Sampling from High Dimensional Probability Distributions

A Project Report

Submitted in partial fulfillment of the

requirements for the Degree of

Master of Technology

in

Signal Processing

by

Anand Jerry George

Supervised by

Prof. Navin Kashyap

Electrical Engineering
Indian Institute of Science,
Bangalore, India - 560012

June 2021

Abstract

Drawing samples from high dimensional probability distributions is a ubiquitous problem in statistics and machine learning. This thesis is dedicated to studying two specific high dimensional sampling problems: 1) sampling from probability distributions supported on a lattice, and 2) sampling uniformly from the set of all $d \times d$ doubly stochastic matrices.

The first part of this thesis deals with the problem of drawing samples from probability distributions supported on a d-dimensional lattice $\Lambda = \mathbf{B}\mathbb{Z}^d$, where $\mathbf{B}$ is a full-rank matrix. Specifically, we consider lattice distributions P_Λ in which the probability at a lattice point is proportional to a given probability density function, f, evaluated at that point. To generate samples from P_Λ, it suffices to draw samples from a pull-back measure $P_{\mathbb{Z}^d}$ defined on the integer lattice. The probability of an integer lattice point under $P_{\mathbb{Z}^d}$ is proportional to the density function $\pi = |\det(\mathbf{B})| f \circ \mathbf{B}$. We rely on Markov Chain Monte Carlo (MCMC) methods for drawing samples from $P_{\mathbb{Z}^d}$. Specifically, Metropolis-Hastings and Hamiltonian Monte Carlo are the MCMC algorithms that we use. We classify the algorithms discussed in this part into two: algorithms in which the state-space of the underlying Markov Chain is discrete and algorithms in which the state-space is continuous.

One of the discrete state-space algorithms that we present is an Independent Metropolis-Hastings algorithm. In particular, we use a sample drawn from the density π after rounding to its nearest integer point as the candidate state. We show that this algorithm is uniformly ergodic when $-\log(\pi)$ is gradient Lipschitz. Then we present two more algorithms with discrete state-space that are based on Symmetric Metropolis-Hastings.

One of the continuous state-space algorithms we introduce in this thesis for sampling from $P_{\mathbb{Z}^d}$ is based on the Metropolis-Hastings framework. In particular, we use π as the proposal distribution and calculate the Metropolis-Hastings acceptance ratio for a well-chosen target distribution. We can use any method, denoted by **ALG**, that ideally draws samples from the probability density π, to generate a proposed state. The target

distribution is a piecewise sigmoidal distribution, chosen such that the coordinate-wise rounding of a sample drawn from the target distribution gives a sample from $P_{\mathbb{Z}^d}$. When **ALG** is ideal, we show that our algorithm is uniformly ergodic if $-\log(\pi)$ satisfies a gradient Lipschitz condition. We propose one more continuous state-space algorithm that is based on Hamiltonian Monte Carlo.

In the second part of this thesis, we look into sampling uniformly from the set of all $d \times d$ doubly stochastic matrices, also known as the Birkhoff polytope. We study the well-known Sinkhorn algorithm that converges to a doubly stochastic matrix starting from a positive initial matrix. We develop an alternative to the Sinkhorn algorithm and analyze its convergence. We would like to state that the aim of the second part of the thesis is far from being accomplished.

Acknowledgment

Above all, I thank my family for encouraging me to explore my interests, whatever they are. Especially, I thank my mother, who stood by me in all the ups and downs of my life.

I am heavily indebted to my project advisor, Professor Navin Kashyap, without whom this study would not have been possible. Although we started with a very specific problem, he gave me complete freedom to stroll through the beautiful topic of MCMC. I can undoubtedly say that the ideas and techniques that I learned during this study are much more than documented in this thesis. Meetings with Prof. Navin turned out to be my weekly dose of confidence booster. In addition, he helped me write my first research publication. I thank him for showing infinite patience during all these times.

Another person whom I would like to thank is my teacher at NIT Calicut, Prof. G. Abhilash. He played a significant role in inculcating me with curiosity and love towards mathematics.

Next, I thank everyone who helped me during this project. I thank Prof. Andreas Eberle of the University of Bonn for giving me access to invaluable resources on MCMC. I thank Anaswara S., a past member of the Code Design and Analysis (CDA) lab, for teaching me the lessons she learned during her study on MCMC. I thank all my teachers at IISc, who consistently increased my love for mathematics. Lastly, I thank all my friends for making my stay at IISc so much fun.

Publications based on this Thesis

- A. J. George and N. Kashyap, "An MCMC method to sample from lattice distributions," Accepted to ISIT 2021.

Contents

List of Figures

Nomenclature

$\mathbb{R}$	Real Number
$\mathbb{Z}$	Integer Number
$\mathcal{N}(\mu, \Sigma)$	Gaussian distribution with mean μ and covariance matrix Σ
$\|\mathbf{x}\|$	l^2 - norm of $\mathbf{x}$
DSM	Doubly Stochastic Matrix
HMC	Hamiltonian Monte Carlo
IMH	Independent Metropolis Hastings
MCMC	Monte Carlo Markov Chain
MH	Metropolis Hastings
rv	Random Variable

Part I

MCMC Methods for Sampling from Lattice Distributions

Chapter 1

Introduction

In this part of the thesis, we look into the problem of sampling from probability distributions supported on a lattice (referred to as a *lattice distribution*). Specifically, we consider lattice distributions in which the probability at a lattice point is proportional to a given probability density function evaluated at that point. Such distributions have found applications in cryptography, lattice coding, and secure communication [1], [2], [3]. An example of a lattice distribution that has received much interest from researchers is the lattice Gaussian. A lattice Gaussian distribution is a probability distribution defined on a lattice Λ, such that for each $\mathbf{x} \in \Lambda$, the probability of $\mathbf{x}$ is proportional to a Gaussian density function evaluated at $\mathbf{x}$. Lattice Gaussian sampling, also known as discrete Gaussian sampling (DGS), is closely related to shortest vector problem (SVP) and closest vector problem (CVP), which are computationally hard lattice problems [4]. Lattice Gaussian sampling is also employed for decoding and signal detection in Multiple Input Multiple Output (MIMO) systems [5]. There are some known methods to generate samples from lattice Gaussian distributions [1], [6], [7], but few methods are available for drawing samples from arbitrary lattice distributions [8]. We try to address this problem. The motivation to go beyond lattice Gaussian is in part due to [3], where samples from a particular fat-tailed lattice distribution are used to achieve information-theoretically perfect security.

Sampling methods for generic lattice distributions are not known before this work. All existing algorithms for lattice distribution sampling are tailored for lattice Gaussians to the best of our knowledge. Examples are non-iterative algorithms like Klein's algorithm [9] and Peikert's algorithm [10]. Klein's algorithm samples from a d-dimensional lattice Gaussian distribution by sequentially sampling from d 1-dimensional lattice Gaus-

sians. Klein's algorithm is known to sample from a distribution close to the desired lattice Gaussian when the variance of the lattice Gaussian is not too small. Peikert's algorithm draws samples directly from d-dimensional lattice Gaussians and hence is faster than Klein's algorithm for several lattices. But a sample drawn by Klein's algorithm is from a distribution closer to the desired lattice distribution than Peikert's algorithm. It is worth noting that Klein's algorithm and Peikert's algorithm are randomized versions of Babai's nearest plane and round-off algorithms, respectively [11]. Lately, Markov Chain Monte Carlo (MCMC) methods have been adopted for lattice Gaussian sampling. Some of the known MCMC algorithms for lattice Gaussian sampling are Independent Metropolis-Hastings-Klein (MHK) algorithm, Symmetric Metropolis-Klein (SMK) algorithm [6], Gibbs algorithm, Metropolis within Gibbs algorithm, Learnable delayed Metropolis within Gibbs algorithm [7] etc.

The algorithms that we present in this part take advantage of the fact that there is a related continuous density function for the lattice distributions that we consider. Sampling from continuous probability distributions is more flexible due to the availability of gradient of the density function. The algorithms that we present belong to the MCMC paradigm. In particular, we rely on the Metropolis-Hastings framework to construct our algorithms. The rest of this chapter reviews the background on lattice distributions and the preliminaries of MCMC algorithms.

1.1 Notations

We denote the state space of a Markov chain by $\mathcal{X}$. The operator $\wedge$ operates on two numbers to output the minimum of them. Nearest integer point to a vector $\mathbf{x} \in \mathbb{R}^d$ obtained by coordinate-wise rounding is denoted by $[\mathbf{x}]$. We use $U[0,1]$ to denote the uniform probability distribution on the interval $[0,1]$. We denote the Borel-sigma algebra on $\mathbb{R}^d$ by $\mathcal{B}(\mathbb{R}^d)$. For two probability measures μ and ν defined on the same probability space, we use the notation $\mu \ll \nu$ to indicate that μ is absolutely continuous with respect to ν.

1.2 Lattice Distributions

We now formally define a lattice and probability distributions defined on a lattice. Let $\mathbf{B} \in \mathbb{R}^{d \times d}$ be a full-rank matrix. The d-dimensional lattice Λ generated by $\mathbf{B}$ is defined

as

$$\Lambda \coloneqq \{\mathbf{Bz} : \mathbf{z} \in \mathbb{Z}^d\}.$$

Any probability distribution defined with Λ as the support is known as a lattice distribution. In this thesis, we look at a specific class of lattice distributions in which the probability distribution is induced by a density function on $\mathbb{R}^d$. That is, the probability of a lattice point is equal to the density function evaluated at that point with appropriate normalization. This thesis mainly considers density functions having the following form

$$f(\mathbf{x}) = \frac{e^{-\psi(\mathbf{x})}}{Z_\psi} \quad \text{for all } \mathbf{x} \in \mathbb{R}^d,$$

where $\psi(\mathbf{x})$ is known as a *potential function*, and $Z_\psi = \int_{\mathbb{R}^d} e^{-\psi(\mathbf{x})} d\mathbf{x}$ is a normalization constant. Equivalently, the density function is strictly positive everywhere on $\mathbb{R}^d$. However, in practice, the algorithm that we develop works for any lattice distribution induced by a density function. Let $P_\Lambda(\mathbf{x})$ for $\mathbf{x} \in \Lambda$ be a lattice distribution induced by the above $f(\mathbf{x})$, i.e.,

$$P_\Lambda(\mathbf{x}) = \frac{e^{-\psi(\mathbf{x})}}{Z} \quad \text{for all } \mathbf{x} \in \Lambda,$$

where

$$Z = \sum_{\mathbf{x} \in \Lambda} e^{-\psi(\mathbf{x})}.$$

Let $\mathbf{B}$ denote a generator matrix of the lattice Λ. Then, for generating a sample from the probability distribution P_Λ, it suffices to sample from $P_{\mathbb{Z}^d}(\mathbf{z}) = P_\Lambda(\mathbf{Bz})$, where $\mathbf{z} \in \mathbb{Z}^d$, and then obtain $\mathbf{x}$ as $\mathbf{x} = \mathbf{Bz}$. So, our problem reduces to one of sampling from the following probability distribution over $\mathbb{Z}^d$:

$$P_{\mathbb{Z}^d}(\mathbf{z}) = \frac{e^{-\psi(\mathbf{Bz})}}{Z} \quad \text{for all } \mathbf{z} \in \mathbb{Z}^d.$$

Let $\varphi(\mathbf{x}) \coloneqq \psi(\mathbf{Bx})$ for all $\mathbf{x} \in \mathbb{R}^d$, so that

$$P_{\mathbb{Z}^d}(\mathbf{z}) = \frac{e^{-\varphi(\mathbf{z})}}{Z} \quad \text{for all } \mathbf{z} \in \mathbb{Z}^d. \tag{1.1}$$

Also, let us define the probability density π as:

$$\pi(\mathbf{x}) \coloneqq \frac{e^{-\varphi(\mathbf{x})}}{K} \quad \text{for all } \mathbf{x} \in \mathbb{R}^d, \tag{1.2}$$

where

$$K = \int_{\mathbb{R}^d} e^{-\varphi(\mathbf{x})} d\mathbf{x}.$$

We have now reduced the problem of sampling from a probability distribution defined on an arbitrary lattice Λ to sampling from a probability distribution defined on $\mathbb{Z}^d$. In later chapters of this thesis, we try to develop algorithms for drawing samples from the lattice distribution $P_{\mathbb{Z}^d}$ induced by the probability density π.

Lattice Gaussian Distribution

A lattice distribution that has received significant attention is the lattice Gaussian distribution. A lattice Gaussian distribution defined on a lattice Λ is given by

$$D_{\Lambda,\sigma,\mathbf{c}}(\mathbf{x}) = \frac{e^{-\frac{\|\mathbf{x}-\mathbf{c}\|^2}{2\sigma^2}}}{\sum_{\mathbf{y}\in\Lambda} e^{-\frac{\|\mathbf{y}-\mathbf{c}\|^2}{2\sigma^2}}} \quad \text{for all } \mathbf{x} \in \Lambda, \tag{1.3}$$

where $\mathbf{c} \in \mathbb{R}^d$ is the mean vector and σ is the variance parameter. Lattice Gaussian distributions have important practical applications, particularly in cryptography [1].

1.3 Markov Chain Monte Carlo (MCMC)

As stated earlier, we take the MCMC route to generate samples from the desired lattice distribution. MCMC is a class of sampling algorithms in which an ergodic discrete-time Markov chain (X_t) is set up whose stationary distribution is the same as the desired probability distribution (also called the *target distribution*). The idea is to simulate this Markov chain for a large enough number of steps to draw samples approximately from the desired probability distribution. In this section, we review two specific instances of MCMC algorithms: Metropolis-Hastings and Hamiltonian Monte Carlo.

1.3.1 The Metropolis-Hastings Algorithm

Given a Markov chain, it is straightforward to find its stationary distribution. However, it is not apparent how to find a Markov chain with the desired stationary distribution. The Metropolis-Hastings algorithm provides a recipe for establishing a Markov chain with the desired stationary distribution. Let $\bar{\pi}$ denote the probability distribution from which we want to draw samples. Then, the Metropolis-Hastings algorithm consists of two steps in generating the next state of the Markov chain:

- Let $\mathbf{x}$ be the current state. Generate a proposed state (also called the candidate state) $\mathbf{y}$ from some probability distribution $q(\mathbf{x}, \cdot)$ (referred to as the *proposal distribution*).

- Accept the proposed state as the next state of Markov chain with probability $\alpha(\mathbf{x}, \mathbf{y})$ given by

$$\alpha(\mathbf{x}, \mathbf{y}) = 1 \wedge \frac{\bar{\pi}(\mathbf{y})q(\mathbf{y}, \mathbf{x})}{\bar{\pi}(\mathbf{x})q(\mathbf{x}, \mathbf{y})}.$$

There are two special cases of Metropolis-Hastings algorithm.

Independent Metropolis-Hastings

If the proposed state is independent of the current state, we call this the *Independent Metropolis-Hastings*. The acceptance ratio is then given by

$$\alpha(\mathbf{x}, \mathbf{y}) = 1 \wedge \frac{\bar{\pi}(\mathbf{y})q(\mathbf{x})}{\bar{\pi}(\mathbf{x})q(\mathbf{y})}.$$

From now on, we refer to the Markov chain associated with an Independent Metropolis-Hastings algorithm by IMH Markov chain.

Symmetric Metropolis-Hastings

In Symmetric Metropolis-Hastings, the proposal distribution q satisfies the following property:

$$q(\mathbf{x}, \mathbf{y}) = q(\mathbf{y}, \mathbf{x}) \quad \text{for all } \mathbf{x}, \mathbf{y} \in \mathcal{X}.$$

The acceptance ratio in this case reduces to

$$\alpha(\mathbf{x}, \mathbf{y}) = 1 \wedge \frac{\bar{\pi}(\mathbf{y})}{\bar{\pi}(\mathbf{x})}.$$

1.3.2 Hamiltonian Monte Carlo (HMC)

In this section, we describe an elegant, physics-inspired algorithm to generate samples from continuous densities [12]. Hamiltonian Monte Carlo employs Hamiltonian dynamics to produce the next state of a Markov chain.

Hamiltonian Dynamics

Hamiltonian dynamics is the time evolution of d-dimensional vectors $\mathbf{x}$ and $\mathbf{p}$ according to the partial differential equations

$$\frac{dx_i}{dt} = \frac{\partial H}{\partial p_i},$$
$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial x_i},$$

for $i = 1, \cdots, d$. In this context, $\mathbf{x}$ and $\mathbf{p}$ are called the *position* and the *momentum* variables respectively. The entity H, referred to as *Hamiltonian*, is given by

$$H(\mathbf{x}, \mathbf{p}) = \varphi(\mathbf{x}) + K(\mathbf{p}),$$

where φ is called the *potential energy* and K is called the *kinetic energy*. The $\mathbb{R}^{2d}$ space with elements $(\mathbf{x}, \mathbf{p})$ is called the phase space. Hamiltonian dynamics has three important properties, viz.

- **Reversibility**: Let T_s denote the mapping from $(\mathbf{x}(t), \mathbf{p}(t))$ to $(\mathbf{x}(t+s), \mathbf{p}(t+s))$, that is, the transformation corresponding to running Hamiltonian dynamics for time s. Then negating $\mathbf{p}$, applying T_s, and then negating $\mathbf{p}$ again recovers the initial state from the final state. Therefore T_s is an invertible mapping.
- **Conservation of Hamiltonian**: Hamiltonian $H(\mathbf{x}, \mathbf{p})$ is invariant under Hamiltonian dynamics. This is equivalent to conservation of energy statement.
- **Volume preservation**: Hamiltonian dynamics preserves volume in $(\mathbf{x}, \mathbf{p})$ space. That is, if we apply the mapping T_s to the points in some region R of $(\mathbf{x}, \mathbf{p})$ space with volume V, the image of R under T_s will also have volume V. Equivalently, we can say that Jacobian of T_s is unity.

