ECON5120 Bayesian Data Analysis

Week 8: Hamiltonian Monte Carlo (Advanced MCMC Methods)

Santiago Montoya-Blandón

University of Glasgow
Adam Smith Business School

March 5, 2025

Lecture outline

1. Motivating Hamiltonian Monte Carlo
2. Digression: Hamiltonian Dynamics
3. Hamiltonian Monte Carlo Algorithm
4. HMC in Practice

Metropolis-Hastings in High Dimensions

- Metropolis-Hastings (MH) algorithm is powerful and general
- If likelihood can be evaluated easily, we usually default to MH
- However, MH is not without issues, particularly in large dimensions:
  - Proposal tuning is difficult
  - Acceptance rate cannot be too large to correctly explore parameter space
  - Introduces substantial correlation in draws
  - Practical convergence is difficult to achieve

UofG

Metropolis-Hastings in High Dimensions

- MH explores parameter space randomly from a given seed
- Given current state and a carefully constructed transition probability, accept candidate draw with some probability
- Candidates tend to be close to current value, creating autocorrelation
- Can we move in a smarter way through the parameter space?
- Idea: introduce *deterministic* (i.e., non-probabilistic) moves

Example: Deterministic Moves for Sampling

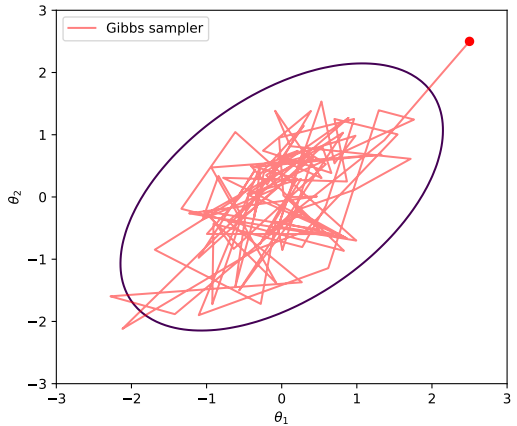

Figure: Comparison of Gibbs and Deterministic steps

Example: Deterministic Moves for Sampling

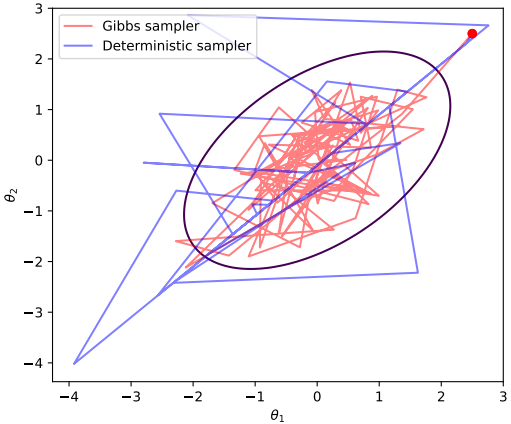

Figure: Comparison of Gibbs and Deterministic steps

Metropolis-Hastings-Green Algorithm

- Metropolis-Hastings-Green algorithm is an extension of MH
- Introduce an auxiliary variable ψ such that parameters are now (θ, ψ)
- Given current state $\theta^{(s)}$: draw $\psi^{(s)}$ from proposal $g(\psi \mid \theta^{(s)})$
- Use deterministic function T to obtain candidates: $(\theta^*, \psi^*) = T(\theta^{(s)}, \psi^{(s)})$

Metropolis-Hastings-Green Algorithm

- Metropolis-Hastings-Green algorithm is an extension of MH
- Introduce an auxiliary variable ψ such that parameters are now (θ, ψ)
- Given current state $\theta^{(s)}$: draw $\psi^{(s)}$ from proposal $g(\psi \mid \theta^{(s)})$
- Use deterministic function T to obtain candidates: $(\theta^*, \psi^*) = T(\theta^{(s)}, \psi^{(s)})$
- Calculate acceptance rate as usual, taking into account Jacobian of T:

$$\alpha(\theta^*, \psi^*; \theta^{(s)}, \psi^{(s)}) = \frac{p(\theta^* \mid y) g(\psi^* \mid \theta^*)}{p(\theta^{(s)} \mid y) g(\psi^{(s)} \mid \theta^{(s)})} \cdot |\nabla T(\theta^{(s)}, \psi^{(s)})|$$

