Hamiltonian Monte Carlo: Using Hamiltonian dynamics to sample in $\mathbb{R}^d$

Calvin Yee Fong, Elijah French

April 23, 2023

1 Abstract

The following is a project on Hamiltonian dynamics and its application to Monte Carlo sampling methods. Background knowledge on Hamiltonian systems, sampling algorithms, why sampling is difficult in high dimension and stochastic processes is presented. After this, applications, drawbacks, and strengths of the method are discussed. The project concludes with a discussion on why Hamiltonian Monte Carlo is beneficial compared to traditional Monte Carlo methods in high dimensional settings, its problems and potential research ideas.

Contents

2 Introduction

Sampling from probability distributions is essential in testing models, making predictions, and generating novel theories. Markov chain Monte Carlo (MCMC) is one of the most well-known methods for sampling from multivariate distributions. However, difficulties arise for MCMC when some atypical distributions are considered. Chains can get stuck and fail to explore much of the desired distribution, greatly increasing sample error. Introducing Hamiltonian mechanics gives a way to ensure exploration by having a momentum term that can push the chain to explore more of the distribution. However, it too can run into difficulties with parameter choice and computational efficiency.

3 Literature Review

For some historical context, MCMC was first introduced by Metropolis and others in 1953. [7] It was initially used in a physical context to model molecules. More deterministic forms of molecular movement that used Hamiltonians were later formed. These approaches were combined with Hamiltonian Monte Carlo by Duane, Kennedy, Pendleton, and Roweth in 1987. Further developments were made to HMC methods in lattice field theory. Statistical applications of HMC, the focus of this project, began to be developed in the mid 90s. [7]

4 Monte Carlo Integration

Sampling from distributions can be used to make calculations. Many important integrals do not have closed forms. Approximating them is then necessary. The following well known theorem in statistics gives a straightforward way to do this.

Theorem 4.1 (Strong Law of Large Numbers (SLLN) [3]) *Given a sequence of pairwise independent identically distributed real-valued random variables* $(Y_k)_{k\in\mathbb{N}}$ *such that* $\mathbb{E}|Y_i| < \infty$.

$$\frac{1}{n}\sum_{k=1}^{n} Y_k \to \mathbb{E}[Y_i]$$

Now, suppose we wanted to estimate the definite integral of some real-valued function $g : [a, b] \to \mathbb{R}$. Define Y to be the real-valued random variable for the uniform distribution on $[a, b]$ which has a probability density function $f_Y(y) = \frac{1}{b-a}$ for $y \in [a, b]$. Then,

$$\int_a^b g(y)dy = \int_a^b ((b-a)g(y))\frac{1}{(b-a)}dy = \int_a^b h(y)f_Y(y)dy$$

Where $h(y) = (b-a)g(y)$. By definition of the expectation of functions of random variables, $\mathbb{E}[h(Y)] = \int_a^b h(y)f_Y(y)dy$. It then follows [5] by the SLLN that

$$\frac{1}{n}\sum_{k=1}^{n} h(Y_k) \to \mathbb{E}[h(Y_i)] = \int_a^b h(y)f_Y(y)dy = \int_a^b g(y)dy$$

Where $(Y_k)_{k\leq n}$ is a random sample from Y. This convergence means we have a way to estimate definite integrals using samples of random variables. That is, we can take a random sample $\{y_j\}_{j\leq k}$ from $[a, b]$ evaluate the sample with $h(y_j) = (b-a)g(y_j)$ and take the average $\frac{1}{k}\sum_{i\leq k} h(y_j)$. The same method can be used in general to estimate integrals in multi dimensional settings.

Thus, given a random sample, we can estimate a definite integral. The question now becomes: how can one attain a random sample for a distribution? Sampling is easy to do for univariate probability distributions. One can merely sample from a uniform distribution on $[0, 1]$, invert the CDF of the proposed distribution, and achieve a random sample for the distribution by plugging the uniform sample into the inverse of the CDF. The problem gets harder for multidimensional distributions. In the following sections, MCMC, a method for sampling from more general distributions will be introduced. Before this is done, some background on Markov Chains will be given.