Using these properties, it can be shown that Hamiltonian dynamics leaves the probability distribution

$$Q(\mathbf{x}, \mathbf{p}) = \frac{1}{Z_H} e^{-\varphi(\mathbf{x})} e^{-K(\mathbf{p})}$$

defined on $\mathbb{R}^{2d}$, invariant (Z_H is the normalization constant). Note that under the distribution Q, the variables $\mathbf{x}$ and $\mathbf{p}$ are statistically independent. The kinetic energy K, is often chosen such that $\mathbf{p}$ has a Gaussian distribution with independent components, i.e., $K(\mathbf{p}) = \frac{\mathbf{p}^T \mathbf{p}}{2\sigma^2}$.

Finding the trajectory of Hamiltonian dynamics in closed form is not always possible, and therefore we have to rely on numerical methods. The leapfrog integrator is the widely used method to simulate the Hamiltonian dynamics and is as described in Algorithm 1.

Hamiltonian Monte Carlo (HMC)

The leapfrog integrator retains the Hamiltonian dynamics' reversibility and volume preservation properties, but it fails to preserve the Hamiltonian H. Thus, a Metropolis-

Algorithm 1: Hamiltonian Dynamics using leapfrog integrator

Input: $\mathbf{x}_0, \mathbf{p}_0, \epsilon, L, \sigma, \nabla\varphi$

Output: $\mathbf{x}_L, \mathbf{p}_L$

for $i = 1 \dots L$ **do**

$\mathbf{p} \leftarrow \mathbf{p}_{i-1} - \frac{\epsilon}{2}\nabla\varphi(\mathbf{x}_{i-1})$;

$\mathbf{x}_i \leftarrow \mathbf{x}_{i-1} + \epsilon\frac{\mathbf{p}}{\sigma^2}$;

$\mathbf{p}_i \leftarrow \mathbf{p} - \frac{\epsilon}{2}\nabla\varphi(\mathbf{x}_i)$;

end

Hastings correction step is required for the dynamics to have Q as the invariant distribution. Also, the Hamiltonian dynamics described so far is deterministic. Randomness is brought into the dynamics by resampling the momentum variables from their probability distribution, after each iteration. This is required to ensure the ergodicity of the Markov chain established. The Hamiltonian dynamics after these modifications is called the HMC algorithm. HMC can also be viewed as a Metropolis-Hastings algorithm which uses Hamiltonian dynamics to generate candidate states. The steps involved in HMC are summarized in Algorithm 2.

1.4 Distance between probability distributions

For assessing the goodness of any sampling algorithm, it is essential to have a metric defined on the space of probability distributions. The metric we use in our analysis is the Total Variation Distance (TVD). For two distributions μ and ν defined on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$, we use $\|\mu - \nu\|_{TV}$ to denote their TVD given by

$$\|\mu - \nu\|_{TV} = \sup_{A \in \mathcal{B}(\mathbb{R}^d)} |\mu(A) - \nu(A)|.$$

If λ is a probability measure such that $\mu \ll \lambda$ and $\nu \ll \lambda$, then an alternate expression for TVD is given by

$$\|\mu - \nu\|_{TV} = \frac{1}{2}\int_{\mathbb{R}^d} \left|\frac{d\mu}{d\lambda}(\mathbf{x}) - \frac{d\nu}{d\lambda}(\mathbf{x})\right| \lambda(d\mathbf{x}), \tag{1.4}$$

where $\frac{d\mu}{d\lambda}$ and $\frac{d\nu}{d\lambda}$ are the Radon-Nikodym derivatives of μ and ν with respect to λ (see Lemma 2.1 in [13]).

Algorithm 2: Hamiltonian Monte Carlo Algorithm

Input: $\varphi, \nabla\varphi, \mathbf{X}_0, \epsilon, L$

Output: Sample from a distribution statistically close to $\pi(\cdot) = \frac{e^{-\varphi(\cdot)}}{K}$

for $t = 1, 2, \ldots$ **do**

- Let $\mathbf{x}$ be the state of $\mathbf{X}_{t-1}$;
- Sample $\mathbf{p}_0$ from $\mathcal{N}(0, \sigma I)$;
- $(\mathbf{y}, \mathbf{p}) \leftarrow$Algorithm 1$(\mathbf{x}, \mathbf{p}_0, \epsilon, L, \sigma, \nabla\varphi)$;
- Calculate acceptance ratio $\alpha = 1 \wedge \exp(\varphi(\mathbf{x}) - \varphi(\mathbf{y}) + \frac{\|\mathbf{p}_0\|^2 - \|p\|^2}{2\sigma^2})$;
- Generate a sample u from $U[0, 1]$;
- **if** $u \leq \alpha$ **then**
  - let $\mathbf{X}_t = \mathbf{y}$;
- **else**
  - $\mathbf{X}_t = \mathbf{x}$;
- **end**
- **if** $t > t_{mix}(\epsilon)$ **then**
  - Output the state of $\mathbf{X}_t$
- **end**

end

1.5 Convergence to stationarity

In this section, we give some definitions useful in evaluating the convergence of a Markov chain to its stationary distribution. We refer the reader to [14], [15] for a comprehensive review of these topics.

Definition 1. A Markov chain with transition kernel P and stationary distribution $\bar{\pi}$ is *uniformly ergodic* if there exists $0 < \delta < 1$ and $M < \infty$ such that for all $\mathbf{x} \in \mathcal{X}$,

$$\|P^t(\mathbf{x}, \cdot) - \bar{\pi}(\cdot)\|_{TV} \leq M(1 - \delta)^t.$$

Theorem 1. *(Theorem 8 in [15]). Let P be the transition kernel of a Markov chain and $\bar{\pi}$ be its stationary distribution. Suppose there exists a $\delta > 0$ and a probability measure ν such that, for all measurable $B \subseteq \mathcal{X}$,*

$$P(\mathbf{x}, B) \geq \delta\nu(B) \quad \textit{for all } \mathbf{x} \in \mathcal{X}.$$

Then the Markov chain with transition kernel P is uniformly ergodic and satisfies the

following inequality:

$$\|P^t(\mathbf{x},\cdot) - \bar{\pi}(\cdot)\|_{TV} \leq (1-\delta)^t \quad \textit{for all } \mathbf{x} \in \mathcal{X}.$$

Definition 2. A Markov chain with transition kernel P and stationary distribution $\bar{\pi}$ is *geometrically ergodic* if there exists $0 < \delta < 1$ such that, for all $\mathbf{x} \in \mathcal{X}$,

$$\|P^t(\mathbf{x},\cdot) - \bar{\pi}(\cdot)\|_{TV} \leq M(\mathbf{x})(1-\delta)^t,$$

with $M(\mathbf{x}) < \infty$.

Definition 3. For a small $\epsilon > 0$, and initial state $\mathbf{x}$, a *mixing time* $t_{\text{mix}}(\epsilon;\mathbf{x})$ is defined as

$$t_{\text{mix}}(\epsilon;\mathbf{x}) = \inf\{t : \|P^t(\mathbf{x},\cdot) - \bar{\pi}(\cdot)\|_{TV} < \epsilon\}.$$

Chapter 2

MCMC on Discrete State-Space

In this chapter, we propose discrete state-space MCMC algorithms for sampling from lattice distributions. As elaborated in the previous chapter, to draw samples from a lattice distribution, it suffices to sample from a distribution $P_{\mathbb{Z}^d}$ on the integer lattice $\mathbb{Z}^d$. We consider $P_{\mathbb{Z}^d}$ having the following form:

$$P_{\mathbb{Z}^d}(\mathbf{z}) = \frac{e^{-\varphi(\mathbf{z})}}{Z} \quad \text{for all } \mathbf{z} \in \mathbb{Z}^d. \tag{2.1}$$

The probability density π is defined as:

$$\pi(\mathbf{x}) \coloneqq \frac{e^{-\varphi(\mathbf{x})}}{K} \quad \text{for all } \mathbf{x} \in \mathbb{R}^d, \tag{2.2}$$

where

$$K = \int_{\mathbb{R}^d} e^{-\varphi(\mathbf{x})} d\mathbf{x}.$$

We propose two algorithms for sampling from $P_{\mathbb{Z}^d}$: an Independent Metropolis-Hastings based algorithm which we call the Independent Metropolis-Hastings with Rounding on $\mathbb{Z}^d$ (IMHRZ), and a Symmetric Metropolis-Hastings algorithm which we call the Random Walk Metropolis with Rounding (RWMR). We prove that IMHRZ is uniformly ergodic under ideal conditions when φ satisfies a particular regularity condition. We also show that RWMR is geometrically ergodic when π is a sub-exponential probability density.

2.1 Independent Metropolis-Hastings with Rounding on $\mathbb{Z}^d$ (IMHRZ)

As mentioned, IMHRZ is based on the Independent Metropolis-Hastings algorithm with $P_{\mathbb{Z}^d}$ as the target distribution. In IMHRZ, we generate a candidate state by drawing

a sample from the probability density π on $\mathbb{R}^d$, and then rounding the sample to its nearest integer point. Any off-the-shelf method like Hamiltonian Monte Carlo (HMC), Metropolis Adjusted Langevin Algorithm (MALA), etc., can be used to draw samples from the probability distribution π. Let us denote the method used to sample from π by **ALG**. If we assume **ALG** produce samples exactly from the probability distribution π, then the value of the proposal distribution q at an integer point $\mathbf{x}$ is the integral of π over the unit hypercube centered at $\mathbf{x}$. That is,

$$q(\mathbf{x}) = \int_{\mathbf{u}\in S} \pi(\mathbf{x}+\mathbf{u})d\mathbf{u}, \tag{2.3}$$

where $S = [-\frac{1}{2}, \frac{1}{2}]^d$. Since the target distribution $P_{\mathbb{Z}^d}$ is proportional to π at every integer lattice point, the acceptance ratio α is calculated as follows:

$$\alpha(\mathbf{x}, \mathbf{y}) = 1 \wedge \frac{\pi(\mathbf{y})q(\mathbf{x})}{\pi(\mathbf{x})q(\mathbf{y})}. \tag{2.4}$$

The steps involved in IMHRZ are described in Algorithm 3.

Algorithm 3: Independent Metropolis-Hastings with Rounding on $\mathbb{Z}^d$

Input: $\mathbf{B}, \mathbf{X}_0, t_{\text{mix}}(\epsilon)$

Output: Sample from a distribution statistically close to P_Λ

for $t = 1, 2, \ldots$ **do**

- Let $\mathbf{x}$ denote the state of $\mathbf{X}_{t-1}$;
- Generate $\mathbf{w}$ from the probability distribution π using **ALG**;
- Round $\mathbf{w}$ to its nearest point in $\mathbb{Z}^d$ to get $\mathbf{y}$;
- Calculate acceptance ratio $\alpha(\mathbf{x}, \mathbf{y})$ given by equation 2.4;
- Generate a sample u from $U[0, 1]$;
- **if** $u \leq \alpha(\mathbf{x}, \mathbf{y})$ **then**
  - let $\mathbf{X}_t = \mathbf{y}$;
- **else**
  - $\mathbf{X}_t = \mathbf{x}$;
- **end**
- **if** $t > t_{mix}(\epsilon)$ **then**
  - Output $\mathbf{B}\mathbf{X}_t$
- **end**

end

2.1.1 Convergence analysis

Now we analyze the convergence speed of IMHRZ algorithm. This analysis assumes that **ALG** is ideal, in the sense that it produce samples exactly from the probability distribution π. Analysis when **ALG** is non-ideal is deferred to Appendix A. The following Proposition is true in general for Independent Metropolis-Hastings algorithm.

Proposition 1. *Let $\mathcal{X}$ be a countable set. A Markov chain established by Independent Metropolis-Hastings algorithm on the state-space $\mathcal{X}$ is uniformly ergodic if*

$$\frac{q(\mathbf{z})}{\nu(\mathbf{z})} \geq \delta > 0, \quad \forall \quad \mathbf{z} \in \mathcal{X}, \tag{2.5}$$

where q is the proposal distribution and ν is the target distribution.

Proof. The transition probability P of an IMH Markov chain with proposal distribution q and target distribution ν is given by

$$P(\mathbf{x},\mathbf{y}) = \begin{cases} q(\mathbf{y}) \wedge \frac{\nu(\mathbf{y})q(\mathbf{x})}{\nu(\mathbf{x})} & ,\text{if } \mathbf{y} \neq \mathbf{x} \\ 1 - \sum_{\mathbf{z}\neq\mathbf{x}} q(\mathbf{z}) \wedge \frac{\nu(\mathbf{z})q(\mathbf{x})}{\nu(\mathbf{x})} & ,\text{if } \mathbf{y} = \mathbf{x}. \end{cases} \quad Then,$$

$$\frac{P(\mathbf{x},\mathbf{y})}{\nu(\mathbf{y})} = \begin{cases} \frac{q(\mathbf{y})}{\nu(\mathbf{y})} \wedge \frac{q(\mathbf{x})}{\nu(\mathbf{x})} & ,\text{if } \mathbf{y} \neq \mathbf{x} \\ \frac{q(\mathbf{y})}{\nu(\mathbf{y})} + \frac{1}{\nu(\mathbf{y})} \sum_{\mathbf{z}\neq\mathbf{x}} \max\left\{0, q(\mathbf{z}) - \frac{\nu(\mathbf{z})q(\mathbf{x})}{\nu(\mathbf{x})}\right\} & ,\text{if } \mathbf{y} = \mathbf{x}. \end{cases}$$

$$\implies P(\mathbf{x},\mathbf{y}) \geq \delta\nu(\mathbf{y}) \quad \text{for all } \mathbf{x},\mathbf{y} \in \mathcal{X}.$$

Therefore by Theorem 1, P is an uniformly ergodic Markov chain. □

Now we will present some target distributions for which (2.5) is satisfied in the case of IMHRZ algorithm. We define a widely used smoothness property of functions called L-smoothness.

Definition 4. A function $f : \mathbb{R}^d \to \mathbb{R}$ is called *L-smooth* if the gradient of f is Lipschitz continuous with parameter L. That is, f should satisfy the following property

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \quad \text{for all } \mathbf{x},\mathbf{y} \in \mathbb{R}^d.$$

The following is a well known fact about L-smooth functions (see Lemma 5 in [16]). If f is L-smooth, then for all $\mathbf{x},\mathbf{y} \in \mathbb{R}^d$,

$$f(\mathbf{y}) \leq f(\mathbf{x}) + (\mathbf{y}-\mathbf{x})^T\nabla f(\mathbf{x}) + \frac{L}{2}\|\mathbf{x}-\mathbf{y}\|^2. \tag{2.6}$$

Proposition 2. *Let $S = [-\frac{1}{2}, \frac{1}{2}]^d$. If φ is an L-smooth function, then for all $\mathbf{z} \in \mathbb{Z}^d$*

$$\frac{q(\mathbf{z})}{P_{\mathbb{Z}^d}(\mathbf{z})} = \frac{Z}{K} \frac{\int_S e^{-\varphi(\mathbf{z}+\mathbf{x})} d\mathbf{x}}{e^{-\varphi(\mathbf{z})}} \geq \frac{Z}{K} \int_S e^{-\frac{L}{2}\|\mathbf{x}\|^2} d\mathbf{x} > 0, \tag{2.7}$$

where q is the proposal distribution of IMHRZ and $P_{\mathbb{Z}^d}$ is the target distribution.

Proof.

$$\begin{aligned}
\frac{q(\mathbf{z})}{P_{\mathbb{Z}^d}(\mathbf{z})} &= \frac{Z}{K} \int_S e^{\varphi(\mathbf{z})-\varphi(\mathbf{z}+\mathbf{x})} d\mathbf{x} \\
&\overset{(a)}{\geq} \frac{Z}{K} \int_S e^{-\mathbf{x}^T \nabla\varphi(\mathbf{z}) - \frac{L}{2}\|\mathbf{x}\|^2} d\mathbf{x} \\
&\overset{(b)}{=} \frac{Z}{K} \frac{1}{2} \int_S e^{-\frac{L}{2}\|\mathbf{x}\|^2} (e^{\mathbf{x}^T \nabla\varphi(\mathbf{z})} + e^{-\mathbf{x}^T \nabla\varphi(\mathbf{z})}) d\mathbf{x} \\
&\overset{(c)}{\geq} \frac{Z}{K} \int_S e^{-\frac{L}{2}\|\mathbf{x}\|^2} d\mathbf{x} > 0,
\end{aligned}$$

where (a) is due to (2.6), (b) is due to the symmetry of S, and (c) is due to AM-GM inequality. □

Therefore, by Proposition 1 and 2 a sufficient condition for IMHRZ to be uniformly ergodic is that φ is an L-smooth function and the transition probability P satisfies the following inequality:

$$\|P^k(\mathbf{z}, \cdot) - P_{\mathbb{Z}^d}(\cdot)\|_{TV} \leq \left(1 - \frac{Z}{K} \int_S e^{-\frac{L}{2}\|\mathbf{u}\|^2} d\mathbf{u}\right)^k \quad \text{for all } \mathbf{z} \in \mathbb{Z}^d.$$

Corollary 1. *IMHRZ algorithm is uniformly ergodic when the target distribution is a lattice Gaussian.*

In Appendix A we analyse the case when **ALG** is not ideal. We show that an upper bound for total variation distance between the distribution at the k^{th} iteration of IMHRZ and target distribution can still be derived. A major drawback of IMHRZ is the practical issue in computing the acceptance ratio α. Computing the value of proposal distribution q requires the evaluation of a high-dimensional integral. The computations required to accurately calculate the value of q at an integer point grow exponentially with dimension. This makes IMHRZ unusable for high dimensions.

