possible, hopefully!

Overview: From MCMC to Hamiltonian Monte Carlo

Overview: From MCMC to Hamiltonian Monte Carlo

- Problem Markov Chain Monte Carlo solves
  - "MCMC is a workhorse for the modern scientific computation" -Xiao-Li Meng
  - but there are problems in which MCMC can be slow and costly for high dimesional data
- Now, Hamiltonian Monte Carlo
  - Hamiltonian Monte Carlo originates from hybrid Monte Carlo in the field of statistical physics
  - It is a hybrid approach alternating between updating p and q, in the *Hamiltonian's equation* , $H(q, p)$, in which the energy is conserved.

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

q: position — p: momentum

$U(q)$: potential energy — $K(p)$: kinetic energy

  - Benefit: By taking the gradient of $H(q, p)$, it allows high-dimesional target distribution to coverge much more quickly and simple methods such as random walk Metropolis or Gibbs sampling.

Hamilton's equation

- Equation of motion

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i} \qquad \frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i}$$

for $i = 1, ..., d$. For any time interval s, these equations are define a mapping T_s, from state at time t to state at time $t + s$.

Let $z = (q, p)$, combine the Hamilton's equations

$$\frac{dz}{dt} = J\nabla H(z)$$

where ∇H is the gradient of H.

$$J = \begin{pmatrix} 0_{d\times d} & I_{d\times d} \\ I_{d\times d} & 0_{d\times d} \end{pmatrix}$$

J is a simplectic matrix, has the property: $M^T \Omega M = \Omega$

- Potential and kinetic energy

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

$K(p) = p^T M^{-1} p/2$ where M is a symmetric, positive-definite matrix, which is typically diagonal, and often scalar multiple of the identity matrix. This makes calculation much simpler!
K(p) is the minus log of the probability density of zero-mean Guassian distribution with covariance matrix M.

Discretizing Hamilton's equations: the leapfrog method

- What is a leapfrog in real life?

Discretizing Hamilton's equations: the leapfrog method

- What is a leapfrog in real life?

- Approximating Hamilton's first gradient in discrete time with step size ϵ. Since matrix M is diagonal we can sum up $m_1, ..., m_d$,

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

- Euler's method

$$p_i(t+\varepsilon) = p_i(t) - \varepsilon \frac{\partial U}{\partial q_i}(q(t))$$
$$q_i(t+\varepsilon) = q_i(t) + \varepsilon \frac{p_i(t)}{m_i}$$

- The leapfrog method

$$p_i(t+\varepsilon/2) = p_i(t) - (\varepsilon/2) \frac{\partial U}{\partial q_i}(q(t))$$
$$q_i(t+\varepsilon) = q_i(t) - (\varepsilon) \frac{p_i(t+\varepsilon/2)}{m_i}$$
$$p_i(t+\varepsilon) = p_i(t+\varepsilon/2) - (\varepsilon/2) \frac{\partial U}{\partial q_i}(q(t+\varepsilon))$$