5 Markov Chains

Assume we are trying to sample from some probability distribution Γ defined on a state space $\Omega \subseteq \mathbb{R}^d$ with probability density ρ.

Definition 5.1 *A (continuous state)* ***Markov Chain*** *[10] is a sequence of random variables* $\{X_n\}_{n\in\mathbb{N}} \subset \Omega$ *such that for any* $A \subseteq \Omega$

$$\mathbb{P}(X_{n+1} \in A | X_n = i_n, ..., X_0 = i_0) = \mathbb{P}(X_{n+1} \in A | X_n = i_n)$$

Under this definition, at a given time $t = n$ the distribution of the next random variable in the sequence only depends on the position of the most recent point in the sequence. This is why Markov chains are commonly referred to as "memory-less" [5]. The question now becomes, how does a Markov Chain move around a distribution?

Definition 5.2 *Given a* $X_n = \mathbf{x}_n \in \Omega$ *for* $n \in \mathbb{N}$*, we define a density function [10] for* X_{n+1} *on* Ω *by* $p_{\mathbf{x}_n}(\mathbf{y})$ *where*

$$\int_\Omega p_{\mathbf{x}_n}(\mathbf{y}) d\mathbf{y} = 1$$

Larger step transitions [10] are similarly defined with density functions on Ω. That is, given $X_n = \mathbf{x}_n$, how is X_{n+k} distributed? We define the k-step transition density $p^{(k)}_{\mathbf{x}_n}(y)$ where

$$\int_\Omega p^{(k)}_{\mathbf{x}_n}(\mathbf{y}) d\mathbf{y} = 1 \quad \forall \mathbf{x}_n \in \Omega$$

For a Markov chain $\{X_n\}_{n\leq k}$ to be a random sample that follows Γ it must be that $p_{\mathbf{x}_n} = \rho$ independent of $\mathbf{x}_n$. This cannot always be guaranteed, however, one can hope for a Markov chain's densities to follow ρ in the limit. We first define a distribution on Ω that would ensure that subsequent terms are distributed in the same way.

Definition 5.3 *A stationary distribution* π *[10] for a Markov chain* $\{X_n\}$ *with transition densities* $p_{\mathbf{x}}$ *for* $\mathbf{x} \in \Omega$ *is a density function defined on* Ω *such that*

$$\pi(\mathbf{y}) = \int_\Omega \pi(\mathbf{x}) p_{\mathbf{x}}(\mathbf{y}) d\mathbf{y}$$

By properties of multi-step transition densities (cite), this would imply that $X_0 \sim \pi \Rightarrow X_k \sim \pi \ \forall k \in \mathbb{N}$. Further,

Definition 5.4 π *is said to be* ***reversible*** *[10] for a Markov chain transition density* p *if*

$$\pi(\mathbf{x}) p_{\mathbf{x}}(\mathbf{y}) = \pi(y) p_{\mathbf{y}}(\mathbf{x})$$

