Hamiltonian Monte Carlo and the NUTS Algorithm

MTH496 (UGP-1)

Prof. Dootika Vats

Zehaan Naik

Dept. of Mathematics and Statistics
Indian Institute of Technology Kanpur

Semester-I, 2024-25

Problem Statement

Hamiltonian Monte Carlo (HMC):
HMC uses Hamiltonian dynamics to generate distant proposals for the Metropolis algorithm.

The Main Problem:
HMC's performance depends on two parameters: step size ϵ and number of steps L. If L is too small, the algorithm behaves like a random walk; if L is too large, computation is wasted.

The NUTS Algorithm:
NUTS (as presented in Hoffman, Gelman, et al. 2014) is an extension to HMC that eliminates the need to set a number of steps L.

Metropolis Hastings Algorithm

The MH Algorithm

1. Draw $y \sim Q(x, .)$
2. Independently, draw $u \sim U(0, 1)$
3. Set $X_{n+1} = y$ if $u \leq \min(1, \alpha(x, y))$
4. Else, set $X_{n+1} = X_n$
5. Keep repeating steps 1-4 until n reaches the desired length.

Hasting's Ratio

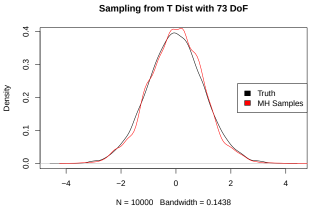
$$\alpha(x, y) = \frac{\pi(y)q(y, x)}{\pi(x)q(x, y)}$$

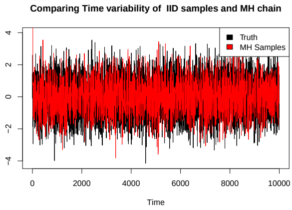
Metropolis Hastings Algorithm

Examples

Consider a target density $\pi \sim T_{73}$ and a proposal mechanism comprised of a Kernel $Q(x, y)$ defined as $y \mid x \sim N(x, 1)$. i.e.

$$q(x, y) = \frac{1}{\sqrt{2\pi}} e^{\frac{(y-x)^2}{2}}$$





The Metropolis-Hastings Algorithm

Important Results:

Theorem - 1

The MH algorithm defines the following transition kernel:

$$P(x, A) = \int_A Q(x, dy)a(x, y) + \delta_x(A) \int [1 - a(x, u)]Q(x, du).$$

Theorem - 2

The MH kernel is π-symmetric and hence π-invariant.

Hamiltonian Dynamics

Introduction:

Assume an imaginary particle a moving on a frictionless surface. The position-momentum vector (p, q) give us:

$$U(q) = -\log(\pi(q)) \qquad \text{- Potential Energy}$$

$$K(p) = \frac{p^2}{2m} \qquad \text{- Kinetic Energy}$$

Together, these two quantities define the state of the particle.

Hamiltonian Dynamics

Hamiltonian Equations

The Hamiltonian is:

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

And the Hamiltonian equations are:

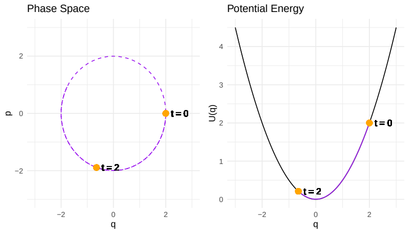
$$\frac{dq}{dt} = \frac{\partial H(p,q)}{\partial p}$$
$$\frac{dp}{dt} = -\frac{\partial H(p,q)}{\partial q}.$$

Hamiltonian Monte Carlo

Examples

Consider sampling from the target $\pi \sim N(0,1)$ and momentum p is distributed as $N(0,1)$. From the equations above:

$$U(q) = \frac{q^2}{2} \qquad K(p) = \frac{p^2}{2}$$



Hamiltonian Dynamics

Assume, for a particle with state (p, q), we get the explicit time dependence, $(p(t), q(t))$, by solving the Hamiltonian equations.
Define, $T_s(p_t, q_t) := (p_{t+s}, q_{t+s})$
Then, $T_s(T_{-s}(p_t, q_t)) = (p_t, q_t)$

An Involution

We draw motivation from the result above to define an involution,

$$P_s : (p_t, q_t, \epsilon = 1) \to (T_{\epsilon s}(p_t, q_t), \epsilon = -\epsilon).$$