2.2 Random Walk Metropolis with Rounding (RWMR)

The computational difficulty in calculating the acceptance ratio in IMHRZ motivates us to look at Symmetric Metropolis-Hastings algorithms. The acceptance ratio of Symmetric Metropolis-Hastings does not have a proposal distribution term, hence easily computable. The only known symmetric proposal algorithm for lattice distribution sampling is the Symmetric Metropolis Klein (SMK) algorithm [6] which is used to sample from lattice Gaussian distribution. This thesis proposes a more straightforward algorithm: Random Walk Metropolis with Rounding (RWMR), which has convergence properties similar to SMK but involves lesser computation. As in the case of SMK, RWMR can be proved to be geometrically ergodic when π is sub-exponential.

In RWMR, a candidate state is obtained by adding to the current state an independent continuous Gaussian random variable after rounding to its nearest integer point. That is, if $\mathbf{x}$ is the current state, the candidate state $\mathbf{y}$ is obtained as

$$\mathbf{y} = \mathbf{x} + [\mathbf{w}],$$

where $\mathbf{w} \sim \mathcal{N}(0, \Sigma)$. Then the proposal distribution q is given by

$$q(\mathbf{x}, \mathbf{y}) = \frac{1}{(2\pi)^{\frac{d}{2}} \det(\Sigma)^{\frac{1}{2}}} \int_S e^{-\frac{1}{2}(\mathbf{y}-\mathbf{x}+\mathbf{u})^T \Sigma^{-1} (\mathbf{y}-\mathbf{x}+\mathbf{u})} d\mathbf{u} = q(\mathbf{y}, \mathbf{x}),$$

where $S = [-\frac{1}{2}, \frac{1}{2}]^d$. The symmetry of the proposal distribution q follows from the symmetry of the Gaussian distribution and the set S. Such a scheme where proposal distribution $q(\mathbf{x}, \mathbf{y})$ is a function of $\mathbf{x}-\mathbf{y}$ is known as a random walk Metropolis algorithm. The procedure to generate one sample from the target distribution is summarized in Algorithm 4. We make it clear that RWMR is a trivial algorithm. We include it in this thesis to inform that there are simple and better symmetric proposal algorithms than SMK. It can be shown that the Markov chain established by Algorithm 4 is geometrically ergodic when π is sub-exponential. However, we do not include the result here, as the proof is almost identical to the proof of Theorem 2 in [6]. Appendix B contains a discussion on the geometric ergodicity of RWMR.

Optimal tuning of parameters

Scaling the variance of proposal distribution is vital in achieving the best possible convergence rate as dimension increases. If the variance is too large, the proposals generated will be far from the current state. Therefore, the probability of the proposed state getting accepted will be less. If the variance is too small, the exploration of the state space

Algorithm 4: Random Walk Metropolis with Rounding Algorithm

Input: $\mathbf{B}, \mathbf{X}_0, t_{\text{mix}}(\epsilon), \mathbf{\Sigma}$

Output: Sample from a distribution statistically close to P_Λ

for $t = 1, 2, \ldots$ **do**

  Let $\mathbf{x}$ denote the state of $\mathbf{X}_{t-1}$;

  Generate $\mathbf{w}$ from $\mathcal{N}(0, \mathbf{\Sigma})$;

  Round $\mathbf{w}$ to its nearest point in $\mathbb{Z}^n$ to get $\mathbf{z}$;

  $\mathbf{y} \leftarrow \mathbf{x} + \mathbf{z}$;

  Calculate acceptance ratio $\alpha(\mathbf{x}, \mathbf{y}) = 1 \wedge \frac{\pi(\mathbf{y})}{\pi(\mathbf{x})}$;

  Generate a sample u from $U[0, 1]$;

  if $u \leq \alpha(\mathbf{x}, \mathbf{y})$ **then**

    let $\mathbf{X}_t = \mathbf{y}$;

  else

    $\mathbf{X}_t = \mathbf{x}$;

  end

  if $t > t_{mix}(\epsilon)$ **then**

    Output $\mathbf{B}\mathbf{X}_t$

  end

end

will be very slow. This is a well-studied problem in random walk Metropolis on continuous state-space [17] [18]. We adopt the same techniques used in random walk Metropolis on continuous state-space for scaling the variance of the proposal distribution. It is well known that the optimum acceptance rate tends to 0.234 as dimension increases for random walk Metropolis. Variance should be scaled as d^{-1} with dimension to maintain this acceptance rate. It is also known that the use of covariance matrix proportional to that of target density for proposal distribution improves the convergence properties of the random walk Metropolis [19]. Overall, the covariance matrix of proposal distribution can be chosen proportional to $d^{-1}\Sigma$, where Σ is the covariance matrix of the target distribution.

Other symmetric Metropolis-Hastings algorithms

In Appendix C, we introduce another symmetric proposal algorithm in which Hamiltonian dynamics is used to generate a candidate state. However, the degradation of

convergence performance of this algorithm when π is a non-isotropic probability distribution makes it of less use in practice.

Chapter 3

MCMC on Continuous State-Space

As we have seen in the previous chapter, the IMHRZ algorithm is not computationally feasible at high dimensions. On the other hand, the RWMR algorithm is computationally efficient. However, analytically deriving its convergence rate is hard. This led us to explore the possibility of MCMC algorithms with continuous state-space. That is, the target distribution, which we denote by $\bar{\pi}$, will now be a probability density on $\mathbb{R}^d$. For this to work, a random variable with the desired probability distribution $P_{\mathbb{Z}^d}$ should be easily derivable from a random variable with distribution $\bar{\pi}$. In this chapter, we show that such choices of $\bar{\pi}$ exist. We again consider $P_{\mathbb{Z}^d}$ having the following form:

$$P_{\mathbb{Z}^d}(\mathbf{z}) = \frac{e^{-\varphi(\mathbf{z})}}{Z} \quad \text{for all } \mathbf{z} \in \mathbb{Z}^d. \tag{3.1}$$

The probability density π is defined as:

$$\pi(\mathbf{x}) \coloneqq \frac{e^{-\varphi(\mathbf{x})}}{K} \quad \text{for all } \mathbf{x} \in \mathbb{R}^d, \tag{3.2}$$

where

$$K = \int_{\mathbb{R}^d} e^{-\varphi(\mathbf{x})} d\mathbf{x}.$$

3.1 Independent Metropolis-Hastings with Rounding (IMHR)

In this section, we introduce an Independent Metropolis-Hastings algorithm for sampling from lattice distribution $P_{\mathbb{Z}^d}$ defined in (3.1). In this algorithm, we suppose that it is

possible to generate samples from the probability density π defined in (3.2). Any state-of-the-art MCMC algorithm such as Hamiltonian Monte Carlo (HMC) [12], or Metropolis adjusted Langevin algorithm (MALA) [20], can be used for this purpose. The idea is to use π as the proposal distribution in the Independent Metropolis-Hastings algorithm. For such a method to be effective in sampling from $P_{\mathbb{Z}^d}$, we need a target distribution $\bar{\pi}$ with the following properties:

- Using a random variable with probability distribution $\bar{\pi}$, we should be able to efficiently derive a random variable with distribution $P_{\mathbb{Z}^d}$.
- The probability distribution $\bar{\pi}$ should be statistically close to π. This will reduce the possibility of rejecting a proposal in the Independent Metropolis-Hastings algorithm, thereby improving its convergence speed to the stationary distribution.

A naive approach would be to choose $\bar{\pi}$ as a piece-wise constant density. That is, $\bar{\pi}(\mathbf{x})$ is equal to $\pi([\mathbf{x}])$ with appropriate normalization for all $\mathbf{x} \in \mathbb{R}^d$. It is easy to see that the rounding operation on a sample generated from $\bar{\pi}$ gives a sample from $P_{\mathbb{Z}^d}$. The drawback of such an approach is that the Markov chain thus generated need not be uniformly ergodic, even for lattice Gaussians (see Appendix D). This motivates us to find a $\bar{\pi}$ that is a better approximation to π.

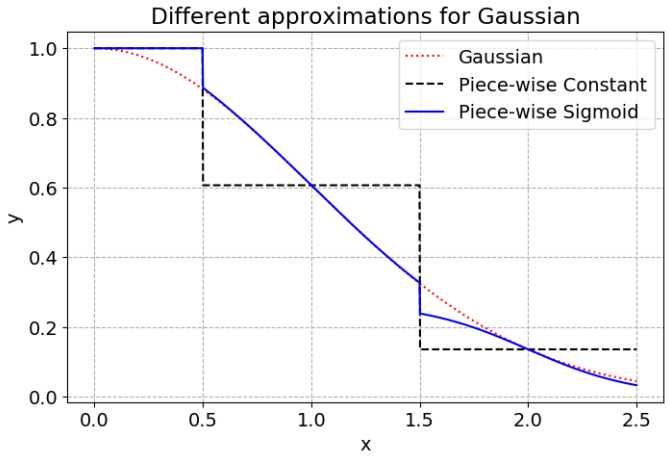

Figure 3.1: Different approximations for Gaussian density

We define a new probability distribution $\bar{\pi}$ which will be called the *target distribution* henceforth, as follows:

$$\bar{\pi}(\mathbf{x}) = \frac{1}{Z} \frac{2e^{-\varphi(\bar{\mathbf{x}})}}{1 + e^{2(\mathbf{x}-\bar{\mathbf{x}})^T \nabla\varphi(\bar{\mathbf{x}})}} \quad \text{for all } \mathbf{x} \in \mathbb{R}^d, \tag{3.3}$$

where Z and φ are the same entities which appear in (3.1), and $\bar{\mathbf{x}}$ is the nearest integer point to $\mathbf{x}$ which is obtained by coordinate-wise rounding. We should visualize $\bar{\pi}$ as a probability density function obtained by approximating π using a sigmoid function within each unit hypercube in $\mathbb{R}^d$ and then normalizing. The sigmoid function is chosen such that, at the center of any unit hypercube, its value and gradient are proportional to the value and gradient of π. This is illustrated in Figure 3.1 where π is a Gaussian density function. Note that the functions plotted in Figure 3.1 are unnormalized. We can obtain a sample from $P_{\mathbb{Z}^d}$ by rounding the sample generated from $\bar{\pi}$ to its nearest integer point. To see this, let $\mathbf{X}$ be a random variable with probability density $\bar{\pi}$. Let $\mathbf{Z} = [\mathbf{X}]$ and let S denote the unit hypercube centered at the origin, i.e., $S = [-\frac{1}{2}, \frac{1}{2}]^d$. Then,

$$\begin{aligned}
\mathbb{P}(\mathbf{Z} = \mathbf{z}) &= \mathbb{P}(\mathbf{X} \in \mathbf{z} + S) \\
&= \int_S \bar{\pi}(\mathbf{z} + \mathbf{u}) d\mathbf{u} \\
&\overset{(a)}{=} \frac{1}{2} \int_S \left(\bar{\pi}(\mathbf{z} + \mathbf{u}) + \bar{\pi}(\mathbf{z} - \mathbf{u})\right) d\mathbf{u} \\
&= \frac{1}{Z} e^{-\varphi(\mathbf{z})} \int_S \left(\frac{1}{1 + e^{2\mathbf{u}^T \nabla\varphi(\mathbf{z})}} + \frac{1}{1 + e^{-2\mathbf{u}^T \nabla\varphi(\mathbf{z})}} \right) d\mathbf{u} \\
&= \frac{1}{Z} e^{-\varphi(\mathbf{z})} \int_S \left(\frac{1}{1 + e^{2\mathbf{u}^T \nabla\varphi(\mathbf{z})}} + \frac{e^{2\mathbf{u}^T \nabla\varphi(\mathbf{z})}}{1 + e^{2\mathbf{u}^T \nabla\varphi(\mathbf{z})}} \right) d\mathbf{u} \\
&= \frac{1}{Z} e^{-\varphi(\mathbf{z})},
\end{aligned}$$

where (a) is due to the symmetry of S. Therefore, to generate samples from $P_{\mathbb{Z}^d}$, it suffices to draw samples from $\bar{\pi}$ and then do coordinate-wise rounding.

Summarizing, IMHR is an Independent Metropolis-Hastings algorithm with π defined in (3.2) as the proposal distribution and $\bar{\pi}$ defined in (3.3) as the target distribution. The steps of IMHR are as described in Algorithm 5.

3.1.1 Convergence analysis

In this section, we analyze the convergence speed of Algorithm 5 to its stationary distribution. Algorithm 5 requires a sub-routine, denoted by **ALG** henceforth, which is ideally capable of drawing samples from π. The following analysis assumes that we have such a sub-routine available. Error due to non-availability of such an ideal sub-routine will be analyzed in the next section.

Algorithm 5: IMHR Algorithm

Input: $\mathbf{X}_0, \pi, \varphi, \nabla\varphi$

Output: Sample from a distribution statistically close to $P_{\mathbb{Z}^d}$

for $t = 1, 2, \ldots$ **do**

Let $\mathbf{x}$ be the state of $\mathbf{X}_{t-1}$;

Generate $\mathbf{y}$ from the probability distribution π;

Round $\mathbf{x}$ to its nearest point in $\mathbb{Z}^d$ to get $\bar{\mathbf{x}}$;

Round $\mathbf{y}$ to its nearest point in $\mathbb{Z}^d$ to get $\bar{\mathbf{y}}$;

$\bar{\pi}(\mathbf{x}) = \frac{2\exp(-\varphi(\bar{\mathbf{x}}))}{1+\exp(2(\mathbf{x}-\bar{\mathbf{x}})^T\nabla\varphi(\bar{\mathbf{x}}))}$;

$\bar{\pi}(\mathbf{y}) = \frac{2\exp(-\varphi(\bar{\mathbf{y}}))}{1+\exp(2(\mathbf{y}-\bar{\mathbf{y}})^T\nabla\varphi(\bar{\mathbf{y}}))}$;

Calculate acceptance ratio $\alpha(\mathbf{x}, \mathbf{y}) = 1 \wedge \frac{\bar{\pi}(\mathbf{y})\pi(\mathbf{x})}{\bar{\pi}(\mathbf{x})\pi(\mathbf{y})}$;

Generate a sample u from $U[0, 1]$;

if $u \leq \alpha(\mathbf{x}, \mathbf{y})$ **then**

let $\mathbf{X}_t = \mathbf{y}$;

else

$\mathbf{X}_t = \mathbf{x}$;

end

if $t > t_{mix}(\epsilon; \mathbf{X}_0)$ **then**

Round $\mathbf{X}_t$ to its nearest point in $\mathbb{Z}^d$ to get $\bar{\mathbf{X}}_t$;

Output $\bar{\mathbf{X}}_t$;

end

end

First, we state a well known theorem which is true in general for an Independent Metropolis-Hastings algorithm.

Theorem 2. *(Theorem 2.1 in [21]) An Independent Metropolis-Hastings algorithm is uniformly ergodic if there exist $\delta > 0$ such that*

$$\frac{\pi(\mathbf{x})}{\bar{\pi}(\mathbf{x})} \geq \delta \quad \textit{for all } \mathbf{x} \in \mathcal{X}, \tag{3.4}$$

where π is the density from which proposed state is generated, and $\bar{\pi}$ is the target density. That is, the transition kernel $\bar{P}$ of the IMH Markov chain satisfies the following:

$$\|\bar{P}^k(\mathbf{x}, \cdot) - \bar{\pi}(\cdot)\|_{TV} \leq (1-\delta)^k \quad \textit{for all } \mathbf{x} \in \mathcal{X}.$$

Now we will give conditions on the probability density π that guarantees uniform ergodicity for Algorithm 5.

Proposition 3. *Let π and $\bar{\pi}$ be as defined in (3.2) and (3.3) respectively. Let $\bar{\mathbf{x}}$ denote the nearest integer point to $\mathbf{x}$. If φ is an L-smooth function, then for all $\mathbf{x} \in \mathbb{R}^d$ we have,*

$$\frac{\pi(\mathbf{x})}{\bar{\pi}(\mathbf{x})} = \frac{Z}{K} \frac{e^{-\varphi(\mathbf{x})}(1 + e^{2(\mathbf{x}-\bar{\mathbf{x}})^T \nabla\varphi(\bar{\mathbf{x}})})}{2e^{-\varphi(\bar{\mathbf{x}})}} \geq \frac{Z}{K} e^{-\frac{dL}{8}} > 0. \tag{3.5}$$

Therefore, by Theorem 2, Algorithm 5 is uniformly ergodic for such a π.