$\forall \mathbf{x}, \mathbf{y} \in \Omega$

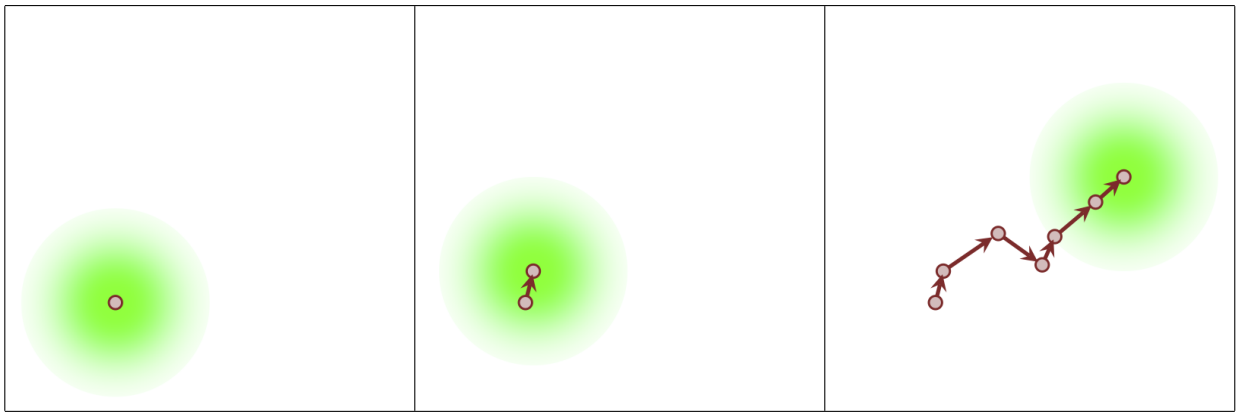

Figure 1: The progession of a continuous state Markov chain with transition probability highlighted in green [2]

Reversibility implies stationarity [5]. $\lim_{k\to\infty} p_{\mathbf{x}}^{(k)}(\mathbf{y})$ is reversible and so is stationary as well. To achieve a random sample from Γ using Markov chains, it suffices to have transition probabilities that are reversible with respect to ρ. At which point the limiting distribution of the chain would be ρ. A sample could then be constructed by going far enough into the Markov Chain where the distribution of subsequent points would approximately be of the underlying distribution. [5]

6 MCMC and the Metropolis-Hastings Algorithm

Monte Carlo integration uses Markov chains and their limiting distributions to generate samples. If one can ensure that the limiting distribution of a chain is the desired distribution, an approximate sample can be made by going far enough in the chain to where they would essentially be randomly sampled from the underlying distribution.

In practice, computations must stop at some point. Therefore, the chain needs to be efficient in how it "picks" subsequent points in the sequence. That is, one should ensure that the next point in the sequence would be a better choice as it relates to exploring the typical set of the distribution. Using specified transition densities for the chain, the Metropolis-Hastings algorithm proposes a new point and rejects or accepts the point depending on whether it is a "better".

Definition 6.1 *The **Metropolis Hastings Algorithm** [8] is an algorithm where given $X_n = \mathbf{x}_n$, and a transition density p*

- *A proposal $X'_{n+1} \in \Omega$ is generated from the distribution $p_{\mathbf{x}_n}$*
- $\alpha = \min\{\frac{\pi(X'_{n+1})p_{\mathbf{x}_n}(X'_{n+1})}{\pi(X_{n+1})p_{X'_{n+1}}(\mathbf{x}_n)}, 1\}$ *is defined*
- $X_{t+1} = \begin{cases} Y_t & \text{with probability } \alpha \\ \mathbf{x}_n & \text{with probability } 1-\alpha \end{cases}$ *is taken*

The success of this algorithm heavily depends on the choice of transition density p. For instance, a transition density which is hard to sample from could make life even more difficult. Nonetheless, a multi-variate normal distribution is typically chosen. That is, $p_{\mathbf{x}} \sim N(\mathbf{x}, \Sigma)$ where Σ is a chosen covariance matrix. A proper choice of transition density ρ will be the concern of the rest of the paper.

6.1 Typical Sets of Distributions

Given a sample space $Q \subseteq \mathbb{R}^d$ and associated target distribution π, suppose we are interested in knowing some arbitrary expectation, $\mathbb{E}_\pi[f]$ for some function f over that sample space. Notice that this arbitrary expectation $\mathbb{E}_\pi[f]$ also includes any probability statement $\mathbb{P}(\pi(\mathbf{q}) \in E)$ for some well defined event $E \subseteq Q$. To see this, we can just take $f = I_E$ to be the indicator function that event E occurred and so $\mathbb{E}_\pi[I_E] = \int_Q I_E(\mathbf{q})\pi(\mathbf{q})d\mathbf{q} = \int_E \pi(\mathbf{q})d\mathbf{q} = \mathbb{P}(\pi(\mathbf{q}) \in E)$. What this conceptual question highlights is that estimating an arbitrary expectation is an inherently difficult task as doing so allows us to make definitive probability statements of some a distribution.

