Introduction to Hamiltonian Monte Carlo Method

Mingwei Tang

Department of Statistics
University of Washington

mingwt@uw.edu

November 14, 2017
Hamiltonian System

- Notation: $q \in \mathbb{R}^d$: position vector, $p \in \mathbb{R}^d$: momentum vector
- Hamiltonian $H(p, q): \mathbb{R}^{2d} \rightarrow \mathbb{R}^1$
- Evolution equation for Hamilton system

$$\begin{align*}
\frac{dq}{dt} &= \frac{\partial H}{\partial p} \\
\frac{dp}{dt} &= -\frac{\partial H}{\partial q}
\end{align*}$$  \quad (1)
Potential and Kinetic

- Decompose the Hamiltonian

\[ H(p, q) = U(q) + K(p). \]

- \( U(q) \): potential energy depend on position
- \( K(p) \): Kinetic energy depend on momentum

**Motivating example: Free fall**

\[ U(q) = mgq \]
\[ K(p) = \frac{1}{2} mv^2 = \frac{p^2}{2m} \]

\[ H(p, q) = mgq + \frac{p^2}{2m} \] is the total energy

- Velocity: \( v = \frac{dq}{dt} = \frac{\partial H}{\partial p} = \frac{p}{m} \)

- Force \( F = \frac{dp}{dt} = -\frac{\partial H}{\partial q} = -mg \)
1. Reversibility:
   - The mapping $T_s: (q(t), p(t)) \rightarrow (q(t + s), p(t + s))$ is one-to-one
   - Has inverse $T_{-s}$: negate $p$, apply $T_s$. negate $p$ again

2. Conserved (Hamiltonian invariant)
   \[
   \frac{dH}{dt} = \frac{dq}{dt} \frac{\partial H}{\partial q} + \frac{dp}{dt} \frac{\partial H}{\partial p} = \frac{\partial H}{\partial p} \frac{\partial H}{\partial q} - \frac{\partial H}{\partial q} \frac{\partial H}{\partial p} = 0
   \]
   $H(p, q)$ is constant over time $t$.

3. Volume preservation:
   - The map $T_s$ preserves the volume
   - For small $\delta$, Jacobian $\left| \det \left( \frac{\partial T_\delta}{\partial (p, q)} \right) \right| \approx 1$
Idea of HMC

- **D**: Observed data, **q**: parameters (latent variables), \( \pi(q) \) prior distribution
- Likelihood function \( L(D|q) \)
- Posterior distribution
  \[
  \Pr(q|D) \propto L(D|q)\pi(q)
  \]
- Position — parameters, potential \( U(q) \) — log-posterior
  \[
  U(q) = -\log [L(D|q)\pi(q)]
  \]
- Introduce ancillary variable \( p \) for Kinetic energy
  \[
  K(p) = \sum_{i=1}^{d} \frac{p_i^2}{2m_i} \propto \log (\mathcal{N}(0, \mathbf{M}))
  \]
  \( p, q \) are independent
- Hamiltonian: \( H(p, q) = U(q) + K(p) \)
Now we defined $U(q)$ and $K(p)$. Relate that to a distribution

- **Canonical distribution**

$$
Pr(p, q) = \frac{1}{Z} \exp \left( -\frac{H(p, q)}{T} \right) = \frac{1}{Z} \exp \left( -\frac{U(q)}{T} \right) \exp \left( -\frac{K(p)}{T} \right) \quad (2)
$$

where $T$: temperature, $Z$ normalizing constant

- Usually set $T = 1$, $Pr(q, p) \propto$ Posterior distribution $\times$ Multivaranit Guassian

- **Goal:** sample $(p, q)$ jointly from canonical distribution
Ideal HMC

- Specify variance matrix $\mathbf{M}$, time $s > 0$
- For $i = 1, \ldots, N$
  1. Sample $p^{(i)}$ from $\mathcal{N}(0, \mathbf{M})$
  2. Starting with current $(p^{(i)}, q^{(i-1)})$, integral on Hamiltonian system for $s$ period:

$$ (p^*, q^*) \leftarrow T_s((p^{(i)}, q^{(i-1)})) $$

(leaves $H(\cdot, \cdot)$ invariant)
  3. $q^{(i)} \leftarrow q^*$, $p^{(i)} \leftarrow -p^*$

- Output $q^{(1)}, \ldots, q^{(N)}$ as posterior samples
- Problem: The Hamiltonian system may not have a closed-form solution
  Need numerical method to for ODE system
Numerical ODE integrator

- Targeting problem:

\[
\begin{align*}
\frac{dq}{dt} &= \frac{\partial H}{\partial p} = M^{-1} p \\
\frac{dp}{dt} &= -\frac{\partial H}{\partial q} = \nabla \log (L(D|q)p(q))
\end{align*}
\]

- Leap-frog method, for small time $\epsilon > 0$

\[
\begin{align*}
p(t + \epsilon/2) &= p(t) - (\epsilon/2) \frac{\partial U}{\partial q}(q(t)) \\
q(t + \epsilon) &= q(t) + \epsilon M^{-1} p(t + \epsilon/2) \\
p(t + \epsilon) &= p(t + \epsilon/2) - (\epsilon/2) \frac{\partial U}{\partial q}(q(t + \epsilon/2))
\end{align*}
\]
Numerical stability for Hamiltonian system

(a) Euler’s Method, stepsize 0.3

(b) Modified Euler’s Method, stepsize 0.3

(c) Leapfrog Method, stepsize 0.3

(d) Leapfrog Method, stepsize 1.2
Property of Leap frog

- Time reversibility: Integrate $n$ steps forward and then $n$ steps backward, arrive at same starting position.
- Symplectic property: Converse the (slightly modified) energy
Specify variance $M$, time $s > 0$

