STAT 535: Statistical Machine Learning **Autumn 2019**

Lecture 9: Hamiltonian Monte Carlo

Instructor: Yen-Chi Chen

The Hamiltonian Monte Carlo (HMC) is a new MCMC approach that has been shown to work better than the usual MH algorithm. It is based on the idea of Hamiltonian dynamics.

The high-level idea of HMC is to generate a proposal from a *better proposal distribution*' and modify the acceptance part so the it has a *higher acceptance rate*. In the usual MH algorithm, we are directly sample from a proposal density $q(y|x)$. The HMC modifies this process using two components: a random momentum (velocity) vector ω and the Hamiltonian dynamics. The momentum is required fro every coordinate of the position x. Thus, if $x \in \mathbb{R}^d$, then we also need a vector of d elements for the momentum. As the name suggest, the momentum vector determines how we move x during the dynamics. The randomness is due to the random momentum vector (and the later acceptance part).

The rough idea of one-run HMC is as follows. Starting at the location x_0:

1. (Proposal step 1) We draw a random momentum vector $\omega_0 \sim p(\omega) \propto e^{-V(\omega)}$, where $V(\omega)$ is called the kinematic energy. Often $p(\omega)$ is taken to be a multivariate Gaussian.
2. (Proposal step 2) Then we apply the Hamiltonian dynamics at location x_0 and velocity ω_0 with the Hamiltonian (energy) $H(x, \omega) = -\log \pi(x) + V(\omega)$ and let the dynamics run for time T. This changes (x_0, ω_0) to (x_T, ω_T). Note that the pair (x, ω) is called the state.
3. (Acceptance step) We accept the new location x_T with a probability of

$$a(x_0, \omega_0, x_T, \omega_T) = \max\left\{1, \frac{\exp(-H(x_T, \omega_T))}{\exp(-H(x_0, \omega_0))}\right\}.$$

To approximate the distribution of π, we will iterate the HMC several times.

Note that in the second step (Hamiltonian dynamics), the dynamics is deterministic. Namely, if we start with the same location and the same momentum, we always end up being in the same destination. So for HMC the proposal density $q(x_T|x_0)$ is determined by the density $p(\omega) \propto e^{-V(\omega)}$ and the initial location x_0

To understand what happen in the HMC, we first note that the Hamiltonian contains two parts.

Potential energy. The targeted density $\pi(x)$ is incorporated into the HMC through the Hamiltonian

$$H(x, \omega) = \underbrace{-\log \pi(x)}_{=U(x)} + V(\omega).$$

The quantity $U(x) = -\log \pi(x)$ is also known as the potential energy.

Kinematic energy. The momentum is drawn from the Kinematic energy. The density $p(\omega) \propto e^{-V(\omega)}$ is crucial in the performance of an HMC algorithm. In general, we will choose $V(\omega) = \sum_{j=1}^{d} \frac{\omega_j^2}{2m_j}$, where d is the dimension of x and m_j is called the mass of the j-th coordinate. Although this looks fancy, but it implied an extremely sample distribution of $p(\omega)$:

$$p(\omega) \propto e^{-\frac{1}{2}\omega^T M^{-1} \omega} \sim N(0, M),$$

where $M = \mathsf{diag}(m_1, \cdots, m_d)$. So in fact, we are generating the momentum from a multivariate Gaussian (and all coordinates are independent).

Hamiltonian dynamics. The Hamiltonian dynamics governs the usual motion of an object under a specified potential energy and the kinematic energy. It provides excellent description on many physical phenomena such as how planets orbiting around a star. When $H(x, \omega)$ is given, the Hamiltonian dynamics is a deterministic equation of motion. Suppose we start with a location $x(0)$ and a momentum $\omega(0)$, the trajectory of the state $\{x(t), \omega(t) : t \in [0, \infty)\}$ is determined by

$$\frac{dx_j(t)}{dt} \equiv x_j'(t) = \frac{\partial H(x(t), \omega(t))}{\partial \omega_j(t)}, \quad \frac{d\omega_j(t)}{dt} \equiv \omega_j'(t) = -\frac{\partial H(x(t), \omega(t))}{\partial x_j(t)}$$

for $j = 1, \cdots, d$. Note that even if we only have access to $r(x) \propto \pi(x)$, we can still compute the dynamics since the potential energy $U(x) = -\log \pi(x) = -\log r(x) + C_0$ for some constant C_0. We can totally ignore the constant C_0 in practice. Because kinematic energy is $V(\omega) = \sum_{j=1}^d \frac{\omega_j^2}{m_j}$, the change in location (in j-th coordinate) is simply

$$x_j'(t) = \frac{\omega_j}{m_j}.$$

Thus, given an initial state $(x(0), \omega(0)) = (x_0, \omega_0)$, after running the Hamiltonian dynamics for time T, we will move to the state $(x(T), \omega(T)) = (x_T, \omega_T)$. The new state (x_T, ω_T) is the new proposal. Then in the HMC, we will make a acceptance decision to see if we will accept this proposal.