However, this is not to say that expectation estimation is impossible. Notice that by the Strong Law of Large numbers, we have good theoretical guarantee that the sample expectation $\sum_{i=1}^n f(\mathbf{q}_i)/n$ converges to the true population expectation $\mathbb{E}_\pi[f]$ if we can sample from π. In other words, we managed to reduce our expectation estimation problem into a sampling problem instead. While this viewpoint gives us a new perspective into our problem at hand, let's make sure we can grasp what it means to accurately estimate $\mathbb{E}_\pi[f]$.

To accurately estimate $\mathbb{E}_\pi[f]$, we should be concerned with where the majority of density is located in the distribution. It may initially appear as though a majority of density for a distribution is located around its modes. However, this is not the case. To see why, lets take a look at the formal definition of $\mathbb{E}_\pi[f]$ and when f is some indicator. Doing so gives us:

$$\mathbb{E}_\pi[I_E] = \int_E \pi(\mathbf{q})\ d\mathbf{q} \tag{1}$$

Notice that in (1), both the probability, $\pi(\mathbf{q})$ and the differential volume, $d\mathbf{q}$ both contribute to the expectation. As such, an accurate expectation estimate would require both an accurate probability estimate and volume estimate. From this framework, we can see how using the modes of a probability distribution is erroneous as even regions with high probability could have very minor to no volume. This is best illustrated in the following figure.

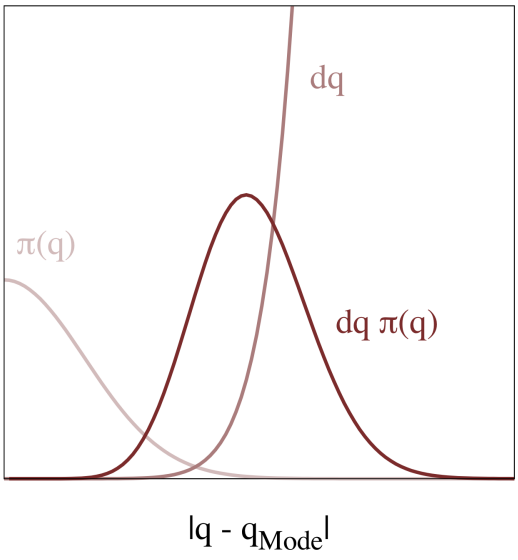

Figure 2: A typical set of some probability distribution. We can see that as we get further away from the mode, the probability $\pi(\mathbf{q})$ decreases but that there is also more volume $d\mathbf{q}$ as well. This leads to the typical set, which is the region where most of the expectation is concentrated on, not being associated to where the mode is at all. [2]

6.2 Curse of High Dimensionality

Using the typical set framework, we saw how sampling from it will produce an accurate estimate for the expectation of interest. However, care must also be taken in accounting for the volume of high expectation especially in high dimensions. This is due to the fact that as the dimension increases, volume increases exponentially which is analogous to $\text{vol}([-r,+r]^d) = (2r)^d$. Suppose that the probability distribution has compact support and that its typical set is concentrated in the central d-dimensional hypercube. Notice that as presented in Figure 3, finding the correct region that contains the typical set becomes much more difficult with increasing dimension. Thus for large d, even locating the typical set is like finding a needle in a haystack.

