STAT525 Project: Hybrid Monte Carlo

Yuxuan Zhang (yuxuanz8)

Abstract

The report introduces the background of Hybrid Monte Carlo and its algorithm. We ran the simulation for multivariate normal distribution using MHC with different parameters. In addition, we compared the results from MHC and Metropolis-Hasting algorithm and found simulation from MHC has better convergence and stable.

1 Introduction

Hamiltonian Monte Carlo , also known as Hybrid Monte Carlo (HMC) algorithm, was first developed in the late 1980s by Duane for the numerical simulation of lattice field theory [1]. It is a hybrid (molecular dynamics/Langevin) algorithm used to guide a Monte Carlo simulation. Then HMC started to appear in scientific computing textbooks. The algorithm is discussed in detail in Liu's book, Chapter 9 [2]. Neal's review in the *Handbook of Markov Chain Monte Carlo* introduced this approach into the mainstream of computational statistics in 2011 [3]. With significantly improved computing power, Hybrid Monte Carlo has demonstrated its powerful performance in physics, statistics and other applications.

Hamiltonian Monte Carlo is derived from simulation in physics using Hamiltonian equations. Molecular dynamics (MD) simulation is a deterministic procedure to integrate the equations of motion based on the classical mechanics principles (Hamiltonian equations). The trajectories has to be generated discretely to simulate moves in real time. Therefore, the time step is picked carefully. Monte Carlo simulation is to sample configuration from a Boltzmann distribution which is determined by the system's potential energy and temperature. The HMC uses MD to generate trail moves in a MCMC sampler.

2 Background Knowledges

Since Hybrid Monte Carlo adopted Hamiltonian equations into statistics computing, we will introduce basic knowledge and notions in the scenario of physics as follows to help us understand of the algorithm.

In physics, a conservation law states that a particular measurable property of an isolated physical system does not change as the system evolves over time [4]. The momentum p is $p = mv$. The kinetic energy of the system is defined as $k(p) = \frac{1}{2}mv^2$. Let $U(x)$ be the potential energy field of the system. Then the total energy of the system at a given time is

$$H(x, p) = U(x) + k(p) \tag{1}$$

The law of conservation says that the total energy is a constant in a closed system. Differentiating both sides of (1) with respect to t, we derive the Newton's law of motion from conservation law.

$$\dot{x}(t) = \frac{\partial H(x,p)}{\partial p} \tag{2}$$

$$\dot{p}(t) = -\frac{\partial H(x,p)}{\partial x} \tag{3}$$

The equation (2) describes the velocity is the partial derivative of kinetic energy with respect to momentum and (3) means that the force equals to negative gradient of potential energy.

Because compute cannot take care of continuous-time simulation, we have to utilize Taylor expansion to discretize the motion.

$$x(t+dt) = x(t) + \frac{p(t)}{m}dt + \frac{\dot{p}(t)}{2m}dt^2 + \cdots \tag{4}$$

$$p(t+dt) = p(t) + \dot{p}(t)dt + \frac{\ddot{p}(t)}{2}dt^2 + \cdots \tag{5}$$

The Taylor expansion has infinite elements which is impossible to compute in the real system. Therefore, we need to truncate it to approximate the true value. Here, we described *leaf frog* algorithm to numerically integrate the differential equations. For simplicity, we neglect mass m here.

$$p(t+\frac{1}{2}\Delta t) = p(t) - \frac{\Delta t}{2}\frac{\partial H}{\partial x}\bigg|_{x(t)} \tag{6}$$

$$x(t+\Delta t) = x(t) + \Delta t p(t+\frac{1}{2}\Delta t) \tag{7}$$

$$p(t+\Delta t) = p(t+\frac{1}{2}\Delta t) - \frac{\Delta t}{2}\,\frac{\partial H}{\partial x}\bigg|_{x(t+\Delta t)} \tag{8}$$