Using Hamiltonian dynamics, we know,

$$|P_s| = 1.$$

Hamiltonian Monte Carlo

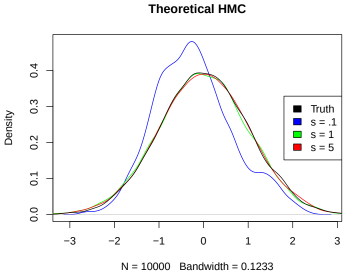
The HMC Algorithm

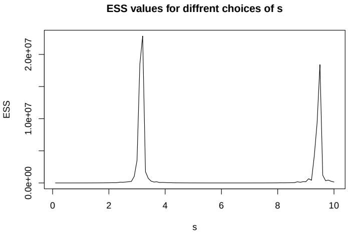
1. Sample $p \sim N(0,1)$ and independently $u \sim U(0,1)$
2. Set
$$(p^*, q^*) = P_s(p, q)$$
3. Set
$$\alpha(p,q) = \frac{\pi(q^*) f_P(p^*)}{\pi(q) f_P(p)}$$
4. Set $q_{n+1} = q*$ if $u \leq \alpha(p,q)$
5. else, Set $q_{n+1} = q$

Note: f_p is the density function for the momentum p.

Standard Normal Example

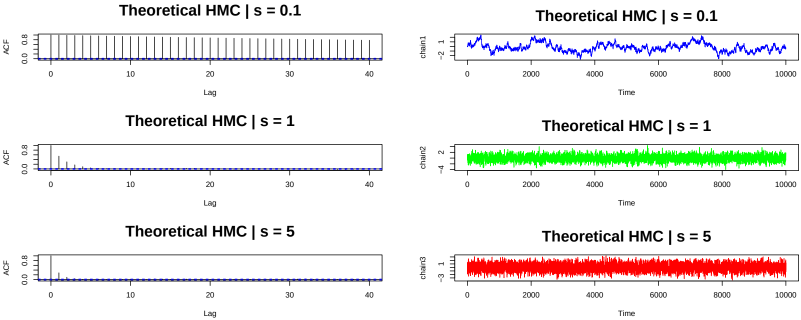
The following plots describe the sample quality and the cyclic nature of the HMC algorithm.





Standard Normal Example

The following plots describe the sample auto-correlation and the time variability of the HMC algorithm.



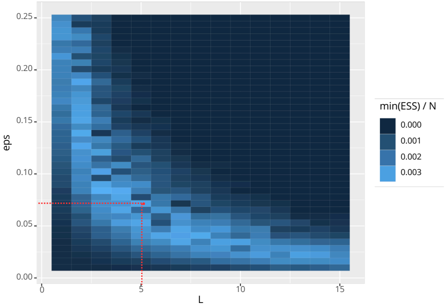
Leap Frog Integrator

Solving the Hamiltonian equations for most choices of π to obtain q_t and p_t explicitly is often not possible. Hence, we use the Leap Frog Integrator.

Note: $s = L\epsilon$

1. $$p(t+\epsilon/2) = p(t) - \frac{\epsilon}{2}\frac{\partial U(q(t))}{\partial q}$$
2. $$q(t+\epsilon) = q(t) + \epsilon\frac{p(t+\epsilon/2)}{m}$$
3. $\vdots$
4. $$q(t+L\epsilon) = q(t+(L-1)\epsilon) + \epsilon\frac{p(t+(2L-1)\epsilon/2)}{m}$$
5. $$p(t+L\epsilon) = p(t+(2L-1)\epsilon/2) - \frac{\epsilon}{2}\frac{\partial U(q(t+L\epsilon))}{\partial q}.$$

Optimum Choice of L and ϵ for a Multivariate Targets

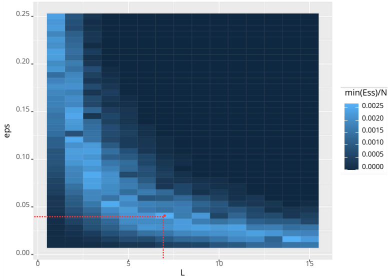


Density

Density: Multivariate Gaussian

$$\pi_G(x) = \prod_{i=1}^{d} \frac{1}{\sigma_i \sqrt{2\pi}} e^{\left(-\frac{x_i^2}{2\sigma_i^2}\right)}$$