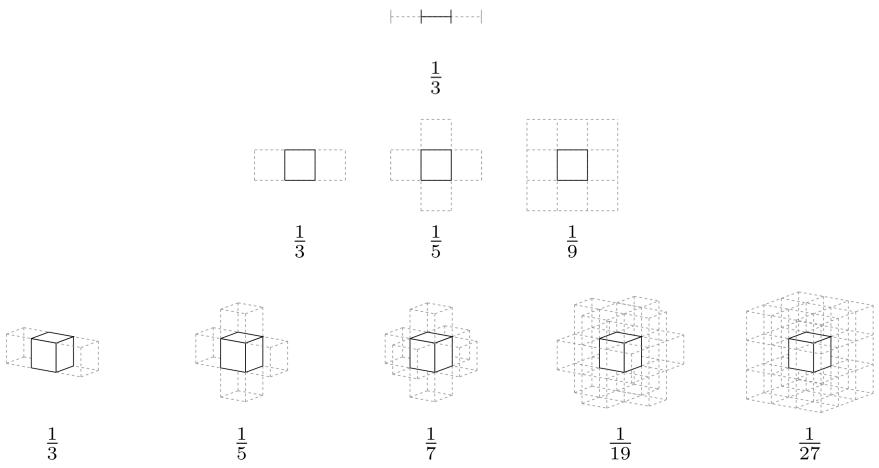

Figure 3: Demonstration of The Curse of High Dimensionality. [2]

7 Hamiltonian Monte Carlo

As mentioned, the choice of transition density p greatly affects the efficacy of the Monte Carlo method [11]. To be efficient in our sampling, we want to sample from as diverse a portion of the distribution as possible. Taking large steps without disturbing the distribution can be difficult. Hamiltonian Monte Carlo attempts to fix some of the problems seen when using other transition probabilities. In particular, the use of momentum and trajectories from a Hamiltonian can help deal with the staying on the typical set and the curse of high dimensionality.

Hamiltonian dynamics operate on a d-dimensional space vector $\mathbf{x}$ with an additional d-dimensional momentum vector ζ [4]. Thus, a Hamiltonian H acts on a $2d$ space with inputs $(\mathbf{x}, \zeta)$ [7]. As with any Hamiltonian [4] its dynamics are governed by,

$$\frac{d}{dt}\mathbf{x} = \nabla_{\zeta} H(\mathbf{x}, \zeta)$$
$$\frac{d}{dt}\zeta = -\nabla_{\mathbf{x}} H(\mathbf{x}, \zeta)$$

A special case of Hamiltonian dynamics is usually considered for Hamiltonian Monte Carlo. In particular that of an additively separable function [7]

$$H(\mathbf{x}, \zeta) = U(\mathbf{x}) + K(\zeta)$$

For a distribution ρ defined on Ω, it is usually assumed that $U(\mathbf{x}) = -\ln(\rho(\mathbf{x}))$ and $K(\zeta) = \frac{1}{2}\zeta^T\mathbf{I}\zeta$ that is, $H(\mathbf{x}, \zeta) = -\ln(\rho(\mathbf{x})) + \frac{1}{2}\zeta^T\mathbf{I}\zeta$. This will be used to construct a density function for the Markov chain used for sampling in hopes to deal with the curse of high dimensionality.

7.1 HMC Algorithm

The ideal HMC Algorithm will be outlined. Given a specified Hamiltonian H and a point and momentum $(\mathbf{x}, \zeta)$ define ϕ_T as the solution operator that evolves $(\mathbf{x}, \zeta)$ under H for T time. The idealized Hamiltonian Monte Carlo sampling method [11] would be completed by first choosing a starting point X_0 in the distribution (typically a node of the distribution). Then, given a $X_n = \mathbf{x}_n$, a momentum $\zeta_{n+1} \sim N(\mathbf{0}, M)$ is sampled. With this, X_{n+1} is chosen to be the position output of $\phi_T(\mathbf{x}_n, \zeta_{n+1})$. This is then repeated until a sufficient sample size is reached for convergence. The coupled sample for momentum and subsequent evolution under ϕ is the transition density.