In addition, we will introduce concept of canonical ensemble briefly. A canonical ensemble is the statistical ensemble that represents the possible states of a mechanical system in thermal equilibrium with a heat bath at a fixed temperature. The canonical ensemble assigns a probability P to each distinct micro state given by the following exponential:

$$P = \frac{1}{Z}e^{-\frac{E}{kT}}$$

Z is the normalization constant, E is the total energy, k is the Boltzmann constant and T is the temperature. Energy in Hamiltonian system also follow the conservation of mechanical energy, and could be wrote as canonical ensemble as following:

$$\Pi(x,p) \propto e^{-\frac{U(x)+k(p)}{T}} = const. \tag{9}$$

3 Hybrid Monte Carlo

A problem of Molecule Dynamics is the strict requirement of a small time-step size. While proposal distribution in Metropolis Monte Carlo sampler is very likely to be rejected if the space is occupied by other molecule. To overcome some of these difficulties, Duane et al. introduced the method of hybrid Monte Carlo which combines the basic idea of MD (i.e., proposing new positions based on Hamiltonian equations) and the Metropolis acceptance-rejection rule to produce Monte Carlo samples from a given target distribution.

Let us integrate all the knowledge introduced in the previous section and see how Hybrid Monte Carlo algorithm is constructed in statistical cases.

Suppose we want to draw samples from $\pi(x) \propto exp[-U(x)]$. If we can draw samples (x, p) from $\Pi(x, p) \propto exp[-H(x, p)]$, and marginally $x \sim \pi$. p is the momentum variable which is the auxiliary variable in the statistics. Transitionally, it is defined as

$$k(p) = \sum_i \frac{p_i^2}{2m_i} \tag{10}$$

Then,

$$\Pi(x, p) \propto exp[-H(x, p)] = exp[-U(x) - k(p)] = exp[-U(x)]exp[-k(p)] \tag{11}$$

$$\Pi(x, p) = \pi(x)\phi(p) \tag{12}$$

The equation 12 shows x and p are independent, and p follows the Gaussian distribution, $\phi(p) \propto exp(-\|p/\sqrt{m}\|^2/2)$. That is the reason why we don't keep the record of p in the following algorithm.

The m_i means the mass of the ith component. We can have an intuitive feeling that the more heavier is the object, the more slower it moves. The efficiency of HMC can be improved by setting m_i differently for p_i according to the properties of x_i. In the next experiment section, we neglect m or, in other words, we treat m_i with equal value which will disappear in the implementation.

The Hamiltonian path is time-reversible. More specifically, Hamiltonian equations of motion are invariant under a reversal of the direction of time. The HMC also integrates Metropolis-Hastings(M-H) Algorithm which contains proposal and rejection. We utilized Hamiltonian transition as the proposal for M-H so that it can be removed in the fraction.

The HMC algorithm can be implemented as follows.

Algorithm 1: Hybrid Monte Carlo

```
Suppose at time t, $x^t = x$ ;
Output: $x^{t+1}$ ;
1. Generate a new momentum vector $p$ from the Gaussian distribution ;
2. Run the leap-frog algorithm, starting from $(x, p)$ for L steps to obtain $(x', p')$ ;
3. Generate u ~ Uniform[0,1] ;
4. Let $A(x', p'|x, p) = exp(-\frac{E'-E}{T})$ ;
if  $min(A, 1) \geq u$ then
    $x^{t+1} = x'$ ;
else
    $x^{t+1} = x$ ;
end
```

4 Application of HMC in Statistics

4.1 2D-Gaussian Distribution

Problem Generate samples from the given two-dimensional Gaussian distribution with mean $\mu = (0, 0)^T$ and covariance matrix $\Sigma = \begin{pmatrix} 1 & 0.4 \\ 0.4 & 2 \end{pmatrix}$.

