Biostats 285: Advanced Bayesian Computing **Spring 2021**

Lecture 3: April 20

Lecturer: Andrew Holbrook *Scribes: Joanna Boland*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

3.1 Hamiltonian Dynamics

The document comprises of a rough outline of the chapter "MCMC Using Hamiltonian Dynamics" from the Handbook of Markov Chain Monte Carlo (Neal, 2011). We combine the MCMC algorithm and **Hamiltonian Dynamics**, which formalizes the motion of molecules following Newton's law of motion using differential equations. In order to better understand the dynamics, we use a real world example. Imagine a frictionless puck that slides over a surface of varying height in two dimensions. At any time point, the physical location and movement of the puck can be described by the **position** vector, q, and the **momentum** vector, p. The **potential energy**, $U(q)$, of the puck is proportional to the height of the surface at the current position of the puck, and its **kinetic energy**, $K(p)$, is equal to $|p|^2/(2m)$, where m is the **mass** of the puck. When the potential energy of the puck does not change, the puck moves at a constant **velocity**, equal to p/m. We model this physics problem using the **Hamiltonian**, $H(q,p)$, which is a function of q and p.

We use the partial derivatives of the Hamiltonian function to describe how the object's q and p change over time, t, such that:

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

for $i = 1, \ldots, d$. For any time interval of duration s, these equations define a mapping T_s, from the state at any time t to the state at time $t + s$. For HMC, we usually formula the Hamiltonian functions as:

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

where $U(q)$ is the negative log pdf of the distribution for q, which is our target. $K(p)$ is defined as

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

where m is a the "mass matrix" that is selected to be symmetric, p.d. (mass can't be 0), and is the covariance matrix of a mean zero Gaussian distribution associated with momentum.

3.1.1 Properties

- **Reversibility:** the mapping T_s from the state at the time t, $(q(t), p(t))$, to the state at time $t + s$, $(q(t+s), p(t+s))$, is one-to-one, and hence has an inverse, T_{-s}. In MCMC, this is important for showing that the updates leave the desired distribution invariant (leading to a reversible Markov chain).
- **Conservation of the Hamiltonian:** The dynamics of this system conserves the energy. For HMC, if H is conserved, the acceptance probability is equal to one. We show this as:

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

- **Volume Preservation:** We can also show that this system preserves volume in (q, p) space. If we map T_s to the points in some region R of (q, p) space, with volume V, the image of R under T_s will also have volume V. For MCMC, this allows to not be concerned about the volume in the acceptance probability changing as the algorithm updates. We show this as:

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

- **Symplecticness:** Volume preservation is also a result from symplecticness of the system. Let $z = (q, p)$, and define

$$J=\begin{bmatrix}0_{d\times d} & I_{d\times d}\\ -I_{d\times d} & 0_{d\times d}\end{bmatrix}.$$

  Symplecticness is when the Jacobian matrix, B_s, of the mapping T_s satisfies

$$B_s^\top J^{-1} B_s = J^{-1}.$$

  Therefore, $\det(B_s^\top)\det(J^{-1})\det(B_s) = \det(J^{-1}) \to \det(B_s)^2 = 1$. This indicates volume conservation.

3.1.2 Discretizing Hamilton's Equations - The Leapfrog Method

In order to solve for these differential equations numerically, we have to approximate these equations in discrete time, using increments of stepsize, ϵ. We start at time zero and iteratively compute the state at times ϵ, 2ϵ, $3\epsilon, \ldots$. Further, we will assume that M is diagonal, with elements $m_1, \ldots, m_d$, such that

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

A very famous and common method to approximate the solutions of a system differential equations is **Euler's method.** This method iteratively performs the following steps, indexed by $i = 1, \ldots, d$:

$$p_i(t+\epsilon)=p_i(t)+\epsilon\frac{dp_i}{dt}(t)=p_i(t)-\epsilon\frac{\partial U}{\partial q_i}(q(t)),$$

$$q_i(t+\epsilon)=q_i(t)+\epsilon\frac{dq_i}{dt}(t)=q_i(t)+\epsilon\frac{p_i(t)}{m_i},$$

and updates the position and momentum at each step. Much better results can be obtained by slightly modifying Euler's method as:

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

$$q_i(t+\epsilon)=q_i(t)+\epsilon\frac{p(t+\epsilon)}{m_i}.$$

We update the value of momentum, p, and then update the position, q, using our new value of momentum. Even better results can be obtained with the **leapfrop method**:

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

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

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

We update the momentum with a half-step, and then update the position using this new momentum. Then, using the new position vector, we update the momentum vector with another half-step.