An extension to this method can be made by introducing a similar acceptance criterion to Metropolis Hastings [2] where α is modified to be

$$\alpha(\phi_T(\mathbf{x}_n, \zeta_{n+1}), (\mathbf{x}_n, \zeta_n)) = \min\{1, \exp(H(\mathbf{x}_n, \zeta_n) - H(\phi_T(\mathbf{x}_n, \zeta_{n+1})))\}$$

7.2 Properties of Hamiltonians

Several properties of Hamiltonians will be necessary for later discussion on the efficacy of the method. Firstly, a Hamiltonian will have conserved quantity along solutions [4]. That is,

$$\frac{d}{dt}H(\mathbf{x}, \zeta) = \nabla_{\mathbf{x}} H \frac{d}{dt}\mathbf{x} + \nabla_{\zeta} H \frac{d}{dt}\zeta = \nabla_{\mathbf{x}} H \nabla_{\zeta} H - \nabla_{\zeta} H \nabla_{\mathbf{x}} H = 0$$

This means that a transition density defined using the Hamiltonian is time independent. Secondly, by Liouville's theorem [4], we know that total phase-space volume evolving under Hamiltonian dynamics will be constant. This means that we don't need to account for changes in volume in the acceptance probabilities [7]. Furthermore, a Hamiltonian trajectory remains on the same energy level set [2]. Lastly, Hamiltonian dynamics are reversible [7]. That is, there exists an inverse of ϕ, ϕ^{-1}. This reversibility property allows for the Markov chain defined by these updates to be reversible and thus have a limiting distribution.

7.3 Solving The Hamiltonian System

The exact form of ϕ cannot be found in general. To perform HMC in practice, one usually needs to solve ϕ approximately [2]. The leapfrog simplectic integrator method can be used when the Hamiltonian's position and momentum components are separable.

Definition 7.1 *Given parameters T and ϵ (run time and step-size respectively) and starting position and momentum $(\mathbf{x}, \eta)$ the **leapfrog integrator** [2]: uses the following algorithm to estimate $\phi_T(\mathbf{x}, \eta)$: Take $(\mathbf{x}_0, \zeta_0) = (\mathbf{x}, \zeta)$ and for $0 \leq k \leq \lfloor \frac{T}{\epsilon} \rfloor$*

- $\zeta_{k'} = \zeta_k - \frac{\epsilon}{2}\frac{\partial V}{\partial q}(\mathbf{x}_k)$
- $\mathbf{x}_{k+1} = \mathbf{x}_k + \epsilon \zeta_{k'}$
- $\zeta_{k+1} = \zeta_{k'} - \frac{\epsilon}{2}\frac{\partial V}{\partial q}(\mathbf{x}_{k+1})$

Then, $\phi_T(\mathbf{x}, \zeta) \approx \hat{\phi}_T(\mathbf{x}, \zeta) = \mathbf{x}_{(\lfloor \frac{T}{\epsilon} \rfloor)}$

The leapfrog integrator travels in the direction specified by the momentum sample for a small amount of time, takes a new momentum variable dependent on the distribution, and then travels in that direction. This continues until the specified time limit is reached. Importantly, the Leapfrog method ensures the preservation of volume on the space [2].

This integrator and the idealized algorithm for HMC can be combined to find a general, applicable method to HMC sampling.

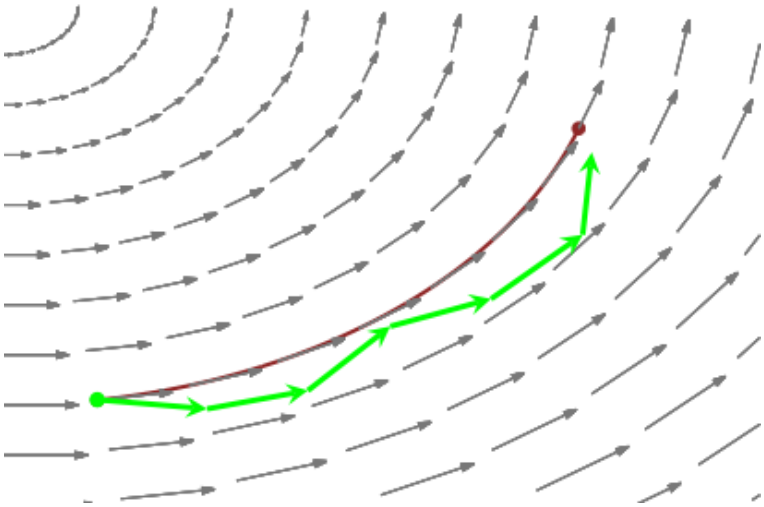

Figure 4: A true trajectory of the Hamiltonian given in magenta, with the approximate trajectory given by the leapfrog integrator in green. [2]