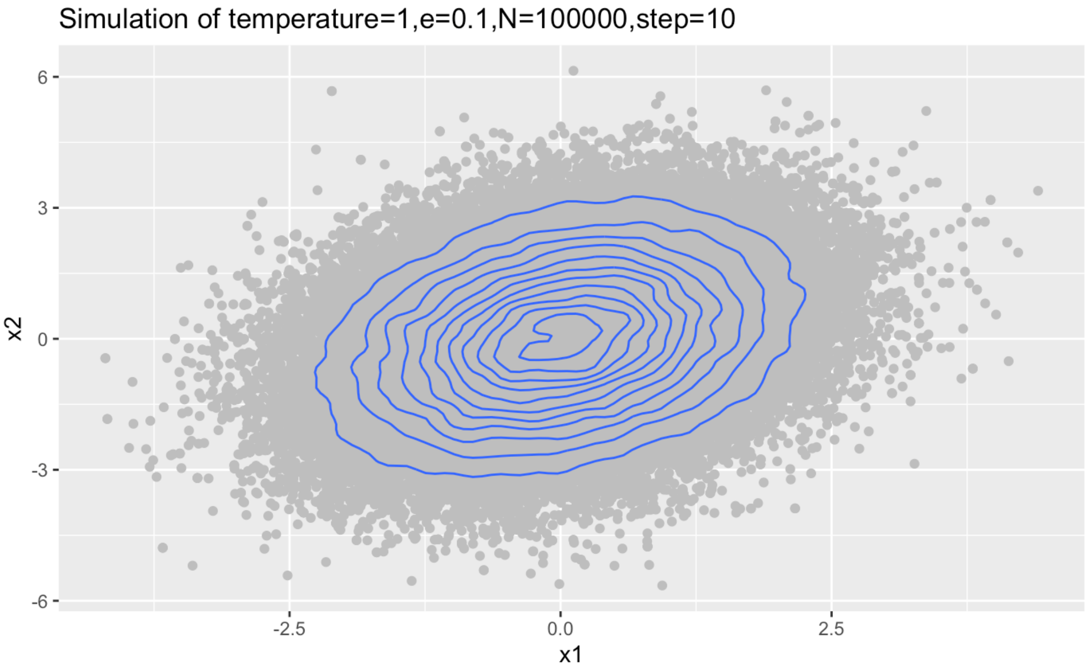

Figure 1: Visualization of Simulation

For simplification, we set mean μ to be zero. If $\mu \neq 0$, X can be transformed by $X = \sigma Z + \mu$ in which Z is the standard normal deviate. I wrote a HMC function in R to pass parameters $T = 1$, $\Delta t = 0.1$, $N = 100k$ and $L = 10$ steps in the leaf-frog algorithm. As you can see in the figure 1, though the contour is not smooth, the shape of the bi-variate normal distribution is correct. The estimated covariance matrix is

$$\hat{\Sigma} = \begin{pmatrix} 0.9957374 & 0.3942664 \\ 0.3942564 & 1.9921759 \end{pmatrix}$$

Let us see the influence of step length in the leaf-frog algorithm. We limit the iteration of sampling at 2000 to see its effect more apparently. In the figure 2, there is four scatter plots with step 1, 5, 10 and 20. The red line is ellipse of true normal distribution. We can see that when computing the integrator with only one step, samples intend to cluster together and have relatively bad movement. From step 5 to 20, the improvement is obvious. The samples scatter the whole area and more samples appear outside the ellipse.

Four covariance matrices is stated as follows. The covariance of simulation with step has largest deviation with the true covaiance.

$\hat{\Sigma}_1 = \begin{pmatrix} 0.8908488 & 0.5890157 \\ 0.5890157 & 1.0299367 \end{pmatrix}$	$\hat{\Sigma}_2 = \begin{pmatrix} 0.9485389 & 0.392125 \\ 0.3921250 & 2.659811 \end{pmatrix}$
$\hat{\Sigma}_3 = \begin{pmatrix} 1.0277220 & 0.4457005 \\ 0.4457005 & 2.0177936 \end{pmatrix}$	$\hat{\Sigma}_4 = \begin{pmatrix} 1.0503215 & 0.3569569 \\ 0.3569569 & 1.9798417 \end{pmatrix}$