3.2 MCMC from Hamiltonian Dynamics

We can simulate a Markov chain in which each iteration resamples the momentum and then performs a Metropolis update using the model of Hamiltonian dynamics. We wish to sample from the **canonical distribution** that is related to the potential energy. For some energy function, $E[x]$, for the state, x, of some physical system, the canonical distribution over states has pdf

$$P(x) = \frac{1}{Z} \exp\left(\frac{-E[x]}{T}\right).$$

Here, T is the temperature of the system, and Z is the normalizing constant. We then define the joint probability of the Hamiltonian dynamics for (q, p) as:

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

The conservation of energy under Hamiltonian dynamics means a trajectory will operate within a hypersurface of constant probability. If $H(q,p) = U(q) + K(p)$, the joint density is

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

Thus $q \perp p$ and have canonical distributions $U(q)$ and $K(p)$. Position, q, is our variable of interest, and we use p to allow us to assume the properties of Hamiltonian dynamics. The canonical distribution of the posterior (with $T = 1$) is defined by the potential energy function as:

$$U(q) = -\log\big[\pi(q)L(q|D)\big],$$

where $\pi(q)$ is the prior density, and $L(q|D)$ is the likelihood function given data, D.

3.2.1 The Two Steps of the HMC Algorithm

Each iteration of the HMC algorithm has two steps. The first step updates momentum p, and the second step can change both q and p. Both steps conserve energy, which means the combination of these steps also conserves energy.

Step 1: We update momentum from randomly drawing from the associated Gaussian distribution. For the kinetic energy, the $i = 1, \ldots, d$ momentum variables are independent, with $p_i \sim N(0, m_i)$. Note that q has not changed and is independent from p, meaning that drawing from the joint distribution conserves energy.

Step 2: A Metropolis update is performed, using Hamiltonian dynamics to propose a new state and update q, but also potentially update p. Using the current state, (q, p), the leapfrog method is implemented for L steps, with a stepsize of ϵ. L and ϵ need to be tuned for computational performance. We negate the momentum variables at the end of this L-step trajectory, giving a proposed state $(q^\star, p^\star)$. This proposed state is accepted and updates the current state with probability

$$\min\Big[1, \exp\big(-H(q^\star,p^\star) + H(q,p)\big)\Big] = \min\Big[1, \exp\big(-U(q^\star) + U(q) - K(p^\star) + K(p)\big)\Big].$$

If the proposed state is not accepted, we keep the previous current state. The pseudocode for the HMC algorithm can be defined as:

for $i = 1, \ldots, L$
  $p^\star \leftarrow N_q(0, 1)$
  $p^\star \leftarrow p^\star - (\epsilon/2)\nabla U(q)$
  for i in $1, \ldots, L$
    $q^\star \leftarrow q^\star + \epsilon p^\star$
    if $i \neq L$
      $p^\star \leftarrow p^\star - \epsilon \nabla U(q^\star)$
    endif
  endfor
  $p^\star \leftarrow p^\star - (\epsilon/2)\nabla U(q^\star)$
  $p^\star \leftarrow -p^\star$
  $U \leftarrow U(q)$
  $K \leftarrow \sum_{i=1}^{d} p_i^2/2$
  $U^\star \leftarrow U(q^\star)$
  $K^\star = \sum_{i=1}^{d} p_i^{2\star}/2$
  $a \leftarrow Uniform(0, 1)$
  if $a < \exp\left(U - U^\star + K - K^\star\right)$
    $q \leftarrow q^\star$
  else
    $q \leftarrow q$
  endif
endfor

References

Neal, R. M. 2011. *MCMC using Hamiltonian Dynamics*, Handbook of Markov Chain Monte Carlo, Pages 113–162.