Optimum Choice of L and ϵ for a Multivariate Targets

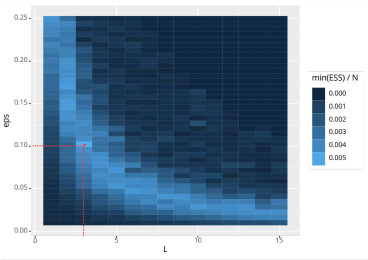


Density

Density: Logistic

$$\pi_L(x) = \prod_{i=1}^{d} \frac{1}{\sigma_i} \frac{e^{(x_i/\sigma_i)}}{\left\{1 + e^{(x_i/\sigma_i)}\right\}^2}$$

Optimum Choice of L and ϵ for a Multivariate Targets

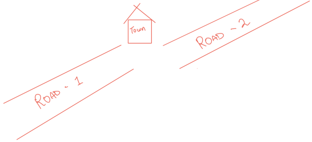


Density

Density: Skewed Gaussian

$$\pi_{SG}(x) = \prod_{i=1}^{d} \frac{2}{\sigma_i\sqrt{2\pi}} e^{\left(-\frac{x_i^2}{2\sigma_i^2}\right)} \Phi\left(\frac{\alpha x_i}{\sigma_i}\right)$$

NUTS: Problem Motivation



Problem Statement

- You are in a city with two roads extending in opposite directions.
- Your friend has travelled down one of these roads and got a flat tyre.
- He calls you for help but doesn't know which road he took.
- He also doesn't know how far he is from the city.
- Your task is to locate your friend and guide him back to the city.

The Naive NUTS Algorithm

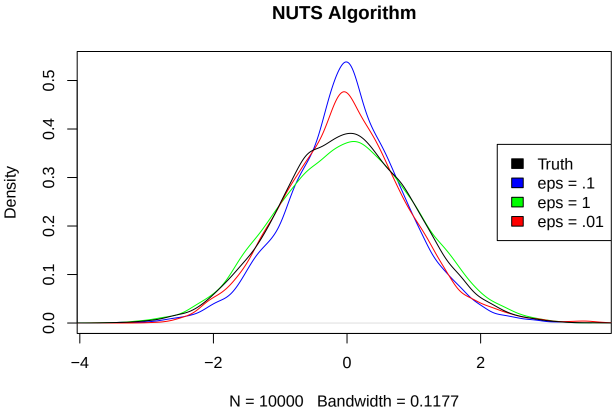
Given $\theta^0, \epsilon, \mathcal{L}, M$:
for $m = 1$ to M **do**
  Resample $r^0 \sim \mathcal{N}(0, I)$.
  Resample $u \sim$ Uniform $\left(\left[0, \exp\left\{\mathcal{L}\left(\theta^{m-1}\right) - \frac{1}{2} r^0 \cdot r^0\right\}\right]\right)$
  Initialize $\theta^- = \theta^{m-1}, \theta^+ = \theta^{m-1}, r^- = r^0, r^+ = r^0, j = 0, \mathcal{C} = \left\{\left(\theta^{m-1}, r^0\right)\right\}, s = 1$
  while $s = 1$ **do**
    Choose a direction $v_j \sim$ Uniform$(\{-1, 1\})$
    if $v_j = -1$ **then**
      $\theta^-, r^-, -, -, \mathcal{C}', s' \leftarrow$ BuildTree $\left(\theta^-, r^-, u, v_j, j, \epsilon\right)$
    else
      $-, -, \theta^+, r^+, \mathcal{C}', s' \leftarrow$ BuildTree $\left(\theta^+, r^+, u, v_j, j, \epsilon\right)$
    end if
    if $s' = 1$ **then**
      $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}'$
    end if
    $s \leftarrow s' \mathbb{I}\left[\left(\theta^+ - \theta^-\right) \cdot r^- \geq 0\right] \mathbb{I}\left[\left(\theta^+ - \theta^-\right) \cdot r^+ \geq 0\right]$.
    $j \leftarrow j + 1$
  end while
  Sample θ^m, r uniformly at random from $\mathcal{C}$
end for