Proof. Let $\mathbf{y} = \mathbf{x} - \bar{\mathbf{x}}$. Then,

$$\begin{aligned}
\frac{\pi(\mathbf{x})}{\bar{\pi}(\mathbf{x})} &= \frac{Z}{K} \frac{e^{\varphi(\bar{\mathbf{x}}) - \varphi(\bar{\mathbf{x}}+\mathbf{y})}(1 + e^{2\mathbf{y}^T \nabla\varphi(\bar{\mathbf{x}})})}{2} \\
&\overset{(a)}{\geq} \frac{Z}{K} \frac{e^{-\mathbf{y}^T \nabla\varphi(\bar{\mathbf{x}}) - \frac{L}{2}\|\mathbf{y}\|^2}(1 + e^{2\mathbf{y}^T \nabla\varphi(\bar{\mathbf{x}})})}{2} \\
&= \frac{Z}{K} \frac{e^{-\frac{L}{2}\|\mathbf{y}\|^2}(e^{\mathbf{y}^T \nabla\varphi(\bar{\mathbf{x}})} + e^{-\mathbf{y}^T \nabla\varphi(\bar{\mathbf{x}})})}{2} \\
&\overset{(b)}{\geq} \frac{Z}{K} e^{-\frac{L}{2}\|\mathbf{y}\|^2} \\
&\overset{(c)}{\geq} \frac{Z}{K} e^{-\frac{dL}{8}} > 0,
\end{aligned}$$

where (a) follows from (2.6) due to L-smoothness of φ, (b) is due to AM-GM inequality and (c) is because $\mathbf{y} \in [-\frac{1}{2}, \frac{1}{2}]^d$. □

Therefore, if π is such that φ is L-smooth, then the transition kernel $\bar{P}$ of Algorithm 5 satisfies the following inequality:

$$\|\bar{P}^k(\mathbf{x}, \cdot) - \bar{\pi}(\cdot)\|_{TV} \leq \left(1 - \frac{Z}{K} e^{-\frac{dL}{8}}\right)^k \quad \text{for all } \mathbf{x} \in \mathbb{R}^d.$$

Corollary 2. *If π is a Gaussian density, then Algorithm 5 produces a uniformly ergodic Markov chain.*

3.1.2 Effect of non-ideality of ALG

As defined in the previous section, **ALG** denotes the method used to draw samples from π in Algorithm 5. For the analysis in this section, we suppose that **ALG** is an MCMC method. In practice, **ALG** could be methods like HMC or MALA. By non-ideality of

ALG, we mean that the TVD between the probability distribution from which **ALG** generate samples and π is nonzero. The non-ideality mentioned above can occur due to the finite time given for convergence in **ALG**. On account of this non-ideality, the proposed state in Algorithm 5 will have a probability distribution different from the one used in the calculation of acceptance ratio (which is π). This alters the stationary distribution of Markov chain associated with Algorithm 5.

Suppose the Markov chain associated with **ALG** is geometrically ergodic for the stationary distribution π. In that case, we show in the following proposition that the error due to non-ideality of **ALG** can be bounded. For a discussion on the conditions under which methods like HMC and MALA are geometrically ergodic, we refer the reader to [22], [20].

Proposition 4. *Let π defined in (3.2) be such that φ is L-smooth. Let $\mathbf{t} \in \mathbb{R}^d$ be a fixed initial state of **ALG**. Suppose **ALG** satisfies the following geometric ergodicity condition.*

$$\|P^n(\mathbf{t}, \cdot) - \pi(\cdot)\|_{TV} \leq V\rho^n, \tag{3.6}$$

*where P is the transition kernel corresponding to **ALG**. (V in the above expression may depend on the fixed initial state $\mathbf{t}$.) Also, let P be such that $\pi \ll P(\mathbf{t}, \cdot)$ and $P(\mathbf{t}, \{\mathbf{t}\}) > 0$. Then for any $\mathbf{x} \in \mathbb{R}^d$, Algorithm 5 generates a Markov chain with transition kernel $\bar{P}$ that satisfies the following inequality:*

$$\|\bar{P}^k(\mathbf{x}, \cdot) - \bar{\pi}(\cdot)\|_{TV} \leq (1 - C\delta)^k + \left(1 + \frac{1}{C\delta}\right)\frac{V\rho^n}{\delta}, \tag{3.7}$$

*where $\bar{\pi}$ is the target distribution defined in (3.3), δ is obtained from (3.4), n is the number of iterations of **ALG**, k is the number of iterations of Algorithm 5, and C is a constant that satisfies the following inequality:*

$$1 - \frac{2V\rho^n}{\delta} \leq C \leq 1 + \frac{2V\rho^n}{\delta}. \tag{3.8}$$

Proof. Let us denote $P^n(\mathbf{t}, \cdot)$ by $q(\cdot)$. By geometric ergodicity of **ALG** we have,

$$\|q - \pi\|_{TV} \leq V\rho^n. \tag{3.9}$$

Although **ALG** generates the proposed state from the distribution q, for calculating the acceptance ratio, we use the distribution π. Hence, the Markov chain generated by Algorithm 5 has the following transition kernel

$$\bar{P}(\mathbf{x}, d\mathbf{y}) \quad = \quad q(d\mathbf{y})\left(1 \wedge \frac{\bar{\pi}(\mathbf{y})\pi(\mathbf{x})}{\bar{\pi}(\mathbf{x})\pi(\mathbf{y})}\right) \; + \; \delta_{\mathbf{x}}(d\mathbf{y})\left(1 - \int_{\mathbb{R}^d} q(d\mathbf{z})\left(1 \wedge \frac{\bar{\pi}(\mathbf{z})\pi(\mathbf{x})}{\bar{\pi}(\mathbf{x})\pi(\mathbf{z})}\right)\right),$$

where $\delta_{\mathbf{x}}(\cdot)$ is the delta measure at $\mathbf{x}$. It is straightforward to verify using the detailed balance equation that the stationary distribution of the above Markov chain with transition kernel $\bar{P}$ is given by

$$\nu(d\mathbf{x}) = \frac{\bar{\pi}(\mathbf{x})q(d\mathbf{x})}{C\pi(\mathbf{x})}, \tag{3.10}$$

where

$$C = \int_{\mathbb{R}^d} \frac{\bar{\pi}(\mathbf{x})q(d\mathbf{x})}{\pi(\mathbf{x})}. \tag{3.11}$$

Also, since φ is L-smooth, from Proposition 3, we have

$$\frac{\pi(\mathbf{x})}{\bar{\pi}(\mathbf{x})} \geq \delta > 0. \tag{3.12}$$

The first step in this proof is to show that the above Markov chain with transition kernel $\bar{P}$ is uniformly ergodic. Then we show that its stationary distribution ν and probability density $\bar{\pi}$ are statistically close. These two results are combined to obtain the result stated in Proposition 4.

Uniform Ergodicity of $\bar{P}$:

From the expression for $\bar{P}$, for all $\mathbf{x} \in \mathbb{R}^d$ and all measurable $A \subseteq \mathbb{R}^d$, we have

$$\begin{aligned}
\bar{P}(\mathbf{x}, A) &\geq \int_A q(d\mathbf{y}) \left(1 \wedge \frac{\bar{\pi}(\mathbf{y})\pi(\mathbf{x})}{\bar{\pi}(\mathbf{x})\pi(\mathbf{y})}\right) \\
&\overset{(a)}{=} \int_A C\nu(d\mathbf{y}) \frac{\pi(\mathbf{y})}{\bar{\pi}(\mathbf{y})} \left(1 \wedge \frac{\bar{\pi}(\mathbf{y})\pi(\mathbf{x})}{\bar{\pi}(\mathbf{x})\pi(\mathbf{y})}\right) \\
&= \int_A C\nu(d\mathbf{y}) \left(\frac{\pi(\mathbf{y})}{\bar{\pi}(\mathbf{y})} \wedge \frac{\pi(\mathbf{x})}{\bar{\pi}(\mathbf{x})}\right) \\
&\overset{(b)}{\geq} C\delta \int_A \nu(d\mathbf{y}) = C\delta\nu(A),
\end{aligned}$$

where (a) and (b) are due to (3.10) and (3.12) respectively. Thus by Theorem **??**, $\bar{P}$ is the transition kernel of a uniformly ergodic Markov chain. Therefore,

$$\|\bar{P}^k(\mathbf{t}, \cdot) - \nu(\cdot)\|_{TV} \leq (1 - C\delta)^k. \tag{3.13}$$

Next, we find a bound on the value of C. Note that from the assumption $P(\mathbf{t}, \{\mathbf{t}\}) > 0$, it follows that for any measurable set $A \subseteq \mathbb{R}^d$ and integer n, $P^n(\mathbf{t}, A) > 0$ whenever $P(\mathbf{t}, A) > 0$. Therefore, $P(\mathbf{t}, \cdot) \ll q$. This, together with the assumption $\pi \ll P(\mathbf{t}, \cdot)$,

allows us to conclude that $\pi \ll q$. Therefore, we have the following:

$$\begin{aligned} 1 = \int_{\mathbb{R}^d} \bar{\pi}(d\mathbf{x}) &\overset{(a)}{=} \int_{\mathbb{R}^d} \frac{\bar{\pi}(\mathbf{x})}{\pi(\mathbf{x})} \pi(d\mathbf{x}) \\ &\overset{(b)}{=} \int_{\mathbb{R}^d} \frac{\bar{\pi}(\mathbf{x})}{\pi(\mathbf{x})} \frac{d\pi}{dq}(\mathbf{x}) q(d\mathbf{x}), \end{aligned} \tag{3.14}$$

where (a) follows from the fact that π and $\bar{\pi}$ are density functions which are positive everywhere and (b) is due to the absolute continuity of π with respect to q. Then,

$$\begin{aligned} |C-1| &\overset{(a)}{=} \left| \int_{\mathbb{R}^d} \frac{\bar{\pi}(\mathbf{x})}{\pi(\mathbf{x})} q(d\mathbf{x}) - \int_{\mathbb{R}^d} \frac{\bar{\pi}(\mathbf{x})}{\pi(\mathbf{x})} \frac{d\pi}{dq}(\mathbf{x}) q(d\mathbf{x}) \right| \\ &= \left| \int_{\mathbb{R}^d} \frac{\bar{\pi}(\mathbf{x})}{\pi(\mathbf{x})} \left(1 - \frac{d\pi}{dq}(\mathbf{x}) \right) q(d\mathbf{x}) \right| \\ &\leq \int_{\mathbb{R}^d} \frac{\bar{\pi}(\mathbf{x})}{\pi(\mathbf{x})} \left| 1 - \frac{d\pi}{dq}(\mathbf{x}) \right| q(d\mathbf{x}) \\ &\leq \frac{1}{\delta} \int_{\mathbb{R}^d} \left| 1 - \frac{d\pi}{dq}(\mathbf{x}) \right| q(d\mathbf{x}) \\ &\overset{(b)}{=} \frac{2}{\delta} \|q - \pi\|_{TV} \\ &\leq \frac{2V\rho^n}{\delta}, \end{aligned}$$

where (a) follows from (3.11) and (3.14), and (b) is due to the alternate definition of TVD given in (1.4). Therefore, we have

$$\begin{aligned} \delta - 2V\rho^n \leq C\delta \leq \delta + 2V\rho^n \\ \implies 1 - \frac{2V\rho^n}{\delta} \leq C \leq 1 + \frac{2V\rho^n}{\delta}. \end{aligned} \tag{3.15}$$

TVD between ν and $\bar{\pi}$:

Now we will show that ν and $\bar{\pi}$ are statistically close probability distributions.

$$\begin{aligned}
\|\nu - \bar{\pi}\|_{TV} &\overset{(a)}{=} \frac{1}{2}\int_{\mathbb{R}^d} \left|\frac{d\nu}{dq}(\mathbf{x}) - \frac{d\bar{\pi}}{dq}(\mathbf{x})\right| q(d\mathbf{x}) \\
&= \frac{1}{2}\int_{\mathbb{R}^d} \left|\frac{\bar{\pi}(\mathbf{x})}{C\pi(\mathbf{x})} - \frac{\bar{\pi}(\mathbf{x})}{\pi(\mathbf{x})}\frac{d\pi}{dq}(\mathbf{x})\right| q(d\mathbf{x}) \\
&= \frac{1}{2C}\int_{\mathbb{R}^d} \frac{\bar{\pi}(\mathbf{x})}{\pi(\mathbf{x})}\left|1 - C\frac{d\pi}{dq}(\mathbf{x})\right| q(d\mathbf{x}) \\
&\leq \frac{1}{2C\delta}\int_{\mathbb{R}^d} \left|1 - C\frac{d\pi}{dq}(\mathbf{x})\right| q(d\mathbf{x}) \\
&\leq \frac{1}{2C\delta}\int_{\mathbb{R}^d} \left(C\left|1 - \frac{d\pi}{dq}(\mathbf{x})\right| + |C-1|\right) q(d\mathbf{x}) \\
&= \frac{1}{\delta}\|q - \pi\|_{TV} + \frac{|C-1|}{2C\delta} \\
&\overset{(b)}{\leq} \frac{V\rho^n}{\delta} + \frac{V\rho^n}{C\delta^2},
\end{aligned} \tag{3.16}$$

where (a) is due to the alternate definition of TVD given in (1.4), and (b) is due to (3.9) and (3.15).

Finally using triangle inequality, we have

$$\begin{aligned}
\|\bar{P}^k(\mathbf{t},\cdot) - \bar{\pi}(\cdot)\|_{TV} &\leq \|\bar{P}^k(\mathbf{t},\cdot) - \nu(\cdot)\|_{TV} + \|\nu - \bar{\pi}\|_{TV} \\
&\leq (1 - C\delta)^k + (1 + \frac{1}{C\delta})\frac{V\rho^n}{\delta}.
\end{aligned} \tag{3.17}$$

□

3.1.3 Simulation Results

This section illustrates the speed of convergence of Algorithm 5 to $P_{\mathbb{Z}^d}$. For this, ideally we would like to show plots of TVD as a function of the number of iterations. However, evaluating distance between high dimensional probability distributions is computationally hard. So, in our simulations, we compute an entity TVD_m instead of TVD. We compute TVD_m as follows: Initialize Algorithm 5 with a fixed point in the state space. Then we run t iterations of Algorithm 5. Repeat this 100,000 times for each value of t. For each t, use these samples to form d 1-dimensional histograms, one for each coordinate. We call the distributions obtained by normalizing the histograms as the empirical marginal distributions, denoted by $h^i(\mathbf{z})$ for $1 \leq i \leq d$ and $\mathbf{z} \in \mathbb{Z}$. We denote the i^{th} marginal distribution of $P_{\mathbb{Z}^d}$ by $P^i_{\mathbb{Z}}$. If $P^i_{\mathbb{Z}}$ is not available in closed form, we estimate it

using RWRM algorithm, with sufficient time given for convergence. Calculate the Total Variation Distance between h^i and $P^i_{\mathbb{Z}}$ using the following formula:

$$\text{TVD}(i) = \frac{1}{2}\sum_{\mathbf{z}\in\mathbb{Z}} |h^i(\mathbf{z}) - P^i_{\mathbb{Z}}(\mathbf{z})| \quad \text{for } 1 \le i \le d.$$

Finally, TVD_m is the maximum of TVD's calculated for marginal distributions.

$$\text{TVD}_\text{m} = \max\{\text{TVD}(i) : 1 \le i \le d\}.$$

We plot TVD_m for different values of t. The TVD_m vs. t plots depicts the number of iterations required for Algorithm 5 to converge to its stationary distribution. However, we would like to assert that convergence inTVD_m is only a heuristic; it does not, in general, imply convergence in TVD.

In another simulation, we plot the autocorrelation function (ACF) of the time series $\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_N$ obtained using Algorithm 5. In many instances, autocorrelation plots have been used to assess the number of iterations of the Markov chain required to produce two almost independent samples [19]. The definition of the autocorrelation function that we use is as follows:

$$\text{ACF}(\tau) = \frac{\sum_{t=1}^{N-\tau} \mathbf{x}_t^T \mathbf{x}_{t+\tau}}{\sum_{t=1}^{N} \mathbf{x}_t^T \mathbf{x}_t},$$

where $\mathbf{x}_t$ is the state of the Markov chain at iteration t and N is the total number of samples in the time series. Now we present results of these simulations for different probability densities π.

Isotropic Gaussian distribution

We first consider the case when π is an isotropic Gaussian density. The potential function φ of an isotropic Gaussian density is $\frac{1}{\sigma^2}$-smooth, where σ^2 is the variance of the Gaussian. Therefore, from Proposition 3, we see that the factor that governs the rate of convergence of Algorithm 5 is $r = \frac{d}{\sigma^2}$. In this simulation, we fix $\sigma^2 = 1$ and vary dimension d to get different values of r. We use state $\mathbf{0}$ as the initial state of the algorithm. TVD_m vs. t for different values of r is shown in Figure 3.2. Autocorrelation vs. τ is shown in Figure 3.3. The number of samples (N) used to calculate the autocorrelation function is 10,000.