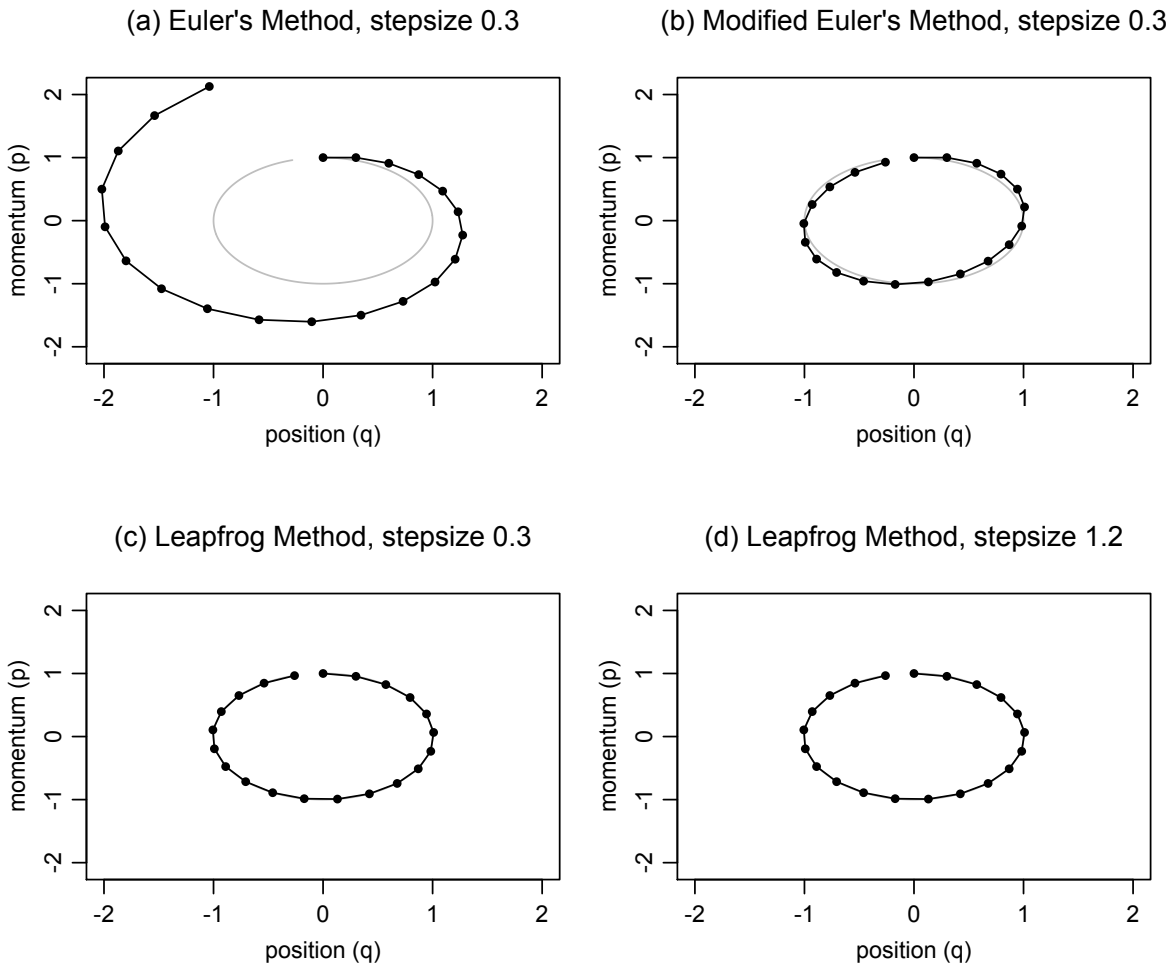
Results using discritized approximation

$H(q, p) = q^2/2 + p^2/2$ an example

Initial value: $q = 0$, $p = 1$

stepsize: $\varepsilon = 0.3$ for (a), (b), and (c)

stepsize: $\varepsilon = 0.3$ for (d)



Hamiltonian Monte Carlo Algorithm

- Canonical Distirbution

$$P(q,p) = \frac{1}{Z} exp(-H(q,p)/T)$$

If $H(q,p) = U(q) + (p)$, the joint density is

$$P(q,p) = \frac{1}{Z} exp(-U(q)/T) exp(-K(p)/T)$$

Posterior distribution with T=1: $U(q) = -log[\pi(q)L(q|D)]$.

- Hamiltonian Monte Carlo Algorithm
Goal: Draw random samples from the pdf proportional to $exp(-H)$