Build Tree Function

function BUILDTREE($\theta, r, u, v, j, \epsilon$)
  if $j = 0$ **then**
    Base case: take one leapfrog step in the direction v
    $\theta', r' \leftarrow$ Leapfrog $(\theta, r, v\epsilon)$
    $\mathcal{C}' \leftarrow \begin{cases} \{(\theta', r')\} & \text{if } u \leq \exp\left\{\mathcal{L}(\theta') - \frac{1}{2} r' \cdot r'\right\} \\ \emptyset & \text{else} \end{cases}$
    $s' \leftarrow \mathbb{I}\left[\mathcal{L}(\theta') - \frac{1}{2} r' \cdot r' > \log u - \Delta_{\max}\right]$
    return $\theta', r', \theta', r', \mathcal{C}', s'$
  else
    Recursion: build the left and right subtrees
    $\theta^-, r^-, \theta^+, r^+, \mathcal{C}', s' \leftarrow$ BuildTree $(\theta, r, u, v, j-1, \epsilon)$
    if $v = -1$ **then**
      $\theta^-, r^-, -, -, \mathcal{C}'', s'' \leftarrow$ BuildTree $\left(\theta^-, r^-, u, v, j-1, \epsilon\right)$
    else
      $-, -, \theta^+, r^+, \mathcal{C}'', s'' \leftarrow$ BuildTree $\left(\theta^+, r^+, u, v, j-1, \epsilon\right)$
    end if
    $s' \leftarrow s' s'' \mathbb{I}\left[\left(\theta^+ - \theta^-\right) \cdot r^- \geq 0\right] \mathbb{I}\left[\left(\theta^+ - \theta^-\right) \cdot r^+ \geq 0\right]$
    $\mathcal{C}' \leftarrow \mathcal{C}' \cup \mathcal{C}''$
    return $\theta^-, r^-, \theta^+, r^+, \mathcal{C}', s'$
  end if
end function

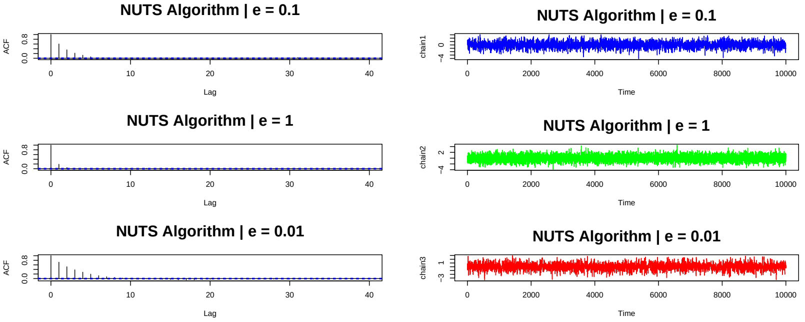
NUTS Example

NUTS Algorithm for a $N(0,1)$ target



NUTS Example

Comparing different choices of ϵ



Comparing NUTS and HMC

	ϵ	L_{optim}	avg(j)
1	0.0223	13	1.3813
2	0.0284	9	1.2615
3	0.0346	7	1.1897
4	0.0407	8	1.143
5	0.0469	7	1.1104
6	0.0530	5	1.0793
7	0.0592	7	1.0596
8	0.0653	4	1.0431
9	0.0715	5	1.033
10	0.0771	4	1.0244
11	0.0832	3	1.015

Table: Target: Multivariate Gaussian

	ϵ	L_{optim}	avg(j)
1	0.0223	15	1.3775
2	0.0284	10	1.2645
3	0.0346	9	1.1976
4	0.0407	8	1.1473
5	0.0469	6	1.107
6	0.0530	5	1.0791
7	0.0592	6	1.0601
8	0.0653	4	1.0423
9	0.0715	5	1.0338
10	0.0776	4	1.0239
11	0.0838	4	1.0169

Table: Target: Logistic

Future Work

I aim to explore two areas within this framework:

- **Dual Averaging:**
  An extension to the NUTS Algorithm that automatically tunes ϵ. This gives us a complete problem-agnostic HMC sampler.
- **Maximising L:**
  To fix the value of L based on device efficiency and maximum computation time allowed. Computing the best values for s and ϵ to both generate distant proposals and mitigate random walk behaviour.

References I

Hoffman, Matthew D, Andrew Gelman, et al. (2014). "The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo.". In: *J. Mach. Learn. Res.* 15.1, pp. 1593–1623.

Holbrook, Andrew (2021). "Dr. Andrew Holbrook's lecture on Hamiltonian Monte Carlo (HMC).". In: *Dr. Andrew Holbrook's lecture on Hamiltonian Monte Carlo (HMC).*

Neal, Radford M (2012). *Handbook of Markov Chain Monte Carlo, Chapter 5.* arXiv preprint arXiv:1206.1901.

Thank you!