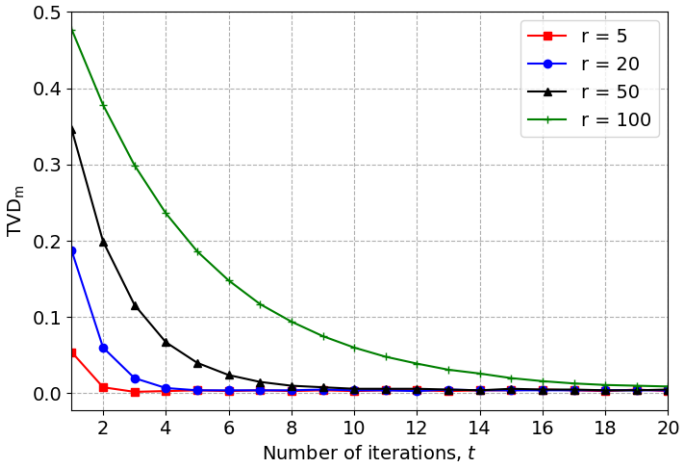

Figure 3.2: TVD_m vs. Number of Iterations (t) for isotropic Gaussian

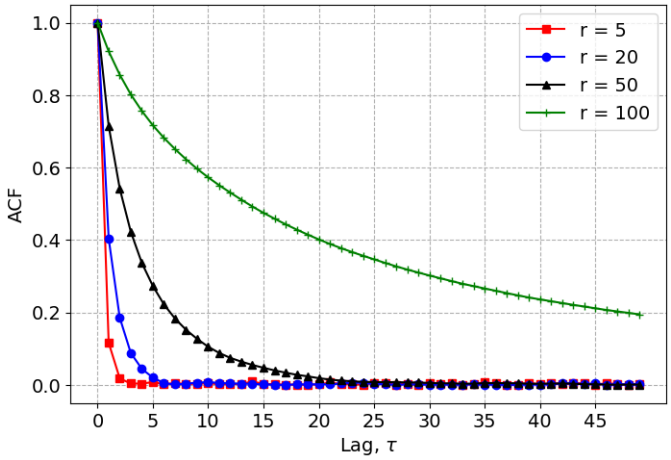

Figure 3.3: ACF vs τ for isotropic Gaussian

Gaussian distribution on the Leech lattice

Next, we consider a lattice Gaussian distribution supported on the Leech lattice with dimension equal to 24 (see pg.133 in [23] for the generator matrix of the Leech lattice). This simulation illustrates the performance of Algorithm 5 for non-isotropic Gaussian. The Leech lattice induces a highly skewed lattice Gaussian distribution on $\mathbb{Z}^d$. The density π now takes the following form:

$$\pi(\mathbf{x}) = Me^{-\frac{\|\mathbf{Bx}\|^2}{2\sigma^2}} \quad \text{for all } \mathbf{x} \in \mathbb{R}^d,$$

where $\mathbf{B}$ is the generator matrix of the Leech lattice and M is the normalization constant. We plot TVD_m vs. t for different values of σ^2. This is shown in the Figure 3.4. State $\mathbf{0}$ was used as the initial state of the algorithm.

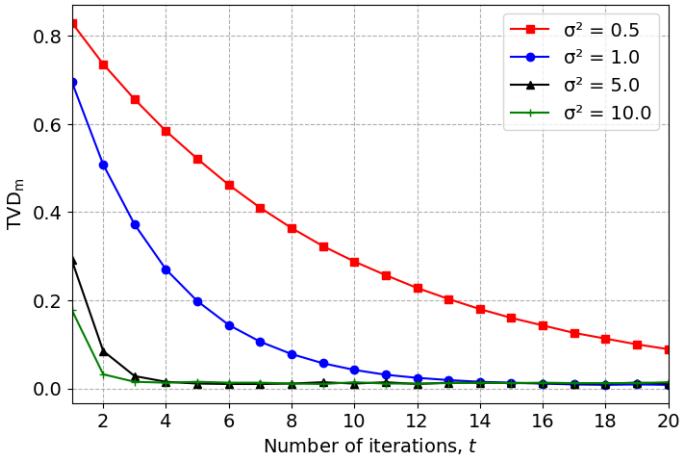

Figure 3.4: TVD_m vs. Number of Iterations (t) for a Gaussian distribution on the Leech lattice

Perfect Security distribution

Finally, we present the simulation results when π is the following probability density which was used to achieve *perfect security* in [3]. The probability density function of a "Perfect Security distribution" is given by:

$$\pi(\mathbf{x}) = M \left(\frac{\Omega_d(\frac{\|\mathbf{x}\|}{2\rho})}{j^2_{\frac{d-2}{2}} - \frac{\|\mathbf{x}\|^2}{4\rho^2}} \right)^2 \quad \text{for all } \mathbf{x} \in \mathbb{R}^d,$$

where

$$\Omega_d(u) = \left(\frac{2}{u}\right)^{\frac{d-2}{2}} J_{\frac{d-2}{2}}(u),$$

$J_k(u)$ is the Bessel function of k^{th} order and j_k is the first zero of k^{th} order Bessel function and M is the normalization constant. Note that here, π is not positive everywhere on $\mathbb{R}^d$. The theoretical guarantees that we derived for Algorithm 5 are not applicable for this distribution. Let $\mathbf{X} = [X_1, X_2, \cdots, X_d]$ be a random vector with perfect security probability distribution. Then, the variance of each component X_i is given by the

following equation [24]:

$$\mathrm{Var}(X_i) = \frac{4\rho^2 j^2_{\frac{d-2}{2}}}{d}.$$

We use HMC to sample from this continuous density π. We plot $\mathrm{TVD_m}$ vs. t for different values of d. We fix the value of ρ such that the variance of the distribution is 1 for each value of d. The $\mathrm{TVD_m}$ vs. t plot is shown in Figure 3.5. State $\mathbf{0}$ was used as the initial state of the algorithm.

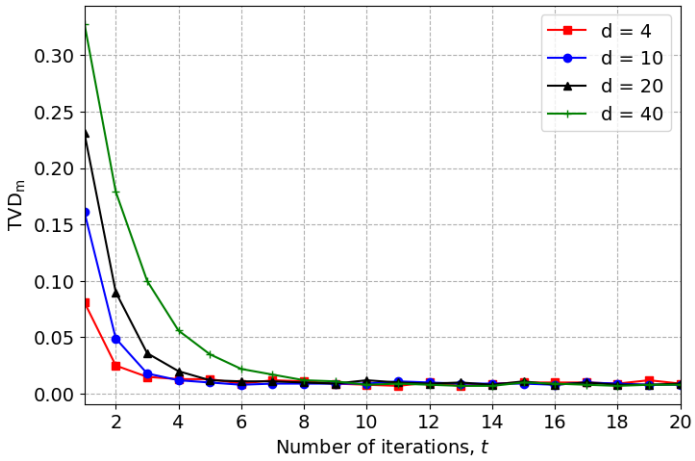

Figure 3.5: $\mathrm{TVD_m}$ vs. Number of Iterations (t) for Perfect Security distribution

3.2 Continuous Relaxation based Hamiltonian Monte Carlo (CRHMC)

A drawback of Algorithm 5 is that we need a sub-routine **ALG** when sampling from lattice distributions beyond lattice Gaussians. As already mentioned, **ALG** itself can be an MCMC method. For Algorithm 5 to produce samples from the target distribution, we must run a sufficient number of iterations of **ALG** within Algorithm 5. This introduces an extra parameter that needs to be chosen, which is the number of iterations of **ALG**. In this section, we introduce a reduction of the overall algorithm when **ALG** is HMC, by which we can avoid the extra parameter mentioned above.

In CRHMC, the idea is to use Hamiltonian dynamics with φ defined as in (3.2) to generate a proposed state. The Metropolis-Hastings acceptance step is done with $\bar{\pi}$ defined in (3.3) as the target distribution. As in HMC, to ensure ergodicity, we need

to resample momentum variables after each iteration. Algorithm 6 describes the steps involved in CRHMC.

Algorithm 6: CRHMC

Input: $\mathbf{X}_0, \varphi, \nabla\varphi, \sigma, \epsilon, L$

Output: A sample approximately from $P_{\mathbb{Z}^d}$

for $t = 1, 2, \ldots$ **do**

Let $\mathbf{x}$ be the state of $\mathbf{X}_{t-1}$;

Sample $\mathbf{p}_0$ from $\mathcal{N}(0, \sigma^2 I_d)$;

$(\mathbf{y}, \mathbf{p}) \leftarrow$Algorithm 1$(\mathbf{x}, \mathbf{p}_0, \epsilon, L, \sigma, \nabla\varphi)$;

$\bar{\mathbf{x}} \leftarrow [\mathbf{x}]$; $\bar{\mathbf{y}} \leftarrow [\mathbf{y}]$;

$\bar{\pi}(\mathbf{x}) = \frac{2\exp(-\varphi(\bar{\mathbf{x}}))}{1+\exp(2(\mathbf{x}-\bar{\mathbf{x}})^T\nabla\varphi(\bar{\mathbf{x}}))}$;

$\bar{\pi}(\mathbf{y}) = \frac{2\exp(-\varphi(\bar{\mathbf{y}}))}{1+\exp(2(\mathbf{y}-\bar{\mathbf{y}})^T\nabla\varphi(\bar{\mathbf{y}}))}$;

$\alpha((\mathbf{x}, \mathbf{p}_0), (\mathbf{y}, \mathbf{p})) = 1 \wedge \frac{\bar{\pi}(\mathbf{y})}{\bar{\pi}(\mathbf{x})} \exp\left(\frac{\|\mathbf{p}_0\|^2 - \|\mathbf{p}\|^2}{2\sigma^2}\right)$;

Generate a sample u from $U[0, 1]$;

if $u \leq \alpha((\mathbf{x}, \mathbf{p}_0), (\mathbf{y}, \mathbf{p}))$ **then**

$\mathbf{X}_t \leftarrow \mathbf{y}$;

else

$\mathbf{X}_t \leftarrow \mathbf{x}$;

end

if $t > t_{mix}(\epsilon; \mathbf{X}_0)$ **then**

Output $[\mathbf{X}_t]$

end

end

In Algorithm 6, if we use $-\log(\bar{\pi})$ as the potential energy in Hamiltonian dynamics, it is precisely the HMC algorithm. However, since $-\log(\bar{\pi})$ is a discontinuous function, it is not feasible to run Hamiltonian dynamics with $-\log(\bar{\pi})$ as the potential energy. The probability density π can be seen as a continuous relaxation of $\bar{\pi}$. So, in Algorithm 6 we use $-\log(\pi)$ as the potential energy in the Hamiltonian dynamics and hence the name Continuous Relaxation based HMC (CRHMC). The TVD_m vs. t plot of CRHMC for sampling from Perfect Security distribution shown in Figure 3.6.

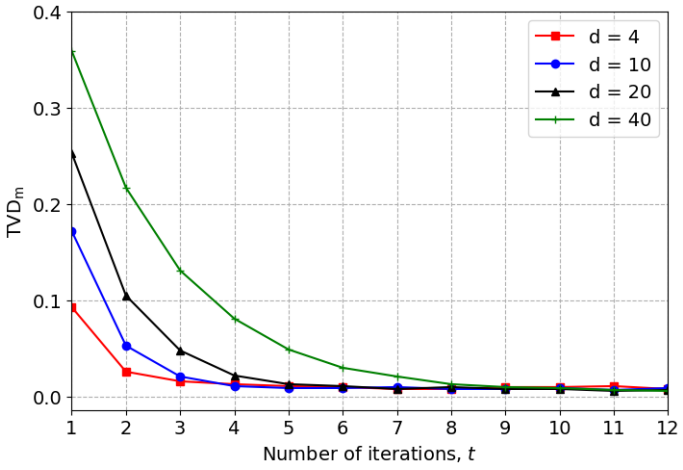

Figure 3.6: TVD_m vs. Number of Iterations (t) for Perfect Security distribution

3.3 Discussion

We presented a simple MCMC algorithm to draw samples from lattice distributions. As demonstrated through the Perfect Security distribution sampling, Algorithm 5 can sample from distributions beyond lattice Gaussians. To the best of our knowledge, prior to this work, there were no efficient algorithms known to generate samples from lattice distributions other than lattice Gaussians. The main feature of Algorithm 5, which makes it competitive even among the lattice Gaussian sampling algorithms, is its computational efficiency. The computations in Algorithm 5 are vector operations, which is highly optimized when current linear algebra libraries (for instance, OpenBLAS or Intel MKL) are used for implementation. Most of the algorithms currently available for sampling from a lattice Gaussian do coordinate-wise sequential sampling using 1-dimensional lattice Gaussian samplers. This method is inefficient when the lattice dimension under consideration is large.

A popular algorithm for sampling from lattice Gaussians is Klein's algorithm [9], [1]. In Figure 3.7, we compare the run-time required per sample generated for Klein's algorithm and Algorithm 5 for different values of dimension when the desired distribution is a lattice Gaussian on $\mathbb{Z}^d$ with variance parameter $\sigma = 1$. This experiment was run in a python environment on a machine with Intel i7-6700 @ 3.40GHz CPU and 8GB RAM. In simulations, we observe that the scaling of the run-time per iteration with dimension is much better for Algorithm 5. However, multiple iterations of Algorithm 5 are required to generate a sample approximately from the stationary distribution, whereas

Klein's algorithm is not iterative. This gives rise to comparable run-time per sample for Klein's algorithm and Algorithm 5, as shown in Figure 3.7. The number of iterations of Algorithm 5 used was the minimum required to bring the TVD_{m} below 0.005.

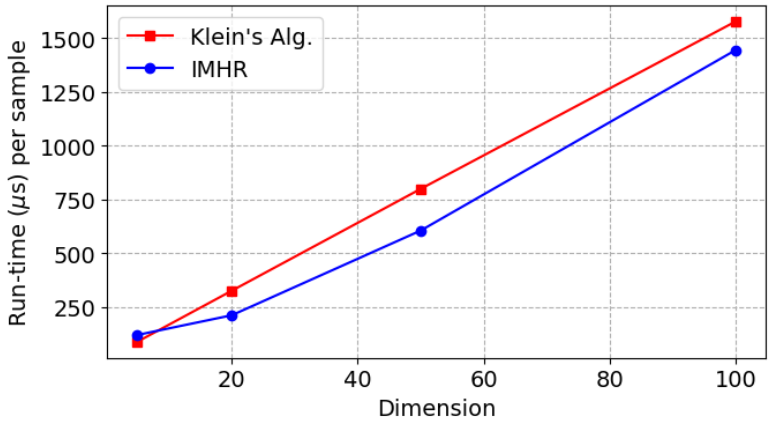

Figure 3.7: Run-time per sample vs Dimension for isotropic lattice Gaussian

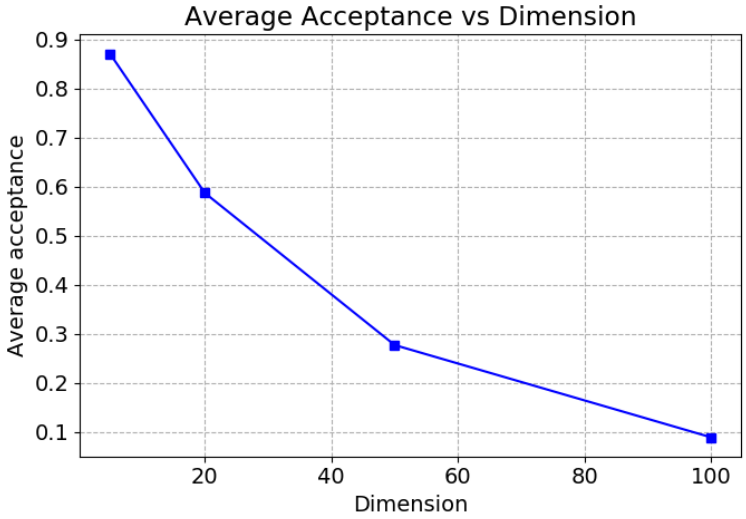

Figure 3.8: Average Acceptance vs Dimension for isotropic lattice Gaussian

From Proposition 3, we see that when π is an isotropic Gaussian density with variance equal to σ^2, the TVD between the probability distribution after k^{th} iteration of Algorithm 5 and the stationary distribution is upper bounded by $(1-\frac{Z}{K}e^{-\frac{d}{8\sigma^2}})^k$. This indicates that our algorithm may not be well suited for distributions with very low variance

and high dimension. Figures 3.2 and 3.4 validate this by illustrating that convergence is slow for low variance and high dimension cases. In simulations, we observe that at high dimensions, the average acceptance, which is the fraction of iterations in which the proposed state is accepted, becomes very low for Algorithm 5. Figure 3.8 shows the degradation of average acceptance with dimension. A low acceptance ratio makes the Independent Metropolis-Hastings algorithm inefficient due to frequent rejection of the proposed state. Therefore, at very high dimensions, we suggest using the Metropolis-within-Gibbs strategy [25]. In the Metropolis-within-Gibbs algorithm, the number of variables updated at a time, determines the average acceptance.

Chapter 4

Conclusion

We have discussed multiple methods for drawing samples from lattice distributions. First, we presented IMHRZ, an Independent Metropolis-Hastings algorithm. IMHRZ can be proved to be uniformly ergodic under certain conditions on the target distribution. However, implementing IMHRZ required evaluating an integral whose dimension is the same as the lattice distribution dimension. This made IMHRZ practically infeasible at high dimensions. Then we looked at RWRM, which is a Symmetric Metropolis-Hastings algorithm. RWRM is a simple algorithm and is geometrically ergodic for sub-exponential target distributions. Then we moved on to algorithms with continuous state-space. We introduced the IMHR algorithm, which is uniformly ergodic for a class of target distributions. We also introduced an algorithm called CRHMC. Out of these, IMHR and RWRM are computationally most efficient. For lattice Gaussians, the IMHR algorithm converges the fastest. Figure 4.1 shows the comparison of $\mathrm{TVD}_{\mathrm{m}}$ for IMHR and RWRM algorithms. In the simulation, distribution π is the unit variance Isotropic Gaussian distribution with dimension equal to 20.

The IMHR algorithm works well in practice, even for distributions like Perfect Security distribution, although there are no theoretical guarantees. Further, the CRHMC method can be helpful for sampling from lattice distributions beyond lattice Gaussians. CRHMC shows performance similar to the IMHR algorithm. A disadvantage of all the methods discussed in this part is that convergence is slow for low variance and high dimension cases.