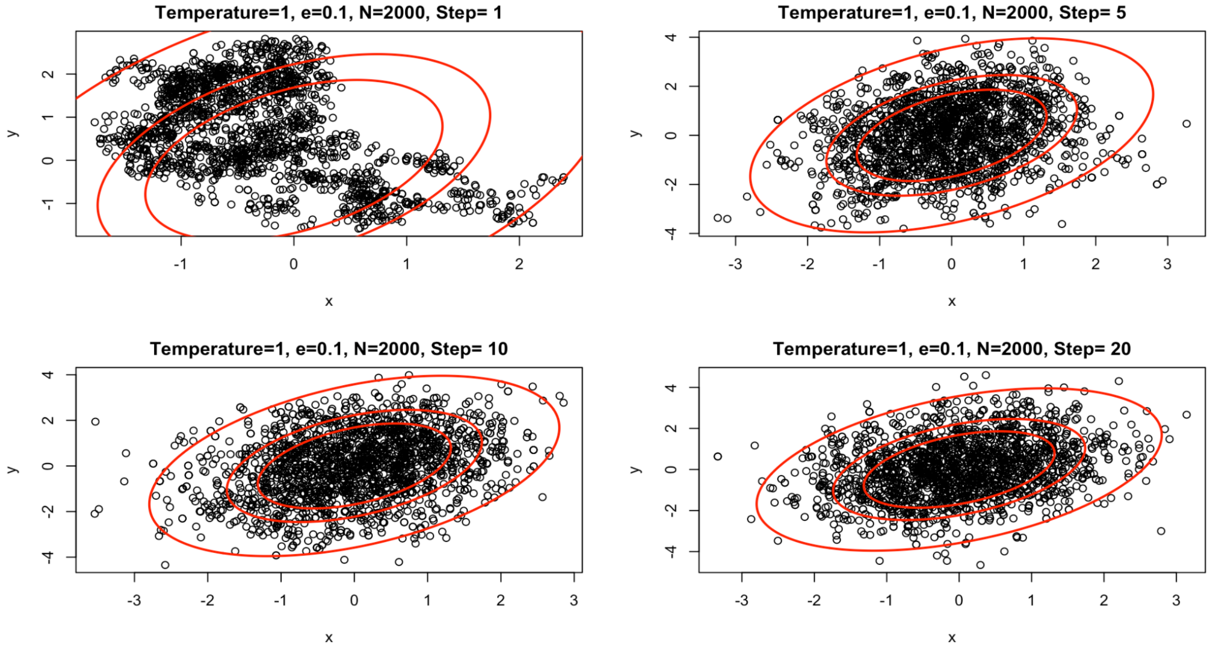

Figure 2: Simulation comparison of different step length

The initial values of previous experiments are generated around the mean. Although our target distribution is the convex, we are still interested in the efficiency of sampling. In the figure 3, the arrows showed the trajectory of the motion in 40 samples. The initial x is $(4, 8)$ which is far away from the mean $\mu = (0, 0)$, $T = 5, \Delta t = 0.2, step = 5$. But it will move to the center soon and spread in the 2-dimensional plane. We do not need to worry about the initialization in this case.

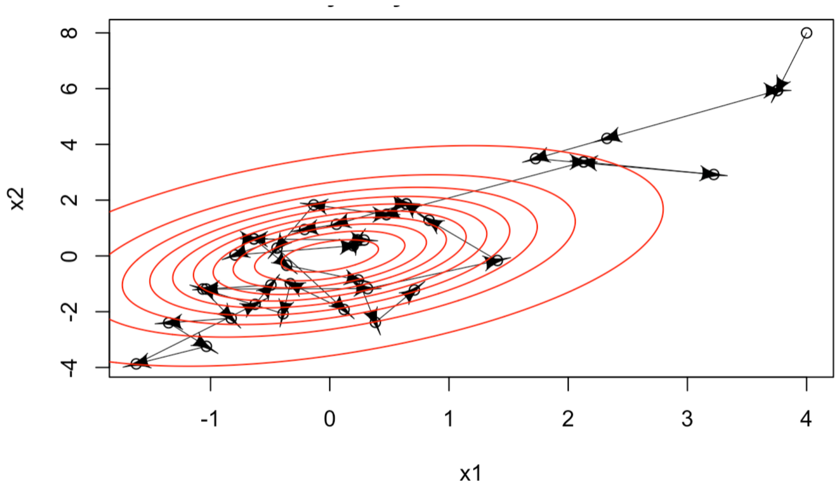

Figure 3: Trajectory of the motion

4.2 Metropolis-Hastings Algorithm

We ran the Metropolis-Hastings algorithm on the same problem and generated 100k samples. The range of x is larger than samples from MHC as we can see in the figure 4. One of the reasons is that bi-variate normal distribution is used as the proposal distribution with identity matrix as the covariance matrix. In HMC, we can control the step length L and Δt to generate x'. Another superiority of HMC is shown when comparing figure 3 and 5. It is more like random work in the Metropolis-Hastings algorithm and needs more movements before entering the center. Generating a new point from the proposal distribution does not take advantage of its differential, therefore, samples from M-H is more spread over the plane and acceptance rate is lower than MHC.