- Take $u \sim U(0, 1)$, and accept or reject according to

$$\theta^{(s+1)} = \begin{cases} \theta^* & \text{if } u \leq \alpha(\theta^*, \psi^*; \theta^{(s)}, \psi^{(s)}) \\ \theta^{(s)} & \text{otherwise} \end{cases}$$

Lecture outline

1. Motivating Hamiltonian Monte Carlo
2. Digression: Hamiltonian Dynamics
3. Hamiltonian Monte Carlo Algorithm
4. HMC in Practice

Hamiltonian Monte Carlo

- Hamiltonian Monte Carlo (HMC) is a special case of the Metropolis-Hastings-Green algorithm
- It chooses a particular shape of determinist function T with good properties
- Transition function T is motivated by energy conservation in particle physics
- We require a short digression into the language used by physicists
- I will focus on intuition, not deriving the equations myself

Hamiltonian Dynamics

- $q \in \mathbb{R}^p$ is the position of an object, with initial position q_0
- $p \in \mathbb{R}^p$ is the momentum of an object, with initial momentum p_0
- $U(q) = -\log \pi(q)$ is the system's potential energy
- $K(p)$ is the kinetic energy
- $H(q, p) = U(q) + K(p)$ is the total energy
- System governed by Hamiltonian equations (conservation of energy):

$$(q_t, p_t) \implies \begin{aligned} \frac{dq}{dt} &= \frac{\partial H}{\partial p} \\ \frac{dp}{dt} &= \nabla \log \pi(q) \end{aligned} \implies (q_{t+1}, p_{t+1}) = T\,(q_t, p_t)$$

- There is a *unique deterministic* function T that satisfies these equations

HMC: Standard normal example

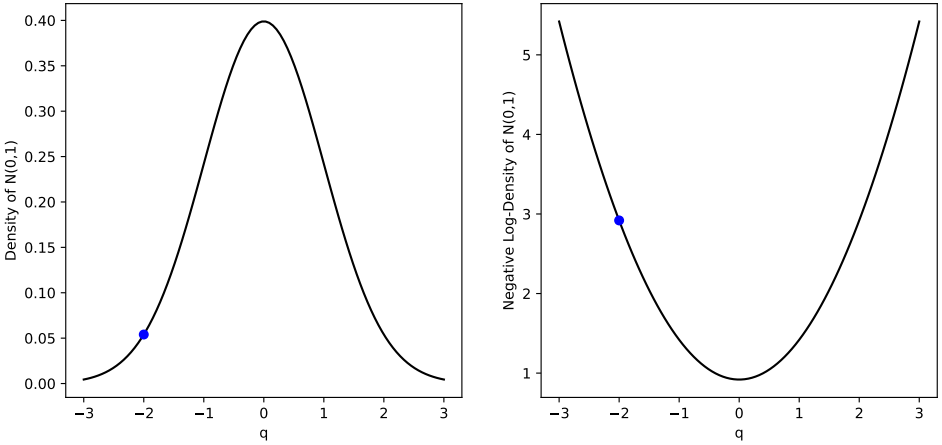

Figure: Standard normal density and its potential energy

Hamiltonian Dynamics

- If we evolve the system according to T, it reaches equilibrium
- Deterministic transition mapping T satisfies:
  1. Reversible: same forwards as backwards in time
  2. Time-invariant: we can write T without the t subscript
  3. Volume preserving: Jacobian equals 1, i.e., $|\nabla T(q, p)| = 1$
- Second idea: evolve system for short time, then re-sample momentum and evolve system again

