Hamiltonian Monte Carlo

M.A, S.B, A.L, L.N-H

University of Toronto

November 28, 2023

What are we doing?

Generate a sample using a **dynamical system**.

Why? There are plenty of easy to tune algorithms that are available to simulate a dynamical system.

The challenge is to set up the dynamical system in a way that allows for a sample to be generated.

Key Definitions

Dynamical System

For the purposes of this presentation we will be considering dynamical systems of this form: Assuming that $\mathbf{x} \in \mathbb{R}^n$

$$\frac{d}{dt}x_1(t) = f_1(\mathbf{x})$$
$$\frac{d}{dt}x_2(t) = f_2(\mathbf{x})$$
$$\vdots$$
$$\frac{d}{dt}x_n(t) = f_n(\mathbf{x})$$

Key Definitions

Phase Space and State

In the system from the slide above, the vector $\mathbf{x} \in \mathbb{R}^n$ is called a **state**. The set of all possible states is called a **phase space**

Key Definitions

Flow

Let $x_0 \in U$ be the initial state and $\mathbf{x}(t)$ solves the system with initial condition $\mathbf{x}(0) = x_0$.
We call the map

$$\phi : \mathbb{R} \times U \to U$$

$$(t, x_0) \xrightarrow{\phi} \mathbf{x}(t)$$

the flow of the system, denoted as $\phi(x_0, t)$.

Examples

Consider the following system:

$$\frac{d}{dt}x = \dot{x} = y$$

$$\frac{d}{dt}y = \dot{y} = -x$$

Examples

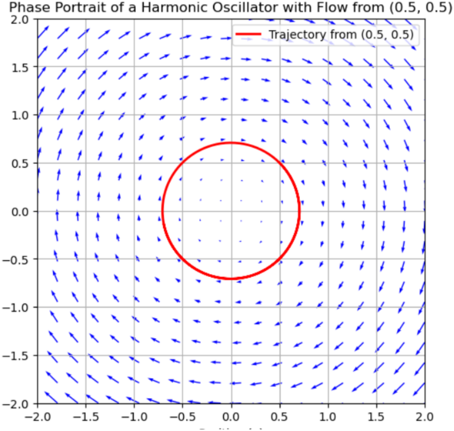


Figure: Phase Portrait with Flow (initial condition: $x_0 = \left(\frac{1}{2}, \frac{1}{2}\right)$

What is a Hamiltonian System?

Hamiltonain System

Suppose that we have the following dynamical system in R^{2n}, defined by

$$\dot{x_i} = \frac{\partial}{\partial y_i} H(\mathbf{x}, \mathbf{y}),$$

$$\dot{y_i} = -\frac{\partial}{\partial x_i} H(\mathbf{x}, \mathbf{y}),$$

where $(\mathbf{x}, \mathbf{y}) = (x_1, y_1, x_2, y_2, \ldots, x_n, y_n) \in \mathbb{R}^{2n}$ is a point in the phase space. The above is a Hamiltonian system, and $H(\mathbf{x}, \mathbf{y})$ is called the Hamiltonian of the system.

Why would we use a Hamiltonian system?

Hamiltonian systems are what are called conservative systems (they preserve energy). Mathematically, it can be shown that conservative systems have the following properties:

- Invariant to flow: $H(\mathbf{x_0}) = H(\phi(\mathbf{t}, \mathbf{x_0}))$
- At each initial condition $\mathbf{x_0}$, the flow $\phi(t, \mathbf{x_0})$ is either a fixed point, or a periodic orbit. So these systems have a nice structure.
- No attracting fixed points. That is, if $\mathbf{x_0}$ is not a fixed point, the flow $\phi(t, \mathbf{x_0})$ will never end up stuck at a fixed point.
- Sometimes it's the case that a Hamiltonian might be written as the sum of kinetic and potential energies, i.e. $H(\mathbf{x}, \mathbf{y}) = U(\mathbf{x}) + K(\mathbf{y})$

Setting Up the Dynamical System for simulation

The goal is to generate a sample for a k dimensional random vector $\theta \sim F_\Theta$ with density $f(\cdot)$ using a $2k$ dimensional Hamiltonian system $(i = 1, 2, ..., k)$:

$$\frac{d}{dt}\theta_i = \dot{\theta}_i = \frac{\partial H(\theta, p)}{\partial p_i}$$

$$\frac{d}{dt}p_i = \dot{p}_i = -\frac{\partial H(\theta, p)}{\partial \theta_i}$$

Setting Up the Dynamical System

In order to transform this system into something that can be used to generate a sample, we make 2 important assumptions.