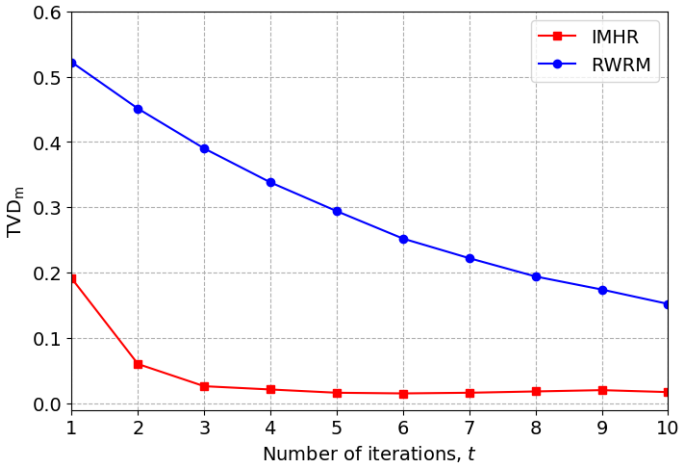

Figure 4.1: TVD_m vs. Number of Iterations (t) for Isotropic Gaussian distribution

4.1 Future Work

One of the open questions is if it is possible to derive an accurate estimate for the mixing time of the IMHR algorithm as a function of d and L? Also, the guarantees of uniform ergodicity are given only when exact samples can be drawn from the probability distribution π. It is an open question if it is possible to find other MCMC algorithms for which exponential convergence can be proved without such assumptions? Deriving the convergence rate of CRHMC is another work that can be taken up in the future. However, it is a technical challenge to analyze algorithms that are based on Hamiltonian Monte Carlo.

Part II

Uniform Sampling from Birkhoff Polytope

Chapter 5

An Alternative to the Sinkhorn Algorithm

In this part, we examine the problem of sampling uniformly from the set of all $d \times d$ doubly stochastic matrices. A doubly stochastic matrix (DSM) is a non-negative matrix in which the row sums and column sums are equal to 1. We denote the set of all $d \times d$ DSMs by $\mathcal{M}_d$. The set $\mathcal{M}_d$ is known as the Birkhoff polytope. A well-known theorem by Birkhoff states that $\mathcal{M}_d$ is the convex hull of the set of all $d \times d$ permutation matrices. In operations research, the following is known as the assignment problem: Find the permutation σ that minimizes $\sum_i C_{i\sigma(i)}$ for a cost matrix C_{ij}. This is equivalent to minimizing $\sum_{i,j} C_{ij}M_{ij}$ for $M \in \mathcal{M}_d$. Thus, the Birkhoff polytope turns out to be the feasible set for the assignment problem [26].

Any Markov chain has a uniform stationary distribution if and only if its transition probability matrix is doubly stochastic. Thus, sampling uniformly from Birkhoff polytope helps study typical Markov chains with uniform stationary distribution [26]. Furthermore, such an algorithm can also be useful in sampling contingency tables and developing randomized algorithms for convex optimization problems [27].

MCMC methods for sampling from convex, compact sets in $\mathbb{R}^d$ have been studied extensively in the literature. Ball walk, Hit and run algorithm, Riemannian Hamiltonian Monte Carlo, etc., are some examples of algorithms used to sample from compact, convex bodies [27]. A simple method specialized for sampling from Birkhoff polytope is a Gibbs sampling algorithm [28]. In this algorithm, in each step, two columns and two rows are selected randomly. The elements at the intersection of these rows and columns are then sampled from their conditional distribution.

We take a different approach for sampling uniformly from the Birkhoff polytope. In this work, we study iterative algorithms that converge to a DSM starting from a positive matrix. A well-known algorithm that achieves this is the Sinkhorn algorithm [29], [30]. If the entries of the initial matrix are randomly sampled, this will induce a probability distribution on the DSM that we obtain by running the algorithm. We aim to study and develop algorithms that converge to a DSM starting from a positive initial matrix. We also would like to identify the distribution from which the entries of the initial matrix are to be sampled so that such an algorithm will converge to a DSM whose distribution is uniform on the Birkhoff polytope.

5.1 Sinkhorn Algorithm

Let $\mathcal{D}(\cdot)$ denote a diagonal matrix with its argument as the diagonal. Given a non-negative matrix A, does there exist a DSM B of the form $\mathcal{D}(r)A\mathcal{D}(c)$ for some vectors r and c ? Let $A = (a_{ij})$ be a $d \times d$ matrix and σ be a permutation of $\{1, 2, \cdots, d\}$. Then, a diagonal of the matrix A is defined as the sequence of elements $a_{1\sigma(1)}, a_{2\sigma(2)}, \cdots, a_{d\sigma(d)}$. Then, the necessary and sufficient condition for the existence of positive vectors r and c such that $B = \mathcal{D}(r)A\mathcal{D}(c)$ is a DSM is that $A \neq 0$ and every positive element of A lies on a positive diagonal [29]. The DSM B is unique if it exists. The matrix A is fully indecomposable is if there do not exist permutation matrices P and Q such that

$$PAQ = \begin{bmatrix} B & C \\ 0 & D \end{bmatrix},$$

where B and D are square matrices. Vectors r and c are unique upto a scalar multiple if and only if A is fully indecomposable.

When the initial matrix A satisfies certain conditions, the Sinkhorn algorithm produces a sequence of matrices that converges to a DSM of the form $\mathcal{D}(r)A\mathcal{D}(c)$. Let A be a non-negative initial matrix. Then in the Sinkhorn algorithm, a matrix A_1 is obtained by normalizing the rows of matrix A by their sums. Matrix A_2 is then obtained by normalizing the columns of A_1 by their sums. This process is continued to generate a sequence of matrices $\{A_n\}$. The sequence A_n converges to a doubly stochastic matrix if and only if the initial matrix A has a positive diagonal. The Sinkhorn algorithm converges to B if there exists DSM B of the form $\mathcal{D}(r)A\mathcal{D}(c)$.

An equivalent way to conceive Sinkhorn algorithm is to look at the fixed point equation. Suppose there exists positive vectors r and c such that $B = \mathcal{D}(r)A\mathcal{D}(c)$ is a DSM.

Then we have $Be = e$ and $B^T e = e$, where e is the all one's vector. This gives us the fixed point equations $r = \frac{1}{Ac}$ and $c = \frac{1}{A^T r}$, where $\frac{1}{x}$ denotes the element-wise inverse of vector x. This naturally gives us the following iteration equations:

$$c_{k+1} = \frac{1}{A^T r_k},$$
$$r_{k+1} = \frac{1}{A c_{k+1}}.$$

The convergence of the Sinkhorn algorithm was proved using a variety of methods. A survey of various methods are given in [31]. The unpublished work [32], shows that for any permutation σ of $\{1, 2, \cdots, d\}$, $I_\sigma = \prod_{i=1}^{d} \frac{a_{i\sigma(i)}}{a_{ii}}$ is an invariant of the Sinkhorn algorithm. Moreover, [33] and [32] shows that these invariants completely determine the DSM that the Sinkhorn algorithm converges to. It is easy to verify that I_σ is an invariant for any algorithm that proceeds by normalizing rows and columns. Hence, if it converges, any such algorithm has the same limit point as that of the Sinkhorn algorithm.

Since we are interested in sampling from Birkhoff polytope, we would like to study the probability distribution that induces uniform distribution on the output of the Sinkhorn algorithm. It was shown in [34] that if the entries of the initial matrix are chosen independently, the distribution that the Sinkhorn algorithm induces on the Birkhoff polytope would not be uniform. For 2×2 case, [32] shows that choosing the entries in the following way would induce a uniform distribution: Choose a_{11} and a_{12} independently from the squared uniform distribution, fix $a_{21} = 1 - a_{11}$ and $a_{22} = 1 - a_{12}$. But, how to achieve this for higher dimension cases is not clear.

In the rest of this chapter, we explore other possible algorithms that converge to DSMs. We want algorithms to converge to a different DSM from what the Sinkhorn algorithm converges to, starting from the same initial matrix. This is hoping that we might be able to find distributions for the initial matrix such that the algorithm induces a uniform distribution on the Birkhoff polytope.

5.2 An alternative to the Sinkhorn algorithm

We describe an algorithm that converges to a different DSM than the Sinkhorn algorithm, starting from the same initial matrix. Let $A = (a_{ij})$ be a non-negative matrix with positive main diagonal. Let $A_0 = (a_{ij,0}) = A$. Then we form a sequence of matrices $A_n = (a_{ij,n})$ as follows:

Let $R_{i,n+1}$ and $C_{j,n+1}$ denote the i^{th} row sum and j^{th} column sum of A_n respectively. That is, for every $1 \leq i, j \leq d$ we have

$$\begin{aligned} R_{i,n+1} &= \sum_{j=1}^{d} a_{ij,n}, \\ C_{j,n+1} &= \sum_{i=1}^{d} a_{ij,n}. \end{aligned} \tag{5.1}$$

Then obtain entries of A_{n+1} as:

$$a_{ij,n+1} = \begin{cases} \frac{a_{ij,n}}{R_{i,n+1}C_{j,n+1}} & , \text{ if } i \neq j \\ \frac{a_{ij,n}}{R_{i,n+1}^{\alpha}C_{j,n+1}^{1-\alpha}} & , \text{ otherwise,} \end{cases}$$

where $0 < \alpha < 1$. Recursively substituting for the values of $a_{ij,n+1}$ in (5.1), we get

$$\begin{aligned} R_{i,n+1} &= \sum_{j \neq i} \frac{a_{ij}}{\prod_{k=1}^{n} R_{i,k}C_{j,k}} + \frac{a_{ii}}{\prod_{k=1}^{n} R_{i,k}^{\alpha}C_{i,k}^{1-\alpha}}, \\ C_{j,n+1} &= \sum_{i \neq j} \frac{a_{ij}}{\prod_{k=1}^{n} R_{i,k}C_{j,k}} + \frac{a_{jj}}{\prod_{k=1}^{n} R_{j,k}^{\alpha}C_{j,k}^{1-\alpha}}, \end{aligned} \tag{5.2}$$

for every $1 \leq i, j \leq d$. Let $P_{i,n} = \prod_{k=1}^{n} R_{i,k}$ and $Q_{j,n} = \prod_{k=1}^{n} C_{j,k}$. Then (5.2) gives

$$\begin{aligned} P_{i,n+1} &= \sum_{j \neq i} \frac{a_{ij}}{Q_{j,n}} + a_{ii}\left(\frac{P_{i,n}}{Q_{i,n}}\right)^{1-\alpha}, \\ Q_{j,n+1} &= \sum_{i \neq j} \frac{a_{ij}}{P_{i,n}} + a_{jj}\left(\frac{Q_{j,n}}{P_{j,n}}\right)^{\alpha}, \end{aligned} \tag{5.3}$$

for every $1 \leq i, j \leq d$. Let $P_n = [P_{1,n}, P_{2,n}, \cdots, P_{d,n}]^T$ and $Q_n = [Q_{1,n}, Q_{2,n}, \cdots, Q_{d,n}]^T$. Let vector u denote the diagonal of A and let V denote the matrix obtained by forcing the principal diagonal of A to be zero. That is,

$$V = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1d} \\ a_{21} & 0 & \cdots & a_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ a_{d1} & a_{d2} & \cdots & 0 \end{bmatrix}.$$

Also let $\odot$ denote the element-wise multiplication operation between two vectors. Entries of the vector $\frac{1}{x}$ are the inverse of elements of the vector x. Exponentiation of a vector is

defined as the element-wise exponentiation. Then (5.3) gives

$$\begin{aligned} P_{n+1} &= u \odot \left(P_n \odot \frac{1}{Q_n} \right)^{1-\alpha} + V \frac{1}{Q_n}, \\ Q_{n+1} &= u \odot \left(Q_n \odot \frac{1}{P_n} \right)^{\alpha} + V^T \frac{1}{P_n}. \end{aligned} \tag{5.4}$$

In simulations we observe that this algorithm converges starting from a positive matrix A. In the next section, we analytically prove the convergence of this algorithm in the special case when A is a symmetric matrix.

5.2.1 Convergence when A is a symmetric matrix

When A is symmetric, we see that $R_{k,n} = C_{k,n}$ and hence $P_{k,n} = Q_{k,n}$ for each $1 \leq k \leq d$. Thus, (5.4) reduces to

$$P_{n+1} = u + V \frac{1}{P_n}. \tag{5.5}$$

To show that the row and column sums $R_{k,n}$ converges to 1 as $n \to \infty$, it suffices to show that the sequence $(P_{k,n})_{n \in \mathbb{N}}$ converges to a positive value. Let the function $g : \mathbb{R}_+^d \to \mathbb{R}_+^d$ be defined as

$$g(x) = u + V \frac{1}{u + V \frac{1}{x}}.$$

Let g^m denote the m compositions of the function g. Suppose the iteration $g^m(x)$ converges to a unique fixed point as $m \to \infty$. Then the even and odd subsequences of the sequence $(P_n)_{n \in \mathbb{N}}$ converges and the limit is the fixed point of the mapping g. Since the limit is same, we conclude that the sequence $(P_n)_{n \in \mathbb{N}}$ converges. Before proving that $g^m(x)$ converges to a unique fixed point as $m \to \infty$, a few definitions are in order. The following development is based on chapter 1 and 2 in [35].

Definition 5. (**Cone**) Let S be a finite dimensional vector space. A subset K of S is called a cone if it is convex, $\mu K \subseteq K$ for all $\mu \geq 0$ and $K \cap (-K) = \{0\}$.

An example of a cone is the standard positive cone $\mathbb{R}_+^d = \{(x_1, x_2, \cdots, x_d) \in \mathbb{R}^d : x_i \geq 0 \quad \forall \quad 1 \leq i \leq d\}$.

Definition 6. (**Partial order in a cone**) A cone K induces a partial order defined as $x \leq_K y$ if $y - x \in K$. Also denote $x \ll_K y$ if $y - x \in \text{int}(K)$.

If $K = \mathbb{R}_+^d$ then $x \leq_K y$ if and only if $x_i \leq y_i$ for all $1 \leq i \leq d$.

Definition 7. (**Parts of a cone**) An equivalence relation $\sim_K$ on K is defined as $x \sim_K y$ if there exists $0 < \alpha \leq \beta$ such that $\alpha y \leq_K x \leq_K \beta y$. Equivalence classes in K are called the parts of the cone K. Particularly, $\text{int}(\mathbb{R}^d_+)$ is a part of the cone $\mathbb{R}^d_+$.

Definition 8. (**Order preserving map**) A function $f : K \to K$ is order preserving if $x \leq_K y$ implies $f(x) \leq_K f(y)$ for all $x, y \in K$.

Definition 9. (**Subhomogeneous maps**) A map $f : K \to K$ is subhomogeneous if $\lambda f(x) \leq_K f(\lambda x)$ for all $x \in K$ and $0 < \lambda < 1$. It is said to be strictly subhomogeneous if $\lambda f(x) \ll_K f(\lambda x)$ for all $x \in K \backslash \{0\}$ and $0 < \lambda < 1$.

Definition 10. (**Thompson's metric**) For $x, y \in K$ with $x \sim_K y$ and $y \neq 0$, define

$$M(x/y) = \inf\{\beta > 0 : x \leq_K \beta y\}. \tag{5.6}$$

Thompson's metric $d_T : K \times K \to [0, \infty]$ is defined by

$$d_T(x, y) = \begin{cases} \log(\max\{M(x/y), M(y/x)\}) & , \text{ if } x \sim_K y \text{ and } y \neq 0 \\ 0 & , \text{ if } x, y = 0 \\ \infty & , \text{ otherwise.} \end{cases}$$

Claim 1. *Thompson's metric d_T is a genuine metric on each part of a closed cone K.*

Proof. • Positivity: We need to show that $d_T(x, y) \geq 0$ and $d_T(x, y) = 0$ if and only if $x = y$. First we make the following observation:

$$\begin{aligned} M(y/x) &= \inf\{\beta > 0 : \frac{1}{\beta} y \leq_K x\} \\ &= \sup\{\alpha > 0 : \alpha y \leq_K x\}^{-1} \\ &\geq \frac{1}{M(x/y)}. \end{aligned}$$

$$\begin{aligned} &\implies \max\{M(y, x), M(x, y)\} \geq 1 \\ &\implies d_T(x, y) \geq 0. \end{aligned}$$

We also have $M(x/x) = 1$, which gives $d_T(x, x) = 0$. Now let $d_T(x, y) = 0$.

$$\begin{aligned} &\implies M(x/y) = M(y/x) = 1 \\ &\implies x \leq_K (1 + \epsilon) y \quad \text{and} \quad y \leq_K (1 + \epsilon) x \quad \text{for all} \quad \epsilon > 0 \\ &\implies y - x + \epsilon y, \quad x - y + \epsilon x \in K \quad \text{for all} \quad \epsilon > 0 \\ &\implies y - x, \quad x - y \in K \quad [\because \text{ K is closed}] \\ &\implies x = y. \end{aligned}$$

- Symmetry: We need to show that $d_T(x, y) = d_T(y, x)$. But this is trivial from the definition of d_T.
- Triangle inequality: We need to show that $d_T(x, z) \leq d_T(x, y) + d_T(y, z)$. We have $x \leq_K \alpha y$ and $y \leq_K \beta z$ for all $\alpha > M(x/y)$ and $\beta > M(y/z)$.

$$\begin{aligned}
&\implies x \leq_K \alpha\beta z \\
&\implies M(x/z) \leq \alpha\beta \quad \text{for all } \alpha > M(x/y) \text{ and } \beta > M(y/z) \\
&\implies M(x/z) \leq M(x/y)M(y/z).
\end{aligned}$$

Thus,

$$\begin{aligned}
\max\{M(x/z), M(z/x)\} &\leq \max\{M(x/y)M(y/z), M(y/x)M(z/y)\} \\
&\leq \max\{M(x/y), M(y/x)\} \max\{M(y/z), M(z/y)\} \\
&\implies d_T(x, z) \leq d_T(x, y) + d_T(y, z).
\end{aligned}$$

□

Lemma 1. *Suppose K is the standard positive cone $\mathbb{R}^d_+$. Then, for all $x, y \in int(\mathbb{R}^d_+)$ we have $d_T(x, y) = \|L(x) - L(y)\|_\infty$, where L denotes the element-wise logarithm operation on a vector.*

Proof. When $K = \mathbb{R}^d_+$, we have

$$\begin{aligned}
M(x/y) &= \inf\{\beta > 0 : x \leq_K \beta y\} \\
&= \inf\{\beta > 0 : x_i \leq \beta y_i \quad \forall \quad 1 \leq i \leq d\} \\
&= \max\left\{\frac{x_i}{y_i} : 1 \leq i \leq d\right\}.
\end{aligned}$$

Thus,

$$\begin{aligned}
d_T(x, y) &= \log(\max\{M(x/y), M(y/x)\}) \\
&= \log\left(\max\left\{\frac{x_i}{y_i} \vee \frac{y_i}{x_i} : 1 \leq i \leq d\right\}\right) \\
&= \max\left\{\log\left(\frac{x_i}{y_i}\right) \vee \log\left(\frac{y_i}{x_i}\right) : 1 \leq i \leq d\right\} \\
&= \max\{|\log(x_i) - \log(y_i)| : 1 \leq i \leq d\} \\
&= \|L(x) - L(y)\|_\infty,
\end{aligned}$$

where L is the element-wise logarithm operation on a vector.