HMC: Standard normal example

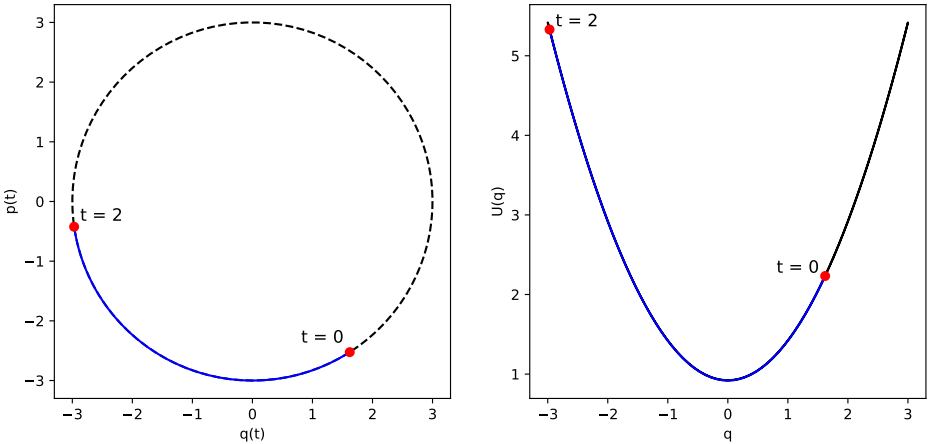

Figure: Hamiltonian evolution with standard normal density

HMC: Standard normal example

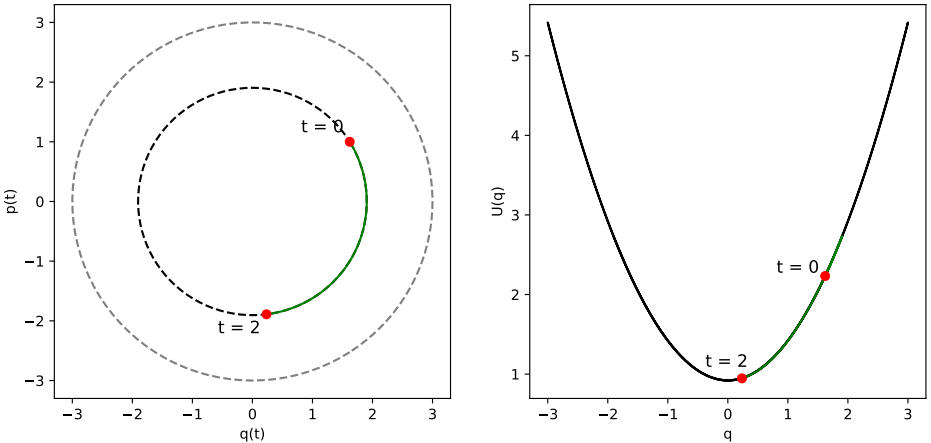

Figure: Hamiltonian evolution with standard normal density

Lecture outline

1. Motivating Hamiltonian Monte Carlo
2. Digression: Hamiltonian Dynamics
3. Hamiltonian Monte Carlo Algorithm
4. HMC in Practice

HMC for Posterior Sampling

- Translating from the physics language to our task
- θ is the equivalent of q, ψ is equivalent of p
- Choose a distribution for ψ, usually $\psi \sim \mathcal{N}_p(0, M)$ with mass matrix M
- Use Hamiltonian transition mapping T as deterministic function
- Use Metropolis-Hastings-Green in this setting

HMC Algorithm

1. Start at a current state $\theta^{(s)}$
2. Draw $\psi^{(s)}$ from proposal $\mathcal{N}_p(0, M)$
3. Use deterministic function T to obtain candidates: $(\theta^*, \psi^*) = T(\theta^{(s)}, \psi^{(s)})$
4. Calculate acceptance rate:

$$\alpha(\theta^*, \psi^*; \theta^{(s)}, \psi^{(s)}) = \frac{p(\theta^* \mid y) g(\psi^* \mid \theta^*)}{p(\theta^{(s)} \mid y) g(\psi^{(s)} \mid \theta^{(s)})}$$