For $i = 1, \ldots, N$

1. Sample $p^{(i)}$ from $\mathcal{N}(0, M)$

2. Starting with current $(p^{(i)}, q^{(i-1)})$, integral on Hamiltonian system for $s$ period:

$$ (p^*, q^*) \leftarrow T_s((p^{(i)}, q^{(i-1)})) $$

3. $q^{(i)} \leftarrow q^*$, $p^{(i)} \leftarrow -p^*$

Output $q^{(1)}, \ldots, q^{(N)}$ as posterior samples

Numerical method does not leave $H(p, q)$ unchanged during integration

$$ H((p^*, q^*)) \neq H((p^{(i)}, q^{(i-1)})) $$

Need to adjust that
HMC in practice

- Specify variance matrix $\mathbf{M}$, step size $\epsilon > 0$, $L$ : number of the leap frog steps

- For $i = 1, \ldots, N$
  1. Sample $p^{(i)}$ from $\mathcal{N}(0, \mathbf{M})$
  2. Starting with current $(p^{(i)}, q^{(i-1)})$,
     
     $$(p^*, q^*) \leftarrow \text{Leapfrog}(p^{(i)}, q^{(i-1)}, \epsilon, L)$$
     
     $p^* \leftarrow -p^*$
  3. Metropolis-Hastings with probability
     
     $$\alpha = \min \left\{ 1, \frac{\Pr(p^*, q^*)}{\Pr(p^{(i)}, q^{(i-1)})} \right\}$$
     
     set $q^{(i)} \leftarrow q^*$, $p^{(i)} \leftarrow p^*$
     (leaves canonical distribution invariant)

- Output $q^{(1)}, \ldots, q^{(N)}$ as posterior samples
Comparison with random walk Metropolis-Hastings

- HMC: proposal based on Hamiltonian dynamics, not random walk
- Random walk Metropolis-Hastings (RWMH) need more steps to get an independent sample
- Optimum acceptance: HMC (65%), RWMH (23%)
- Computation $d$:
  - Number of iterations to get a independent sample: $\mathcal{O}(d^{1/4})$ vs RWMH: $\mathcal{O}(d)$
  - Total number of computations $\mathcal{O}(d^{5/4})$ vs RWMH: $\mathcal{O}(d^2)$

See (Roberts et al. 2001) and (Neal 2011) for more details
Tuning parameters

- **Stepsizes $\epsilon$:**
  - Large $\epsilon$: Low acceptance rate
  - Small $\epsilon$: Waste computation, random walk behavior ($\epsilon L$) too small
  - Might need different $\epsilon$ for different region, e.g. choose $\epsilon$ by random

- **Number of leap-frog steps $L$:**
  - Trajectory length is crucial for exploring state space systematically
  - More constrained in some directions, but much less constrained in other directions
  - U-turns in long-trajectory
NUTS

- Solution: No-U-Turn Sampler (NUTS) (Hoffman et al. 2014)
  - Adaptive way to select number of leap-frog step $L$
  - Adaptive way to select step size $\epsilon$
- The exact algorithm behind Stan!
NUTS: Select $L$

- **Criterion for "U-turns"**

\[
\frac{d}{dt} \frac{||q_t - q_0||^2}{2} = (q_t - q_0)^T \cdot p_t < 0
\]  

- **Start from** $(p^{(i)}, q^{(i-1)})$

1. Run leap-frog steps until (3) happens. Have candidate set $B$ of $(p, q)$ pairs
2. Select subset $C \subseteq B$ satisfies detail balanced equation
3. Random select $q^{(i)}$ from $C$
Selecting stepsize $\epsilon$

- Warm-up phase $M_{\text{adapt}}$
- $H_t$ be the acceptance probability at $t$-th iterations e.g
  \[ H_t = \min \left\{ 1, \frac{\Pr(p^*, q^*)}{\Pr(p^{(t)}, q^{(t-1)})} \right\} \]
- $h_t(\epsilon) = \mathbb{E}_t[H_t|\epsilon]$
- one step Dual averaging in each iteration for solving
  \[ h_t(\epsilon) = \delta \]
  where $\delta$ is the optimum acceptance rate, for HMC $\delta = 0.65$
- Find $\epsilon$ after $M_{\text{adapt}}$ iterations
Summary

- **HMC**: A MCMC algorithm make use of Hamiltonian dynamics
  - Parameters as position, posterior likelihood as potential energy
  - Propose new state based on Hamiltonian dynamics
  - Leap-frog for numerical simulation, sensitive for tuning

- **NUTS**: A HMC with adaptive tuning on \((L, \epsilon)\) for more efficient proposal
  - \(L\): Avoid U-turns
  - \(\epsilon\): Dual-averaging optimization to make the acceptance rate close to optimum
Implement your Own HMC

- Review the Hamiltonian dynamics

\[
\begin{align*}
\frac{dq}{dt} &= \frac{\partial H}{\partial p} = M^{-1}p \\
\frac{dp}{dt} &= -\frac{\partial H}{\partial q} = \nabla \log (L(D|q)\pi(q))
\end{align*}
\]

- Need gradient

\[\nabla \log (L(D|q)\pi(q)) = \nabla \log (L(D|q)) + \nabla \log(\pi(q))\]

- Stan: automatic gradient calculation

- Gradient: Stan can do gradient-based optimization (quasi-Newton method L-BFGS)