$$P(q,p) \propto exp[L(q) - K(p)]$$

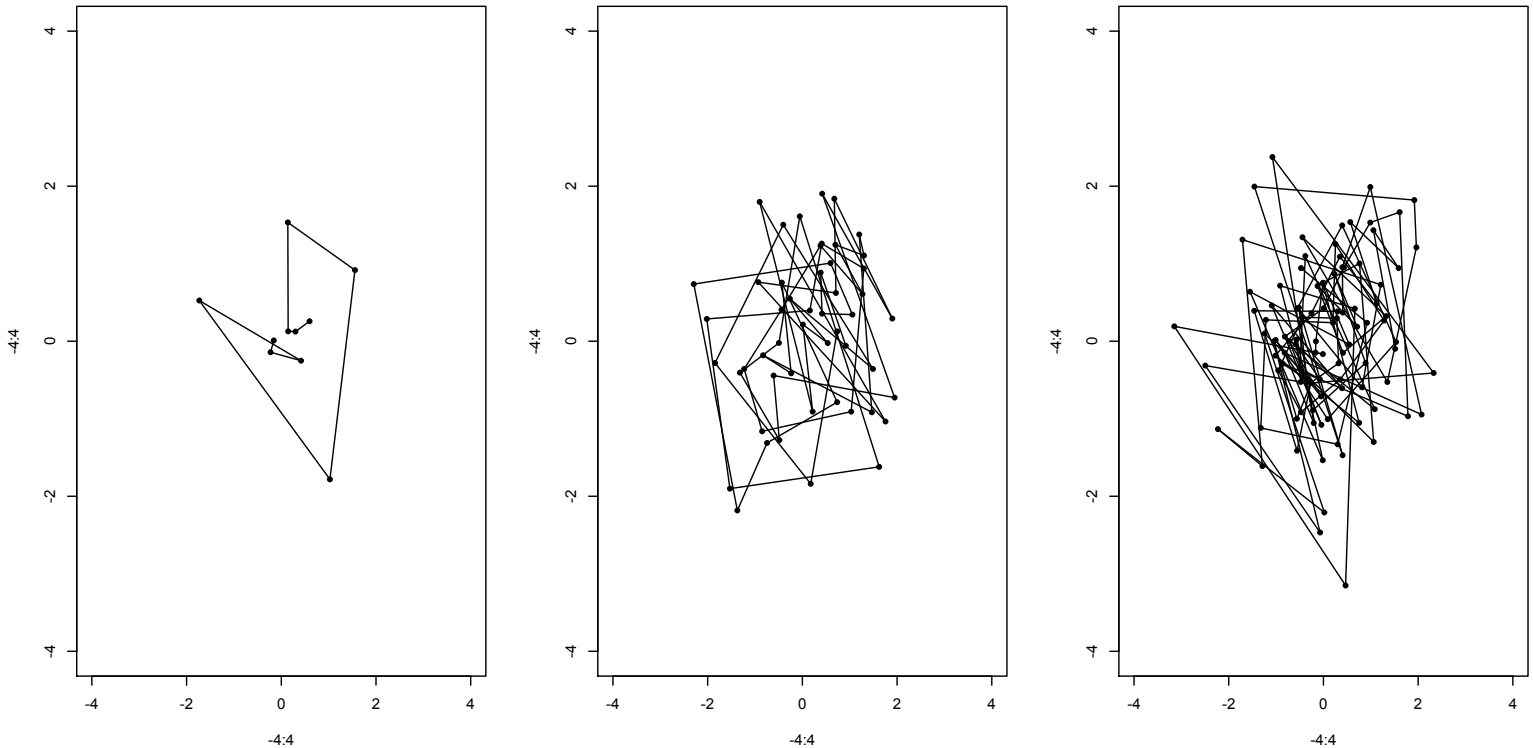
$L(q)$ is the log of the potential energy (target distribtution)
Step 1: Draw momentum variable, p_i from N(0,1)

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

Step 2:Given p_i, apply L leapfrog updates to position and momentum, using metropolis rejection ratio

$$\alpha = \text{min}[1, \text{exp}(\text{-H(q*,p*)+H(q,p)})]$$

Simulation Results



momentum=p_i from$N(0,1)$ **position=**$q_i(t+\varepsilon)$

Metropolis=$exp(-U(q*)+U(q)-K(p*)+K(p))$

Conservation of energy

$$\sum_{i=1}^{d}\left[\frac{\partial}{\partial q_i}\frac{dq_i}{dt}+\frac{\partial}{\partial p_i}\frac{dp_i}{dt}\right]=0$$

Properties of Hamiltonian dynamics

Properties of Hamiltonian dynamics

Conservation of the Hamiltonian

$$\frac{dH}{dt} = \sum_{i=1}^{d} \left[\frac{dq_i}{dt} \frac{\partial H}{\partial q_i} + \frac{dp_i}{dt} \frac{\partial H}{\partial p_i} \right] = \sum_{i=1}^{d} \left[\frac{dH}{dp_i} \frac{\partial H}{\partial q_i} - \frac{dH}{dq_i} \frac{\partial H}{\partial p_i} \right] = 0$$

Volume preservation

$$\sum_{i=1}^{d} \left[\frac{\partial}{\partial q_i} \frac{dq_i}{dt} + \frac{\partial}{\partial p_i} \frac{dp_i}{dt} \right] = \sum_{i=1}^{d} \left[\frac{\partial}{\partial q_i} \frac{dH}{dp_i} - \frac{dH}{dq_i} \frac{\partial H}{\partial p_i} \right] = \sum_{i=1}^{d} \left[\frac{\partial^2 H}{\partial q_i \partial p_i} - \frac{\partial^2 H}{\partial p_i \partial q_i} \right] = 0$$

Reversibility We can map T_s from the state at time t, $(q(t), p(t))$, the state at time $t+s$, $(q(t+s), p(t+s))$, is one-to-one, hence, has an inverse T_{-s}.

$$(q(t), p(t)) \rightleftarrows (q(t+s), p(t+s)),$$

t : *current time*, s : *time interval*

Simplecticness

$$B_s^T J^{-1} B_s = J^{-1}$$

This implies volume conservation, since $\det(B_s^T)\det(B_s)=\det(J^{-1})$ is one. When dimension is more than 1, the simplecticness condtion is stronger than volume perservation.