□

Lemma 2. *Let $X = int(\mathbb{R}^d_+)$. If $E \subseteq X$ is compact in $(X, \|\cdot\|_\infty)$, then it is compact in (X, d_T).*

Proof. Let $E \subseteq X$ be compact in $(X, \|\cdot\|_\infty)$.
The element-wise logarithm function $L : (X, \|\cdot\|_\infty) \to (\mathbb{R}^d, \|\cdot\|_\infty)$ is continuous. Then, for every $x \in X$ and $\epsilon > 0$ there exits $\delta > 0$ such that

$$\|L(y) - L(x)\|_\infty < \epsilon \quad \text{whenever} \quad \|y - x\|_\infty < \delta \text{ and } y \in X.$$

Thus by Lemma 1, for every $x \in X$ and $\epsilon > 0$ there exits $\delta > 0$ such that

$$d_T(y, x) < \epsilon \quad \text{whenever} \quad \|y - x\|_\infty < \delta \text{ and } y \in X.$$

This implies that the identity function $I : (X, \|\cdot\|_\infty) \to (X, d_T)$ is continuous. Since the image of a compact set under a continuous function is compact, we have $E = I(E)$ compact in (X, d_T). □

In fact, $(\text{int}(\mathbb{R}^d_+), d_T)$ is a complete metric space and its topology coincides with the norm topology [35].

Definition 11. (**Contraction**) A mapping f from a metric space (X, d) to itself is a contraction if

$$d(f(x), f(y)) < d(x, y) \text{ for all } x, y \in X, x \neq y. \tag{5.7}$$

Lemma 3. *(Lemma 2.1.7 in [35]) Let K be a closed cone. If $f : int(K) \to int(K)$ is order-preserving and strictly subhomogeneous, then f is a contraction.*

Proof. Let $x, y \in \text{int}(K)$, and $x \neq y$. Then, $d_T(x, y) = \log(\lambda)$ for some $\lambda > 1$. Also since K is closed, we have $x \leq_K \lambda y$ and $y \leq_K \lambda x$. Thus,

$$\begin{aligned} &\frac{1}{\lambda} f(x) \ll_K f(\frac{x}{\lambda}) \leq_K f(y), \\ &\frac{1}{\lambda} f(y) \ll_K f(\frac{y}{\lambda}) \leq_K f(x). \end{aligned}$$

$$\begin{aligned} &\implies \max\{M(f(x)/f(y)), M(f(y)/f(x))\} < \lambda \\ &\implies d_T(f(x), f(y)) < \log(\lambda) = d_T(x, y). \end{aligned}$$

□

We will work on the standard positive cone $\mathbb{R}^d_+$. It is easy to see that g is order-preserving on $\mathbb{R}^d_+$. Let $x \leq_K y$ and let $v_1, v_2, \cdots, v_d$ denote the rows of the matrix V. Then,

$$\begin{aligned} g(y) - g(x) &= V\left[\frac{1}{u + V\frac{1}{y}} - \frac{1}{u + V\frac{1}{x}}\right] \\ &= V\begin{bmatrix} \frac{<v_1, \frac{1}{x} - \frac{1}{y}>}{\left(u_1 + <v_1, \frac{1}{x}>\right)\left(u_1 + <v_1, \frac{1}{y}>\right)} \\ \vdots \\ \frac{<v_d, \frac{1}{x} - \frac{1}{y}>}{\left(u_d + <v_d, \frac{1}{x}>\right)\left(u_d + <v_d, \frac{1}{y}>\right)} \end{bmatrix} \\ &\in \mathbb{R}^d_+. \end{aligned}$$

Hence $g(x) \leq_K g(y)$. Therefore g is order-preserving. We can also show that g is strictly subhomogeneous. For $0 < \lambda < 1$, we have

$$\begin{aligned} g(\lambda x) - \lambda g(x) &= (1-\lambda)u + \lambda V\left[\frac{1}{\lambda u + V\frac{1}{x}} - \frac{1}{u + V\frac{1}{x}}\right] \\ &= (1-\lambda)u + \lambda V\begin{bmatrix} \frac{(1-\lambda)u_1}{\left(\lambda u_1 + <v_1, \frac{1}{x}>\right)\left(u_1 + <v_1, \frac{1}{x}>\right)} \\ \vdots \\ \frac{(1-\lambda)u_d}{\left(\lambda u_d + <v_d, \frac{1}{x}>\right)\left(u_d + <v_d, \frac{1}{x}>\right)} \end{bmatrix} \\ &\in \text{int}(\mathbb{R}^d_+). \end{aligned}$$

Hence $\lambda g(x) \ll_K g(\lambda x)$. Thus by Lemma 3, g is a contraction on $(\text{int}(\mathbb{R}^d_+), d_T)$. Now we use the following theorem to show that iteration $g^k(x)$ converges for all $x \in \text{int}(\mathbb{R}^d_+)$.

Theorem 3. *(Theorem 3.2.2 in [35]) If $f : X \to X$ is a contraction on a compact metric space, then there exists a unique $x^* \in X$ such that $f(x^*) = x^*$ and $\lim_{k\to\infty} f^k(x) = x^*$ for all $x \in X$.*

Proof. Let $w(x)$ denote the set of all subsequential limits of $f^k(x)$. Since X is a compact metric space, every infinite subset of X has a limit point in X. Hence $w(x)$ is non-empty. Let $x^* \in w(x)$. Then there exists a subsequence $f^{k_i}(x)$ that converges to x^*. Also, since

f is a contraction, it is continuous, which implies that $f^{k_i+1}(x) \to f(x^*)$. Then,

$$\begin{aligned} d(x^*, f(x^*)) &\leq d(\mathbf{x}^*, f^{k_i}(x)) + d(f^{k_i}(x), f^{k_i+1}(x)) + d(f^{k_i+1}(x), f(x^*)) \\ &\leq \lim_{i\to\infty} d(f^{k_i}(x), f^{k_i}(f(x))) \\ &\leq \lim_{i\to\infty} d(f^{k_i-1}(f(x)), f^{k_i-1}(f^2(x))) \quad [\because f \text{ is a contraction}] \\ &\leq d(f(x^*), f^2(x^*)) \quad [\because \text{Triangle inequality}]. \end{aligned}$$

If $f(x^*) \neq f^2(x^*)$, we have $d(f(x^*), f^2(x^*)) < d(x^*, f(x^*))$ which is a contradiction. Thus, $f(x^*) = f^2(x^*)$ and hence $x^* = f(x^*)$. Thus x^* is a fixed point of f. Since f is a contraction, x^* is unique. Now, compactness of X implies that $f^k(x) \to x^*$ for all $x \in X$. □

It can be verified that range of g is a bounded set. Each component g_i of g is given by

$$\begin{aligned} g_i(x) &= u_i + < v_i, \frac{1}{u + V\frac{1}{x}} > \\ &\leq u_i + < v_i, \frac{1}{u} > . \end{aligned}$$

Thus, we have $u \leq_K g(x) \leq_K u + V\frac{1}{u}$. Thus $g(x)$ lives inside a rectangle. Then g can be restricted to a compact subset in $(\text{int}(\mathbb{R}^d_+), \|\cdot\|_\infty)$ which by Lemma 2 is a compact subset in $(\text{int}(\mathbb{R}^d_+), d_T)$. Therefore the iteration $g^k(x)$ converges by Theorem 3.

5.2.2 Numerical Experiments

This section presents a numerical experiment that show that the alternative to the Sinkhorn algorithm just discussed converge to a DSM different from the one Sinkhorn algorithm converges to. We consider the case when d equals 2. Let squared uniform be the probability distribution with density $f(x) = \frac{1}{2\sqrt{x}}$, for $0 < x \leq 1$. Let $A = (a_{ij})$ denote the initial matrix. Then, given an iterative algorithm that converges to a DSM, our experiment is as follows: Sample the entries a_{11} and a_{12} independently and identically from the squared uniform distribution. Set $a_{21} = 1 - a_{11}$ and $a_{22} = 1 - a_{12}$. Run the iterative algorithm to generate DSM, starting from matrix A. Let $B = (b_{ij})$ be the DSM that the algorithm converges to. Empirically estimate the probability distribution of b_{11} by repeating the above procedure multiple times (30,000 times for this experiment). Figure 5.1 shows the cumulative distribution function (CDF) of b_{11} when the alternative to the Sinkhorn algorithm is used for the iteration, for different values of α used in

(5.1). It also displays the CDF when the Sinkhorn algorithm is used for the iteration. We observe that the probability distribution of b_{11} is different when different iterative

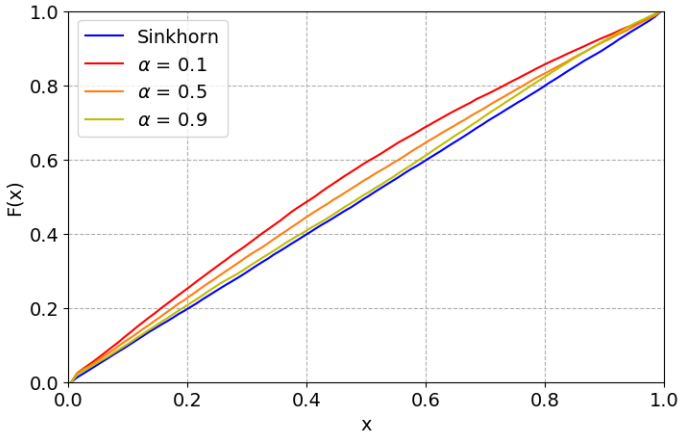

Figure 5.1: CDF of b_{11} for different values of α

algorithms are used to generate DSM. This shows that the alternative to the Sinkhorn algorithm that we introduced converges to a DSM that is different from the limit point of the Sinkhorn algorithm.

5.3 Conclusion and Future Work

This work showed that there exist iterative algorithms that converge to a DSM different from the one Sinkhorn algorithm converges to. There are more possibilities for such algorithms. For example, let $R_{i,n+1}$ and $C_{j,n+1}$ be as defined in (5.1). Let a sequence of matrices $A_n = (a_{ij,n})$ for $n \in \mathbb{N}$ be formed as follows:

$$a_{ij,n+1} = \begin{cases} \frac{a_{ij,n}}{R_{i,n+1}C_{j,n+1}} & , \text{ if } i \neq j \\ \frac{a_{ij,n}}{(R_{i,n+1}C_{j,n+1})^p} & , \text{ otherwise.} \end{cases}$$

for every $1 \leq i, j \leq d$. In numerical experiments, we observe that the sequence $(A_n)_{n\in\mathbb{N}}$ converge to a DSM when $0 < p \leq 1$ and initial matrix is positive. The matrix to which this algorithm converges is dependent on the choice of p.

Another possible iterative algorithm is to obtain $A_n = (a_{ij,n})$ as follows:

$$a_{ij,n+1} = \frac{a_{ij,n}}{(1 + w_{ij}(R_{i,n+1} - 1))(1 + w_{ij}(C_{j,n+1} - 1))},$$

where $0 < w_{ij} \leq 1$ for every $1 \leq i, j \leq d$. Even this algorithm converges to a DSM starting from a positive initial matrix. The matrix to which this algorithm converges is dependent on the choice of w_{ij}.

We proved the convergence of the alternative algorithm to Sinkhorn only when the initial matrix A is symmetric. Convergence for general case is yet to be shown. From what distribution should we choose the entries of the initial matrix so that any such iterative algorithm converges to a DSM whose distribution is uniform on the Birkhoff polytope, is still an open problem.

Appendices

Appendix A

Effect of non-ideality of ALG in IMHRZ

While proving the uniform ergodicity of the IMHRZ algorithm, we assumed that the method used to draw samples from π, **ALG** is ideal. Here we analyze the case when **ALG** is non-ideal. We assume that **ALG** itself is an MCMC method. Due to finite time given for convergence in **ALG**, the proposal generated will have different probability distribution from one which is used in the calculation of acceptance ratio. For probability densities for which **ALG** is geometrically ergodic, the error due to this non-ideality can be bounded as shown in the following Proposition.

Proposition 5. *Let π be such that φ is L-smooth. Let $\mathbf{t} \in \mathbb{R}^d$ be a fixed initial state of **ALG**. Also let **ALG** satisfy the following geometric ergodicity condition*

$$\|P^n(\mathbf{t}, \cdot) - \pi(\cdot)\|_{TV} \leq V\rho^n, \tag{A.1}$$

*where P is the transition kernel corresponding to **ALG**.Then for any $\mathbf{x} \in \mathbb{Z}^d$, Algorithm 3 generates a Markov chain with transition probability $\bar{P}$ that satisfies the following inequality*

$$\|\bar{P}^k(\mathbf{x}, \cdot) - P_{\mathbb{Z}^d}(\cdot)\|_{TV} \leq (1 - C\delta)^k + (1 + \frac{1}{C\delta})\frac{V\rho^n}{\delta}, \tag{A.2}$$

*where δ is obtained from (2.5), n is the number of iterations of **ALG**, and C is a constant that satisfy the following inequality*

$$1 - \frac{2V\rho^n}{\delta} \leq C \leq 1 + \frac{2V\rho^n}{\delta}. \tag{A.3}$$

Proof. From geometric ergodicity of **ALG** we have

$$\sup_{A\subseteq\mathbb{R}^d} P^n(\mathbf{t}, A) - \pi(A) \leq V\rho^n$$

Here $\mathbf{t}$ is a fixed initial state of **ALG**. We obtain the candidate state $\mathbf{z} \in \mathbb{Z}^d$ by quantizing the sample returned by **ALG** to its nearest integer point. Let $\bar{q}$ denote the proposal distribution thus obtained.

$$\bar{q}(\mathbf{z}) = P^n(\mathbf{t}, A_{\mathbf{z}}),$$

where $A_{\mathbf{z}}$ is the d dimesional unit hypercube centered at $\mathbf{z}$. Let q denote the proposal distribution if **ALG** does an ideal sampling from π.

$$q(\mathbf{z}) = \pi(A_{\mathbf{z}})$$

Then,

$$\begin{aligned}
\|\bar{q}(\cdot) - q(\cdot)\|_{TV} &= \sup_{B\subseteq\mathbb{Z}^d} \sum_{\mathbf{z}\in B} \bar{q}(\mathbf{z}) - q(\mathbf{z}) \\
&= \sup_{B\subseteq\mathbb{Z}^d} \sum_{\mathbf{z}\in B} P^n(\mathbf{t}, A_{\mathbf{z}}) - \pi(A_{\mathbf{z}}) \\
&= \sup_{B\subseteq\mathbb{Z}^d} P^n(\mathbf{t}, A_B) - \pi(A_B) \\
&\leq \sup_{A\subseteq\mathbb{R}^d} P^n(\mathbf{t}, A) - \pi(A) \\
&\leq V\rho^n
\end{aligned}$$

Although **ALG** generates the proposed state from the distribution $\bar{q}$, for calculating the acceptance ratio, we use the distribution q. Thus Markov chain generated by MH step has the following transition probability

$$\bar{P}(\mathbf{x}, \mathbf{y}) = \begin{cases} \bar{q}(\mathbf{y}) \min\left(1, \frac{P_{\mathbb{Z}^d}(\mathbf{y})q(\mathbf{x})}{P_{\mathbb{Z}^d}(\mathbf{x})q(\mathbf{y})}\right), & \text{if } \mathbf{y} \neq \mathbf{x}, \\ 1 - \sum_{\mathbf{z}\neq\mathbf{x}} \bar{q}(\mathbf{z}) \min\left(1, \frac{P_{\mathbb{Z}^d}(\mathbf{z})q(\mathbf{x})}{P_{\mathbb{Z}^d}(\mathbf{x})q(\mathbf{z})}\right), & \text{if } \mathbf{y} = \mathbf{x}. \end{cases}$$

Here $P_{\mathbb{Z}^d}$ is the desired d-dimensional lattice distribution. Stationary distribution of the above Markov chain $\bar{P}$ is

$$\bar{P}_{\mathbb{Z}^d}(\mathbf{x}) = \frac{P_{\mathbb{Z}^d}(\mathbf{x})\bar{q}(\mathbf{x})}{Cq(\mathbf{x})},$$

where,

$$C = \sum_{\mathbf{x}} \frac{P_{\mathbb{Z}^d}(\mathbf{x})\bar{q}(\mathbf{x})}{q(\mathbf{x})}.$$