5. Take $u \sim U(0, 1)$, and accept or reject according to

$$\theta^{(s+1)} = \begin{cases} \theta^* & \text{if } u \leq \alpha(\theta^*, \psi^*; \theta^{(s)}, \psi^{(s)}) \\ \theta^{(s)} & \text{otherwise} \end{cases}$$

HMC: Standard normal example

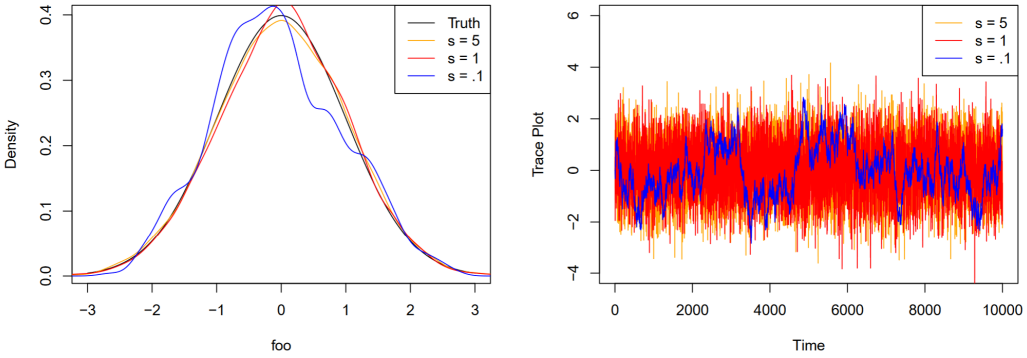

Figure: HMC draws from a $\mathcal{N}(0, 1)$ distribution

HMC Performance

- Under the normality assumption for ψ, we have $K(\psi) = -\log \phi(\psi)$
- $\phi(\cdot)$ is the standard normal probability density function
- Replacing our definitions for θ and ψ, acceptance rate is

$$\begin{aligned}\alpha(\theta^*, \psi^*; \theta^{(s)}, \psi^{(s)}) &= \frac{\exp\{-U(\theta^*) - K(\psi^*)\}}{\exp\{-U(\theta^{(s)}) - K(\psi^{(s)})\}} \\ &= \exp\{-H(\theta^*, \psi^*) + H(\theta^{(s)}, \psi^{(s)})\} = 1\end{aligned}$$

- Acceptance rate equals 1 due to energy conservation:

$$H(\theta^*, \psi^*) = H(\theta^{(s)}, \psi^{(s)})$$

- Intuition: HMC is an MH algorithm that always accepts!

Lecture outline

1. Motivating Hamiltonian Monte Carlo
2. Digression: Hamiltonian Dynamics
3. Hamiltonian Monte Carlo Algorithm
4. HMC in Practice

HMC Implementation

- So far, things look like HMC is a perfect algorithm
- Main issue: transition mapping T is actually not available
- Need to approximate the transition mapping
- Usual way in physics is to use Euler's approximation, but this does not perform well with small number of iterations
- Usual standard in practice: use Leapfrog integrator instead
- Idea is to start a given point and evolve system in discrete steps