Definition 7.2 *(**Hamiltonian Monte Carlo Algorithm**) Given the probability distribution* Γ *defined on* $\Omega \subseteq \mathbb{R}^d$ *with density function* ρ *define a Hamiltonian* $H(\mathbf{x}, \zeta) = -\ln(\rho(\mathbf{x})) + \frac{1}{2}\zeta^T \mathbf{I}\zeta$. *Taking an initial point* $X_0 = \mathbf{x}_0$, *run time* T, *and step size* ϵ *a general algorithm can be defined as*

- *Given* $X_n = \mathbf{x}_n$ *a momentum* $\zeta'_{n+1} \sim N(\mathbf{0}, \mathbf{I})$ *is sampled*
- *Using the leapfrog integrator method with run time* T *and step size* ϵ, *the position portion* X'_{n+1} *of* $\hat{\phi}_T(\mathbf{x}_n, \zeta)$ *is proposed as the next sample*
- $\alpha = \min\{1, \exp(H(\mathbf{x}_n, \zeta_n) - H(\hat{\phi}_T(\mathbf{x}_n, \zeta_{n+1})))\}$ *is defined*
- *Then take* $X_{n+1} = \begin{cases} X'_{n+1} & \text{with probability } \alpha \\ \mathbf{x}_n & \text{with probability } 1-\alpha \end{cases}$
- *Repeat*

This process is repeated until stationarity at ρ is approached (warm up period). At which point, the process is continued until a sufficient sample size is found. This algorithm is used for sampling in the following sections.

8 Results

Given that the transition density used in MCMC is usually a guassian with unit variance, it typically has poor performance in sampling from a distribution with high correlation. This usually manifests via low acceptance rates (rejections in the Metropolis-Hastings Algorithm) and poor exploration of the typical set due to the correlated structure. The following Figure illustrates a practical use of HMC. That is how HMC can be used to sample from a highly correlated distribution.

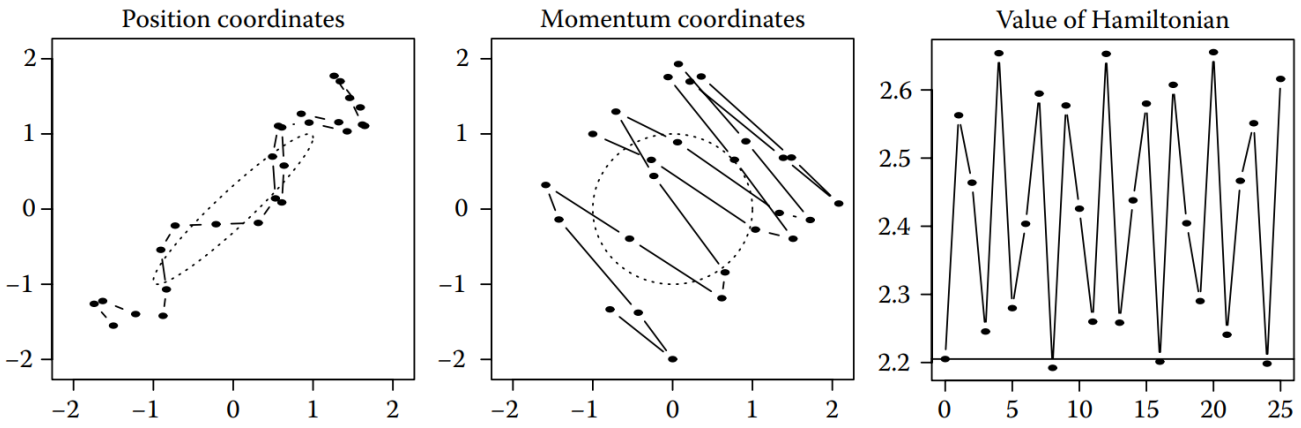

Figure 5: A trajectory for a two-dimensional Gaussian distribution with a correlation of 0.95, simulated using 25 leapfrog steps and a stepsize $\epsilon = 0.25$. The ellipses plotted represent one standard deviation from the mean. [7]