Here is one caveat in the HMC:

No matter we accept or reject the proposal, we will draw a new momentum in the next iteration.

Namely, only the location x_T will be kept after this iteration. The momentum will be deleted and we will draw a new momentum (from the kinematic energy) without using any information from the previous iteration.

Potential informs momentum: better proposal. The Hamiltonian dynamics allow the target density $\pi(x)$ changes the momentum vector via the equation

$$\omega_j'(t) = -\frac{\partial H(x(t), \omega(t))}{\partial x_j(t)} = \omega_j'(t) = -\frac{\partial \log \pi(x(t))}{\partial x_j(t)}.$$

Thus, even if the original momentum $\omega(0)$ may be pointing toward a bad direction (with least density), the dynamics will adjust its orientation so that it tends to point toward a higher density area. So this proves the proposal location x_T.

Energy conservation of the Hamiltonian dynamics: high acceptance rate. The Hamiltonian dynamics has a powerful property called *energy conservation*, which implies that the *acceptance probability is very high*. It is not hard to see that the change of Hamiltonian energy with respect to time is

$$\begin{aligned}\frac{dH(x(t), \omega(t))}{dt} &= \sum_{j=1}^d \left\{ \frac{\partial H(x(t), \omega(t))}{\partial \omega_j(t)} \frac{d\omega_j(t)}{dt} + \frac{\partial H(x(t), \omega(t))}{\partial x_j(t)} \frac{dx_j(t)}{dt} \right\} \\ &= 0.\end{aligned}$$

Namely, the Hamiltonian energy will always stay the same during the dynamics. This is a powerful property! Now we examine the acceptance probability:

$$a(x_0, \omega_0, x_T, \omega_T) = \max\left\{1, \frac{\exp(-H(x_T, \omega_T))}{\exp(-H(x_0, \omega_0))}\right\}.$$

The acceptance probability uses the ratio between the initial Hamiltonian energy and the final Hamiltonian energy after applying the dynamics. Because the Hamiltonian energy is conserved during the dynamics, this ratio will always be 1! Namely, *the acceptance probability is* 1 if we apply a real Hamiltonian dynamics. In fact, we need this acceptance step because in practice, we are using a numerical approximation to the Hamiltonian dynamics so there could be some energy loss due to the approximation. So the final acceptance step is to account for this numerical error.

HMC and detailed balance. Here we have seen that the HMC tends to give a better proposal and have a high acceptance rate. But to make sure we are indeed sampling from the desired density, we need to show that the generated points converge to a stationary distribution that is the desired density π. First, it is easy to see that the generated points form a Markov chain since in each iteration, we only use the information from the previous location. So we only need to show that π satisfies the detailed balance equation of the transition under HMC. In the HMC (that we indeed perform the Hamiltonian dynamics), since the dynamics is deterministic, given the time T being fixed, the mapping

$$(x_0, \omega_0) \to (x_T, \omega_T)$$

is deterministic. Namely, there exists ϕ_1, ϕ_2 such that $x_T = \phi_1(x_0, \omega_0)$ and $\omega_T = \phi_2(x_0, \omega_0)$. An interesting fact about Hamiltonian dynamics is that if we reverse the time, the trajectory will remain the same. Namely, if we start the dynamics with initial location x_T and momentum $-\omega_T$, after time T we will come back to x_0 and ω_0. Namely, $x_0 = \phi_1(x_T, -\omega_T), \omega_0 = \phi_2(x_T, -\omega_T)$. Thus, there is a one-one correspondence between $(x_0, \omega_0) \leftrightarrow (x_T, -\omega_T)$.

To show the detail balanced, we need to show that

$$\pi(x)p(x \to y) = \pi(y)p(x \to x),$$

where $p(x \to y)$ is the transition density.

Here is an intuitive explanation about the detailed balanced. Suppose that there is only one ω such that $\phi_1(x, \omega) = y$ and let $\tilde{\omega} = \phi_2(x, \omega)$ be the corresponding velocity. Then we also have $\phi_1(y, -\tilde{\omega}) = x$ and $\phi_2(y, -\tilde{\omega}) = \omega$. Thus, there is also only one η such that the dynamics moves (y, η) to (x, ω) and the choice is $\eta = -\tilde{\omega}$.