HMC: Standard normal example

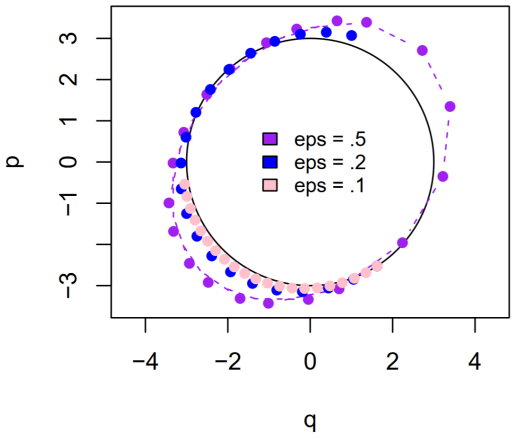

Figure: Euler's approximation

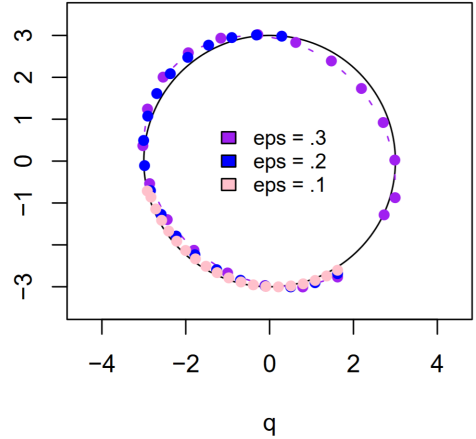

Figure: Leapfrog approximation

HMC Implementation: Summary

- By using gradient information of the posterior, HMC is fast and efficient
- It uses deterministic moves to make exploration of parameter space faster
- Simultaneously reduces autocorrelation due to this property
- Requires approximation using leapfrog integrator
- In practice, you need to tune:
  1. Mass matrix M
  2. Number of leapfrog steps L or step size ϵ

HMC: Standard normal example

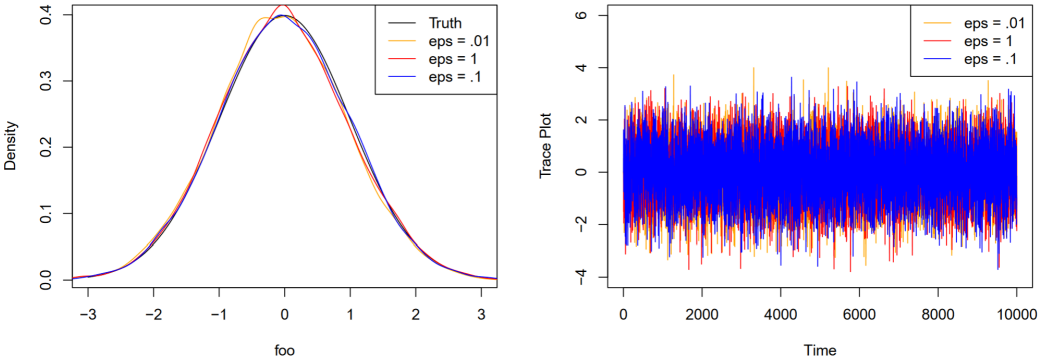

Figure: HMC draws from a $\mathcal{N}(0, 1)$ distribution using Leapfrog integrator

HMC: Standard normal example

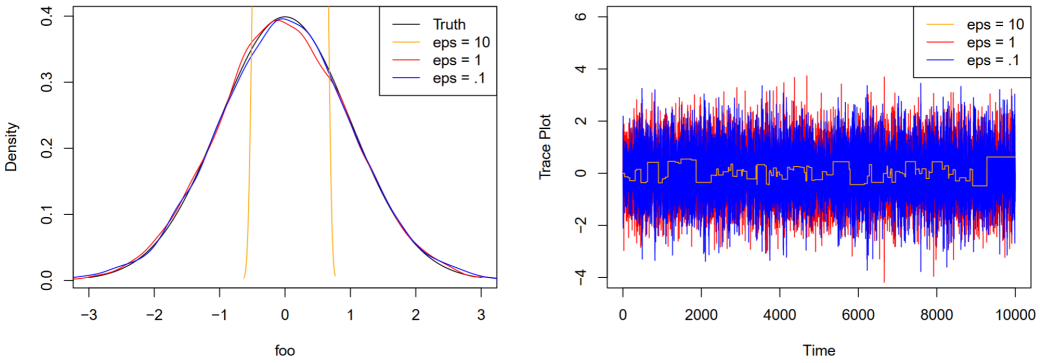

Figure: HMC draws from a $\mathcal{N}(0, 1)$ distribution using Leapfrog integrator