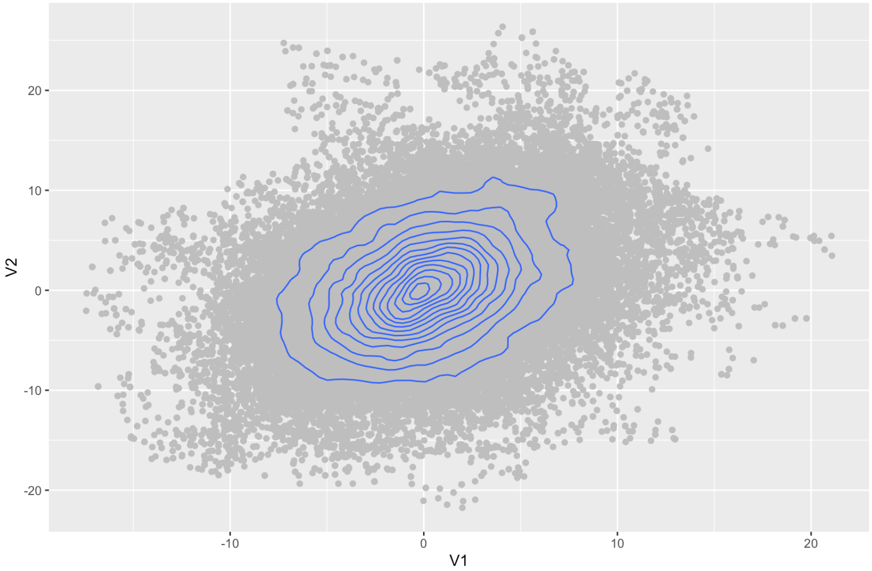

Figure 4: Simulation of Metropolis-Hastings Algorithm

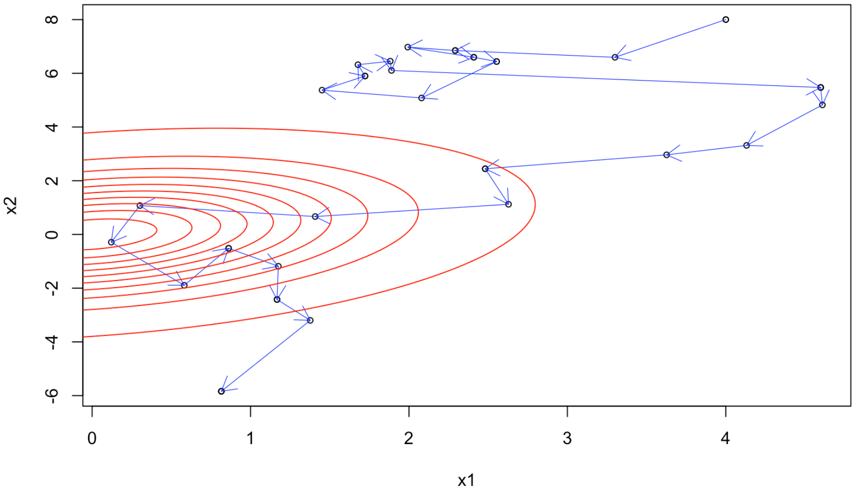

Figure 5: Trajectory in Metropolis-Hastings Algorithm

5 Discussion

From the experiments, HMC algorithm is better than M-H algorithm, but with step length L increase, the running time will increase. M-H is much more easier to implemented as long as selecting the appropriate proposal distribution. On the other hand, leaf-frog step in the HMC needs to compute $\frac{\partial H}{\partial x}$ which would be a difficult part if the target distribution is hard to get the differential. There are also several parameters to play with in the MHC. In the experiments, we simulated samples from different temperature separately. Furthermore, the parallel tempering method is useful for some problems.

References

[1] Simon Duane, A.D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics Letters B*, 195(2):216 – 222, 1987.

[2] Jun S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer Publishing Company, Incorporated, 2008.

[3] R.M. Neal. *MCMC using hamiltonian dynamics*. CRC Press, Department of Statistics, University of Toronto, 2011.

[4] Wikipedia contributors. Conservation law — Wikipedia, the free encyclopedia, 2020. [Online; accessed 5-May-2020].