1. Assume that the Hamiltonian function may be written as $H(\theta, p) = U(\theta) + K(p)$
2. Assume that θ comes from a distribution with probability density function $f(\theta)$

So if we are able to successfully simulate a dynamical system with these properties, then the resulting generated θ values will indeed be a sample for θ

Canonical Distribution - What should U(θ) be?

From statistical mechanics, we may assign a joint probability function to $(\theta, \mathbf{p})$:

$$\pi(\theta, \mathbf{p}) = \frac{1}{Z} e^{-H(\theta, \mathbf{p})} = \frac{1}{Z} e^{-U(\theta)} e^{-K(\mathbf{p})} = \underbrace{\exp[-U(\theta)]}_{\pi(\theta)} \underbrace{\frac{1}{Z} \exp[-K(\mathbf{p})]}_{\pi(\mathbf{p}|\theta)}$$

- We know what $\pi(\theta)$ is: $\pi(\theta) = f(\theta)$ so we let $U(\theta) = -\log(f(\theta))$
- $\pi(\theta, \mathbf{p}) = \frac{1}{Z} f(\theta) e^{-K(\mathbf{p})}$

So far we have:

$$\dot{\theta}_i = \frac{\partial K(p)}{\partial p_i}$$

$$\dot{p}_i = \frac{\partial \log(f(\theta))}{\partial \theta_i}$$

We only have half the story, we now restrict our attention to $K(p)$

Choice of Kinetic Energy

The first substantial obstacle that we come across is choosing the kinetic energy system.

Empirically, it seems to be the case that a quadratic form for the kinetic energy performs well (Betancourt, 2017). $K(p) = \frac{1}{2}p^T M^{-1}p$

- Now $\pi(p|\theta) = \frac{e^{-\frac{1}{2}p^T \boldsymbol{M}^{-1}p}}{|\boldsymbol{M}|^{\frac{1}{2}}(2\pi)^{\frac{k}{2}}}$ $\left(f(\mathbf{x}) = \frac{e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}}{|\boldsymbol{\Sigma}|^{1/2}(2\pi)^{k/2}} \right)$
- M is a tuning parameter of our model.

The System

$$\dot{\theta} = \frac{\partial K(p)}{\partial p_i} = p^T \mathbf{M^{-1}} p$$

$$\dot{p}_i = -\frac{\partial U(\theta)}{\partial \theta_i} = \frac{\partial \log(f(\theta))}{\partial \theta_i}$$

Simulating the flow: The leapfrog integration algorithm

Assuming $M = I$, given some starting point $(\theta(t), \mathbf{p})$, and step size ϵ:

$$\begin{aligned} p_i\left(t + \frac{\epsilon}{2}\right) &= p_t(t) - \frac{\epsilon}{2}\frac{\partial U}{\partial \theta_i}(\theta(t)) \\ \theta_i(t+\epsilon) &= \theta_i(t) + \epsilon p_i\left(t + \frac{\epsilon}{2}\right) \\ p_i(t+\epsilon) &= p_i\left(t + \frac{\epsilon}{2}\right) - \frac{\epsilon}{2}\frac{\partial U}{\partial \theta_i}(\theta(t+\epsilon)) \end{aligned}$$

Leapfrog Integration Examples

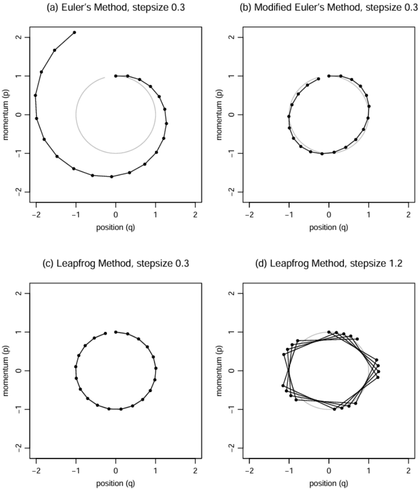


Figure: Leapfrog Algorithm (from Neal, 2002)

Correcting Numerical Error

- Sometimes the leapfrog algorithm can lead to numerical error
- The system is supposed to be energy preserving, we can use this fact to measure the numerical error
- One way to correct for numerical error is to use something similar to the metropolis algorithm.