In this case, $p(x \to y) = p(\omega)$ because ω is the only choice that moves x into y. Similarly, we have $p(y \to x) = p(-\tilde{\omega})$. Then

$$\begin{aligned}
\pi(x)p(x \to y) &= \pi(x)p(\omega) \\
&= \frac{1}{Z_0} \exp\{-U(x) - V(\omega)\} \\
&= \frac{1}{Z_0} \exp\{-H(x, \omega)\} \\
&= \frac{1}{Z_0} \exp\{-H(y, \tilde{\omega})\} \qquad \text{(Energy conservation)} \\
&= \frac{1}{Z_0} \exp\{-U(y) - V(\tilde{\omega})\} \\
&= \pi(y)p(\tilde{\omega}) \\
&= \pi(y)p(-\tilde{\omega}) \qquad (\omega \text{ is from a Gaussian}) \\
&= \pi(y)p(y \to x)
\end{aligned}$$

so the detailed balance is satisfied. Actually, this idea can be generalized to the case where we have more than one momentum leading to y; there is always a one-one correspondence between ω and $\tilde{\omega}$ and the detailed balance is always satisfied.

The HMC algorithm and the leapfrog method. The practical usage of the HMC involves a discretized step of the dynamics. This discretization is called the leapfrog method. Suppose that x_0 is the input location and we are only able to evaluate $r(x) \propto \pi(x)$. Also, let ϵ be the step size in the discretization and L is the number of updates in the dynamics. Namely, $\epsilon \cdot L = T$ is the time that we apply the dynamics.

1. Generate the initial momentum $\omega_0 \sim N(0, M^{-1})$.
2. Set $x^{(0)} = x_0$.
3. For the momentum, make a half-step update:
$$\omega^{(0)} = \omega_0 - \frac{\epsilon}{2}\nabla \log r(x^{(0)}).$$
4. For $\ell = 1, \cdots, L-1$, do the followings:
   (a) Update position: $x^{(\ell)} = x^{(\ell-1)} + \epsilon \cdot \omega^{(\ell-1)}$.
   (b) Update momentum: $\omega^{(\ell)} = \omega^{(\ell-)} - \epsilon \cdot \nabla \log r(x^{(\ell)})$.
5. Make one last update on the position: $x^{(L)} = x^{(L-1)} + \epsilon \cdot \omega^{(L-1)}$.
6. Make another half-step update of the momentum:
$$\omega^{(L)} = \omega^{(L-1)} - \frac{\epsilon}{2}\nabla \log r(x^{(L)}).$$
7. Compute the acceptance probability:
$$a(x_0, \omega_0, x^{(L)}, \omega^{(L)}) = \min\left\{1, \frac{\exp(-H(x^{(L)}, \omega^{(L)}))}{\exp(-H(x_0, \omega_0))}\right\}.$$
8. Accept $x_{\mathsf{new}} = x^{(L)}$ with a probability of $a(x_0, \omega_0, x^{(L)}, \omega^{(L)})$. If we reject, then $x_{\mathsf{new}} = x_0$.
9. Return x_{new}.

A practical challenge is how do we numerically approximate the dynamics part in the HMC algorithm. In the dynamics, the momentum and the location are updated simultaneously. But in practice, we have to make a choice on which one to update first. This leads to a problem that the algorithm is non-symmetric with respect to time. To see this, suppose that we start at x_1 with a momentum ω_1 and move to x_2 with a momentum ω_2. The actual Hamiltonian dynamics is time-reversible, meaning that if we apply the algorithm to $(x_2, -\omega_2)$, we will get back (x_1, ω_1). However, if we only use the leapfrog procedure (step 4), we will not move $(x_2, -\omega_2)$ back to (x_1, ω_1). So the `half step update` (step 3) before and after the `for` loop (step 6) is to resolve this problem and make the algorithm symmetric with respect to time.

Here is an `R` code for the HMC from https://arxiv.org/pdf/1206.1901.pdf, and excellent introduction to the HMC.

```
HMC = function (U, grad_U, epsilon, L, current_q)
{
  q = current_q
  p = rnorm(length(q),0,1)  # independent standard normal variates
  current_p = p
  # Make a half step for momentum at the beginning
  p = p - epsilon * grad_U(q) / 2
```

```
  # Alternate full steps for position and momentum
  for (i in 1:L)
  {
    # Make a full step for the position
    q = q + epsilon * p
    # Make a full step for the momentum, except at end of trajectory
    if (i!=L) p = p - epsilon * grad_U(q)
  }
  # Make a half step for momentum at the end.
  p = p - epsilon * grad_U(q) / 2
  # Negate momentum at end of trajectory to make the proposal symmetric
  p = -p
  # Evaluate potential and kinetic energies at start and end of trajectory
  current_U = U(current_q)
  current_K = sum(current_p^2) / 2
  proposed_U = U(q)
  proposed_K = sum(p^2) / 2
  # Accept or reject the state at end of trajectory, returning either
  # the position at the end of the trajectory or the initial position
  if (runif(1) < exp(current_U-proposed_U+current_K-proposed_K))
  {
    return (q)  # accept
  }
  else {
    return (current_q)  # reject
  }
}
```

Although the HMC works well theory, it requires tuning a couple of parameters for practical use. The choice of step size ϵ and the number of updates L and the mass M will all affect the final performance. Detailed discussion about the tuning can be found in Section 5.4 of https://arxiv.org/pdf/1206.1901.pdf.