Effect of linear transformation

Effect of linear transformation

The following facts will help us improve performance with some knowledge of scales and correlations

- Kinetic energy used is: $K(p) = p^T M^{-1} p/2$
- Stability limit for ε is determined by width of the distributin in the most constrainted direction for a Guassian: this means it is the smallst eigenvalues of the covariance matrix for q.
- Stability for a general quadratic Hamiltonians with

$$K(p) = p^T M^{-1} p/2$$

$$K(p\prime) = p\prime^T M^{-1} p\prime/2$$

have solutions using linear transformation.
where $p\prime = Ap$ for some non-singular matrix A.

$$K'(p') = (A^T p')^T M^{-1}(A^T p')/2 = (p')^T (AM^{-1}A^T)p'/2 = (p')(M')^{-1}p'/2$$

where $M' = (AM^{-1}A^T)^{-1} = (A^{-1})^T MA^{-1}$

Tuning HMC

Tuning HMC

- What Step size?
  - Large step size: low acceptance for trajectory states
  - Small step size: wastes compuation time, can have slow exploration in random walk
- But!
  - Step size is almost independent of how many leapfrog steps are done, as shown earlier
  - Error usually does not increase with the number of leapfrog steps
- An toy example in 2-dimension

$$H(q,p) = q^2/2\sigma^2 + p^2/2$$

$$\begin{pmatrix} q(t+\varepsilon) \\ p(t+\varepsilon) \end{pmatrix} = \begin{pmatrix} 1-\varepsilon^2/2\sigma^2 & \varepsilon \\ -\varepsilon/\sigma^2 + \varepsilon^3/4\sigma^4 & 1-\varepsilon^2/2\sigma^2 \end{pmatrix} \begin{pmatrix} q(t) \\ p(t) \end{pmatrix}$$

- Convergence: depends on the eigevalues

$$(1-\varepsilon^2/2\sigma^2) \pm (\varepsilon/\sigma)\sqrt{\varepsilon^2/4\sigma^2 - 1}$$

When ε/σ <2, the eigenvalues are complex, and both have squared magnitude of

$$(1-\varepsilon^2/2\sigma^2)^2 + (\varepsilon/\sigma)(1-\varepsilon^2/4\sigma^2) = 1$$

The perfomance of HMC depends strongly on choosing good values for ε and L.

Optimal Acceptance Rate

Optimal Acceptance Rate

$$P(accept) = 2\Phi\left((0-\mu)/\sqrt{2\mu}\right) = 2\Phi(-\sqrt{\mu/2}) = a(\mu)$$

where $\mu = E[\Delta_d]$is proportional to ς^2, follows the proportionality

$$C_{rw} \propto 1/(a(\mu)\mu)$$

Minimized at μ=2.8, and a(μ)=0.23.
For HMC,

$$C_{HMC} \propto 1/(a(\mu)\mu^{1/4})$$

minimized at μ=0.41, and a(μ)=0.65.
65% is optimal acceptance rate.

Exploring the Distribution of Potential Energy

The scaling of HMC is strongly depended on the resampling of the momentum variables. Becaue $U(q)$ is a sum of d independent terms, its standard deviation will grow in proportion to $d^{1/2}$.

Efficiency of Hamiltonian Monte Carlo

Efficiency of Hamiltonian Monte Carlo

Gelman points out:

- HMC requires the gradient of log-posterior: "computing so for a complex model is at best tedious and at worst impossible.
- User needs to specify at least two parameters: a step size ε and a number of steps L for which to run a simulated Hamiltonian.
- Poor choice of either above will result a dramatic drop in HMC's efficiency.
- Methods for the adaptive MCMC lieterature can be used to tune ε on the fly, but setting L typically requires one or more costly turning runs. As well as the expertise to interpret the results of these tuning turns.
- This is a downside for the HMC, which is why some people choose not to use HMC.

MCMC estimators:

$$\text{Mean } \hat{v} = \frac{1}{N}\sum_{i=1}^{N} v_i$$

$$\text{variance } \sigma^2_{\hat{v}} = \frac{\text{var}(\hat{v})}{N}$$

Efficiency of MCMC:

$$\eta = \frac{\text{var}(v)}{N\sigma^2_{\hat{v}}}$$

Scaling With Dimensionality

We sample from any Metropolis-style algorithm as mentioned earlier from this relationship

$$P(x) = (1/Z)exp(-E(x))$$

$$1 = E(P(x*)/P(x)) = E(exp(-E(x*) - E(x))) = E(exp(-\Delta))$$

By Jensen's inequality

$$E(\Delta) \geq 0$$

For each state i,

$$U(q) = \Sigma u_i(q_i)$$

Let Δ_1 denote $E(x*) - E(x)$
Summing over the d states, we have

$$E(x) = U(q) + K(p)$$

As dimension gets larger, the energy difference $E(x)$ increases, acceptance probability decreases

$$min(1, exp(-\Delta_d))$$