Correcting Numerical Error

Let $\phi(t_0, \theta_0, \mathbf{p_0}) = (\theta_*, \mathbf{p}_*)$. The metropolis ratio is defined as $\alpha = \dfrac{\pi(\theta_*, \mathbf{p}_*)}{\pi(\theta_0, \mathbf{p_0})}$. Using this metropolis ratio, the following acceptance-rejection scheme is constructed:

1. Generate $u \sim U(0, 1)$:
2. If $u \leq \alpha$, then the numerical error is acceptable (we 'accept' $(\theta_*, \mathbf{p}_*)$)
3. If $u > \alpha$, then the numerical error is unacceptable (we 'reject' $(\theta_*, \mathbf{p}_*)$) and keep $(\theta_0, \mathbf{p_0})$)

The Algorithm

The following algorithm generates N observations from a distribution that has density $f(\theta)$:

1. Given some starting point $\theta_0 \in \mathbb{R}$, step size ϵ, covariance matrix $\mathbf{M}$, and leapfrog iteration count L store θ_0 in a list of samples $\mathbf{w}$: $\mathbf{w}[0] = \theta_0$.
2. Iteratively, for $i = 1, 2, ..., N$
    1. Set $\theta_{i-1} = \mathbf{w}[i-1]$
    2. Generate $p_0 \sim \mathcal{N}_k(\mathbf{0}, \mathbf{M})$
    3. Given initial condition (θ_{i-1}, p_{i-1}), simulate the flow $\phi^t(\theta_{i-1}, p_{i-1})$ and let $\phi^L(\theta_{i-1}, p_{i-1}) = (\theta_i, p_i)$
    4. Set $\mathbf{w}[i] = \theta_i$
    5. Increment i and repeat

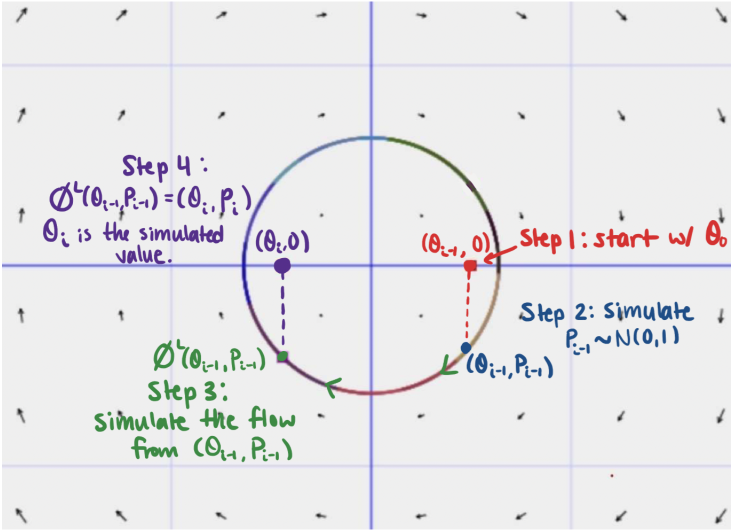
Example: Simulating from a standard normal distribution

Suppose $\Theta \sim \mathcal{N}(0,1)$. The density for $\mathcal{N}(0,1)$ is $f(\theta) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\theta^2}$. Assuming $K(p) = \frac{1}{2}p^2$. Following the derivations above, a dynamical system that can be used to simulate from a standard normal distribution is as follows:

$$\dot{\theta} = \frac{dK(p)}{dp} = p$$

$$\dot{p} = -\frac{dU(\theta)}{d\theta} = -\theta$$

Example: (Diagram)


Step 4:
$\phi^L(\theta_{i-1},P_{i-1})=(\theta_i,P_i)$
θ_i is the simulated value.
$(\theta_i,0)$
$(\theta_{i-1},0)$
Step 1: start w/ θ_0
Step 2: Simulate $P_{i-1}\sim N(0,1)$
(θ_{i-1},P_{i-1})
$\phi^L(\theta_{i-1},P_{i-1})$
Step 3: Simulate the flow from (θ_{i-1},P_{i-1})

Our Website for Simulations

We created a website for users to generate samples using the Hamiltonian Monte Carlo:

https://simran-bilkhu.shinyapps.io/Hamiltonian-Monte-Carlo-Algorithm/

This website allows us to simulate from a Hamiltonian Monte Carlo from the **Beta** distribution:

$$f(y|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}, \;\; 0 < y < 1$$

and the **Normal** distribution:

$$f(y|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y-\mu)^2}{2\sigma^2}}, \;\; -\infty < y < \infty$$

References

Neal, Radford M. MCMC Using Hamiltonian Dynamics. arXiv, June 9, 2012. http://arxiv.org/abs/1206.1901.

Betancourt, Michael. A Conceptual Introduction to Hamiltonian Monte Carlo. arXiv, July 16, 2018. https://doi.org/10.48550/arXiv.1701.02434.