We can see that HMC samples generated in the position coordinates plot fallows the correlated guassian shape pretty well. Furthermore, the momentum samples do fallow a standard guassian shape as chosen. Finally, we can see that the leapfrog integrator used with the specified parameter choices does a decent job at preserving the Hamiltonian. Note that usually one uses HMC sample 20,000 sample instead of the 20 presented here. Having many samples also improves the quality of the estimates too.

While sampling from gaussian distributions has its uses, we should also consider sampling from empirical datasets to judge the robustness of each method. This is presented in Table 1. Firstly, we can see that HMC has convergence problems when sampling the Pima Indian dataset.

Dataset	Dimensionality	Relative performance
Pima Indian	8	0.58
Austrailan Credit	14	NA
German Credit	24	2.00
Caravan	86	0.27

Table 1: Relative sampling performance of HMC vs MCMC with MCMC as the baseline on four datasets. [1]

This is likely due to HMC requiring some regularity as it needs the gradient of the probability distribution. As such MCMC is more robust than HMC as it does not require any gradients. As for performance, we can see that both methods are within the same performance class and that one may be faster than the other depending of the structure on the given dataset.

9 Conclusion

Traditional MCMC methods often run in to trouble traversing certain distributions. In particular those defined in higher dimensions. Hamiltonian dynamics help in defining a transition density that both preserves the underlying distribution and traverses the typical set more efficiently. In conclusion, the non-linear differential equations given by Hamiltonian dynamics are a useful tool in creating transition densities to make MCMC better in certain situations. Of key importance to future research is some method to relax the requirement to know the gradient of the probability distribution function like [6] and more insights into the theoretical guarantees of HMC.

References

[1] Adrian Barbu and Song-Chun Zhu. *Monte Carlo Methods*. 2020. DOI: https://doi.org/10.1007/978-981-13-2971-5. URL: https://link.springer.com/book/10.1007/978-981-13-2971-5.

[2] Michael Betancourt. *A Conceptual Introduction to Hamiltonian Monte Carlo*. 2018. URL: https://arxiv.org/abs/1701.02434.

[3] Ziteng Cheng. *Lecture Notes of STA347: Probability University of Toronto*. 2022.

[4] Adam Stinchcombe et el. *APM446 - MAT1508 Lecture Notes*. 2023.

[5] Omidali Aghababaei Jazi. *Lecture Notes of STA447 / 2006: Stochastic Processes University of Toronto*. 2023.

[6] Geoffrey McGregor and Andy T. S. Wan. *Conservative Hamiltonian Monte Carlo*. 2022. URL: https://arxiv.org/abs/2206.06901.

[7] Neal Radford. *MCMC Using Hamiltonian Dynamics*. Ed. by Steve Brooks et al. 2011. DOI: 10.1201/b10905. URL: https://www.mcmchandbook.net/HandbookChapter5.pdf.

[8] C. P. Robert. *The Metropolis-Hastings Algorithm*. 2016.

[9] Alex Rogozhnikov. *Hamiltonian Monte Carlo Explained*. 2016. URL: https://arogozhnikov.github.io/2016/12/19/markov_chain_monte_carlo.html.

[10] *The Metropolis-Hastings Algorithm*. URL: https://www.colorado.edu/amath/sites/default/files/attached-files/2_28_2018.pdf.

[11] Nisheeth K. Vishnoi. *An Introduction to Hamiltonian Monte Carlo Method for Sampling*. 2021. URL: https://arxiv.org/abs/2108.12107.