This can be verified using detailed balance equation. Since φ is L-smooth, from Proposition 2, we have

$$\frac{q(\mathbf{x})}{P_{\mathbb{Z}^d}(\mathbf{x})} \geq \delta > 0,$$
$$\frac{\bar{q}(\mathbf{x})}{\bar{P}_{\mathbb{Z}^d}(\mathbf{x})} = \frac{Cq(\mathbf{x})}{P_{\mathbb{Z}^d}(\mathbf{x})} \geq C\delta > 0.$$

Thus $\bar{P}$ is a uniformly ergodic Markov chain. Hence

$$\|\bar{P}^k(\mathbf{x},\cdot) - \bar{P}_{\mathbb{Z}^d}(\cdot)\|_{TV} \leq (1-C\delta)^k. \tag{A.4}$$

We also have,

$$\begin{aligned}
|C-1| &= \left|\sum_{\mathbf{x}} \frac{P_{\mathbb{Z}^d}(\mathbf{x})\bar{q}(\mathbf{x})}{q(\mathbf{x})} - P_{\mathbb{Z}^d}(\mathbf{x})\right| \\
&\leq \sum_{\mathbf{x}} \frac{P_{\mathbb{Z}^d}(\mathbf{x})}{q(\mathbf{x})}|\bar{q}(\mathbf{x}) - q(\mathbf{x})| \\
&\leq \frac{1}{\delta}\sum_{\mathbf{x}} |\bar{q}(\mathbf{x}) - q(\mathbf{x})| \\
&\leq \frac{2V\rho^n}{\delta}. \\
\implies 1 - \frac{2V\rho^n}{\delta} &\leq C \leq 1 + \frac{2V\rho^n}{\delta}.
\end{aligned} \tag{A.5}$$

Now the TVD between $\bar{P}_{\mathbb{Z}^d}$ and $P_{\mathbb{Z}^d}$ can be bounded as follows:

$$\begin{aligned}
\|\bar{P}_{\mathbb{Z}^d}(\cdot) - P_{\mathbb{Z}^d}(\cdot)\|_{TV} &= \frac{1}{2}\sum_{\mathbf{x}\in\mathbb{Z}^d} \left|\frac{P_{\mathbb{Z}^d}(\mathbf{x})\bar{q}(\mathbf{x})}{Cq(\mathbf{x})} - P_{\mathbb{Z}^d}(\mathbf{x})\right| \\
&= \frac{1}{2}\sum_{\mathbf{x}\in\mathbb{Z}^d} \frac{P_{\mathbb{Z}^d}(\mathbf{x})}{Cq(\mathbf{x})}|\bar{q}(\mathbf{x}) - Cq(\mathbf{x})| \\
&\leq \frac{1}{2C\delta}\sum_{\mathbf{x}\in\mathbb{Z}^d} C|\bar{q}(\mathbf{x}) - q(\mathbf{x})| + |1-C|\bar{q}(\mathbf{x}) \\
&= \frac{1}{\delta}\|\bar{q}(\cdot) - q(\cdot)\|_{TV} + \frac{|1-C|}{2C\delta} \\
&\leq \frac{V\rho^n}{\delta} + \frac{V\rho^n}{C\delta^2}.
\end{aligned}$$

Finally using triangle inequality, we have

$$\begin{aligned}
\|\bar{P}^k(\mathbf{t},\cdot) - P_{\mathbb{Z}^d}(\cdot)\|_{TV} &\leq \|\bar{P}^k(\mathbf{t},\cdot) - \bar{P}_{\mathbb{Z}^d}(\cdot)\|_{TV} + \|\bar{P}_{\mathbb{Z}^d}(\cdot) - P_{\mathbb{Z}^d}(\cdot)\|_{TV} \\
&\leq (1-C\delta)^k + (1 + \frac{1}{C\delta})\frac{V\rho^n}{\delta}.
\end{aligned} \tag{A.6}$$

□

Appendix B

Geometric Ergodicity of RWMR

Here we elaborate on the geometric ergodicity of RWMR algorithm. First we present few definitions.

Definition 12. A probability density π is called sub-exponential if it satisfies the following condition

$$\lim_{\|\mathbf{x}\|\to\infty} \hat{\mathbf{x}} \cdot \nabla \log \pi(\mathbf{x}) = -\infty, \tag{B.1}$$

where $\hat{\mathbf{x}}$ denotes the unit vector in the direction of $\mathbf{x}$. This implies that for any $a > 0, h > 0$, there exists an r such that for all $\mathbf{x}$ satisfying $\|\mathbf{x}\| > r$

$$\frac{\pi(\mathbf{x} + a\hat{\mathbf{x}})}{\pi(\mathbf{x})} \leq e^{-ah}. \tag{B.2}$$

Definition 13. A Markov chain with transition kernel P is said to satisfy minorization condition with respect to a set $C \subseteq \mathcal{X}$ if there exists $k > 0, \delta > 0$ and a probability measure ν such that

$$P^k(\mathbf{x}, B) \geq \delta\nu(B), \quad \mathbf{x} \in C \tag{B.3}$$

for all measurable $B \subseteq \mathcal{X}$. The set C is referred to as a small set.

Corollary 3. *For a Markov chain generated by Metropolis-Hastings algorithm over $\mathbb{Z}^d$, any non-empty compact subset of $\mathbb{Z}^d$ is small if the proposal distribution is positive everywhere.*

Proof. Let q denote the proposal distribution and P be the transition probability. Since $q(\mathbf{x}, \cdot)$ is positive erverywhere, we have

$$P(\mathbf{x}, \mathbf{y}) \geq q(\mathbf{x}, \mathbf{y})\alpha(\mathbf{x}, \mathbf{y}) > 0.$$

Let $C \subseteq \mathbb{Z}^d$ be compact. Then,

$$\inf_{\mathbf{x}\in C} P(\mathbf{x},\mathbf{y}) > 0, \quad \forall \quad \mathbf{y} \in \mathbb{Z}^d.$$

Let

$$\epsilon = \sum_{\mathbf{y}\in\mathbb{Z}^d} \inf_{\mathbf{x}\in C} P(\mathbf{x},\mathbf{y}),$$
$$\nu(\mathbf{y}) = \epsilon^{-1} \inf_{\mathbf{x}\in C} P(\mathbf{x},\mathbf{y}).$$

Then we have,

$$P(\mathbf{x},\mathbf{y}) \geq \epsilon\nu(\mathbf{y}), \quad \forall \mathbf{x} \in C, \quad \forall \mathbf{y} \in \mathbb{Z}^d$$

Therefore C is a small set. □

Definition 14. A Markov chain with discrete state space $\mathcal{X}$ satisfies the drift condition if there are constants $0 < \lambda < 1$, $b < \infty$ and a Lyapunov function $V : \mathcal{X} \to [1, \infty)$ such that

$$\sum_{\mathbf{y}\in\mathcal{X}} P(\mathbf{x},\mathbf{y})V(\mathbf{y}) \leq \lambda V(\mathbf{x}) + b\mathbb{1}_C(\mathbf{x}) \tag{B.4}$$

for all $\mathbf{x} \in \mathcal{X}$, where $C \subseteq \mathcal{X}$ is a small set.

It is well known that drift condition is a sufficient condition for a Markov chain to be geometrically ergodic [14].

Proposition 6. *The Markov chain generated by RWMR is geometrically ergodic if π is a sub-exponential probability density.*

Proof. From Corollary 3 it is clear that any compact subset of $\mathbb{Z}^n$ is a small set. Therefore to prove the drift condition, it suffices to show that

$$\lambda = \limsup_{\|\mathbf{x}\|\to\infty} \sum_{\mathbf{y}\in\mathbb{Z}^n} \frac{P(\mathbf{x},\mathbf{y})V(\mathbf{y})}{V(\mathbf{x})} < 1, \tag{B.5}$$

for some Lyapunov function V. By arguments similar to proof of Theorem 2 in [6] we conclude that the above condition is satisfied. □

Appendix C

Hamiltonian Dynamics based Symmetric Proposal Algorithm

Hamiltonian dynamics can generate proposals for the Metropolis-Hastings algorithm, which are far away from the current state of the Markov chain but has a high probability of getting accepted as the next state. The challenge in using Hamiltonian dynamics for sampling from discrete distributions is to maintain the symmetric nature of the proposal distribution q. In Algorithm 7 we describe a simple way to accomplish this. First, the momentum variables are introduced as required by Hamiltonian dynamics. Here, the state-space of the Markov chain is the set of all integer points in phase space. To each component of the current state of the Markov chain, which is an integer point, we add a uniform random variable with support $[-\frac{1}{2}, \frac{1}{2}]$. With the point thus obtained as the initial state, we run Hamiltonian dynamics. The final point, given by Hamiltonian dynamics, is rounded off to the nearest integer point to get the proposed state. Using conservation of Hamiltonian and volume-preserving properties of Hamiltonian dynamics, we can show that such a proposal distribution is symmetric. As in the case of HMC, at the beginning of each iteration, momentum variables need to be resampled from lattice Gaussian distribution $D_{\mathbb{Z}^d,\sigma,0}$ (see equation 1.3). We use a simple rejection sampling method mentioned in [1] for this.

Proposition 7. *The proposal distribution in Algorithm 7 is symmetric in nature.*

Proof. Let $\mathbf{w}, \mathbf{z}$ be two points in the phase space and let T denote the transformation corresponding to Hamiltonian dynamics. Also let $A_{\mathbf{w}}$ and $A_{\mathbf{z}}$ denote the unit hypercube centered at $\mathbf{w}$ and $\mathbf{z}$ respectively. It suffices to prove that the volume of points in $A_{\mathbf{w}}$

Algorithm 7: Hamiltonian Dynamics based Symmetric Algorithm

Input: $\mathbf{B}, \varphi, \nabla\varphi, \mathbf{X}_0, \sigma, \epsilon, L$

Output: Sample from a distribution statistically close to P_Λ

for $t = 1, 2, \ldots$ **do**

Let $\mathbf{x}$ be the state of $\mathbf{X}_t$ Sample $\mathbf{p}_0$ from $D_{\mathbb{Z}^d,\sigma,0}$;

Generate $\mathbf{u}_x \sim \mathbf{U}^d[-\frac{1}{2}, \frac{1}{2}]$ and $\mathbf{u}_p \sim \mathbf{U}^d[-\frac{1}{2}, \frac{1}{2}]$;

$\mathbf{x}^i \leftarrow \mathbf{x} + \mathbf{u}_x$;

$\mathbf{p}^i \leftarrow \mathbf{p}_0 + \mathbf{u}_p$;

$(\mathbf{x}^f, \mathbf{p}^f) \leftarrow$Algorithm 1$(\mathbf{x}^i, \mathbf{p}^i, \epsilon, L, \nabla\varphi)$;

Round $\mathbf{x}^f, \mathbf{p}^f$ to its nearest point in $\mathbb{Z}^n$ to get $\mathbf{y}, \mathbf{p}$;

Calculate the acceptance ratio:

$\alpha((\mathbf{x}, \mathbf{p}_0), (\mathbf{y}, \mathbf{p})) = 1 \wedge \exp\left(\varphi(\mathbf{x}) - \varphi(\mathbf{y}) + \frac{\|\mathbf{p}_0\|^2 - \|\mathbf{p}\|^2}{2\sigma^2}\right)$;

Generate a sample u from $U[0, 1]$;

if $u \leq \alpha((\mathbf{x}, \mathbf{p}_0), (\mathbf{y}, \mathbf{p}))$ **then**

let $\mathbf{X}_t = \mathbf{y}$;

else

$\mathbf{X}_t = \mathbf{x}$;

end

if $t > t_{mix}(\epsilon)$ **then**

Output the state of $\mathbf{B}\mathbf{X}_t$

end

end

that gets transformed by T to points in $A_\mathbf{z}$ is same as the volume of points in $A_\mathbf{z}$ that gets transformed to points in $A_\mathbf{w}$. We will use the properties of Hamiltonian dynamics described in section 1.3.2 to prove this. Define $A_{\mathbf{w},\mathbf{z}}$ as follows:

$$A_{\mathbf{w},\mathbf{z}} := \{\mathbf{u} \in A_\mathbf{z} : \exists \mathbf{v} \in A_\mathbf{w} \text{ s.t. } T\mathbf{v} = \mathbf{u}\}.$$

$$\mathbf{u} \in A_{\mathbf{w},\mathbf{z}} \implies \exists \mathbf{v} \in A_\mathbf{w} \text{ and } \mathbf{u} \in A_\mathbf{z} \text{ s.t. } T\mathbf{v} = \mathbf{u}.$$

By reversibility of T,

$$T\mathbf{v} = \mathbf{u} \implies T\mathbf{u} = \mathbf{v}.$$

$$\mathbf{u} \in A_{\mathbf{w},\mathbf{z}} \implies \exists \mathbf{u} \in A_\mathbf{z} \text{ and } \mathbf{v} \in A_\mathbf{w} \text{ s.t. } T\mathbf{u} = \mathbf{v}.$$

$$\implies \mathbf{v} = T\mathbf{u} \in A_{\mathbf{z},\mathbf{w}}.$$

$$\implies TA_{\mathbf{w},\mathbf{z}} \subseteq A_{\mathbf{z},\mathbf{w}}.$$

$$TA_{\mathbf{z},\mathbf{w}} \subseteq A_{\mathbf{w},\mathbf{z}}.$$

Due to reversibility we have,

$$TTA_{\mathbf{z},\mathbf{w}} = A_{\mathbf{z},\mathbf{w}}.$$
$$\implies A_{\mathbf{z},\mathbf{w}} \subseteq TA_{\mathbf{w},\mathbf{z}}.$$
$$\implies A_{\mathbf{z},\mathbf{w}} = TA_{\mathbf{w},\mathbf{z}}.$$

Thus, set $A_{\mathbf{w},\mathbf{z}}$ get transformed to set $A_{\mathbf{z},\mathbf{w}}$ under the transformation T and vice-versa. By volume preservation property of Hamiltonian dynamics,

$$Volume(A_{\mathbf{w},\mathbf{z}}) = Volume(A_{\mathbf{z},\mathbf{w}}).$$

□

Optimal tuning of parameters

The main parameters to be fixed in the HDS algorithm are L and ϵ. L and ϵ corresponds to the number of steps and step size used in the Leapfrog integrator. A large value of ϵ will diminish the accuracy of the Leapfrog integrator, while a small value limits the exploration capability of the algorithm. Since L directly impacts the computations required, it is kept constant to have a fair comparison of mixing properties with other algorithms. If we do not scale ϵ with dimension, the acceptance rate degrades. It is known that, like random walk Metropolis, Hamiltonian dynamics also has an optimum acceptance rate which is 0.64 [36]. The step size ϵ should be scaled with dimension as $d^{-\frac{1}{4}}$ to maintain approximately the same acceptance rate. Also, we scaled ϵ according to target distribution as $\sqrt{\lambda}$, where λ is the mean of eigenvalues of covariance matrix corresponding to target density. This will ensure that step size is not too large or small for the target density under consideration. In our experiments, we initially found out the values of ϵ and L which works well for bivariate isotropic Gaussian density. Then ϵ was scaled as mentioned above. Specifically, we used the following equation for ϵ. The value of L used was 10.

$$\epsilon = 0.6 \left(\frac{2}{d}\right)^{\frac{1}{4}} \sqrt{\lambda} \tag{C.1}$$

Appendix D

Piece-wise Constant Approximation for Gaussian Density

In this section, we substantiate the claim in Section 3.1 that the choice of $\bar{\pi}(\mathbf{x}) = \pi([\mathbf{x}])$ can give rise to a Markov chain which is not uniformly ergodic. We show this for a simple case where π is a 1-dimensional Gaussian density. From Theorem 2.1 in [21], it follows that, if $\operatorname{ess\,inf} \frac{\pi(\mathbf{x})}{\bar{\pi}(\mathbf{x})} = 0$ with respect to $\bar{\pi}$ measure, then Independent Metropolis Hastings algorithm is not even geometrically ergodic. Let π be a 1-dimensional Gaussian density and $\bar{\pi}(x) = \pi([x])$. Let $\bar{x}$ denote $[x]$ and let $y = x - \bar{x}$. Then,

$$\begin{aligned} \frac{\pi(x)}{\bar{\pi}(x)} &= M \frac{e^{-(\bar{x}+y)^2}}{e^{-\bar{x}^2}} \\ &= M e^{-2\bar{x}y} e^{-y^2}, \end{aligned} \tag{D.1}$$

where M is a constant. By definition, essential infimum of $\frac{\pi(x)}{\bar{\pi}(x)}$ with respect to $\bar{\pi}$ measure is the greatest number a such that the set,

$$A = \{x \in \mathbb{R} : \frac{\pi(x)}{\bar{\pi}(x)} < a\}$$

has zero $\bar{\pi}$-measure. It is clear from (D.1) that, by choosing a large value for x, $\frac{\pi(x)}{\bar{\pi}(x)}$ can be made arbitrary close to 0 within a set of nonzero $\bar{\pi}$ measure.

$$\implies \operatorname{ess\,inf} \frac{\pi(x)}{\bar{\pi}(x)} = 0.$$

This shows that for $\bar{\pi}(\mathbf{x}) = \pi([\mathbf{x}])$, Independent Metropolis Hastings algorithm need not even be geometrically ergodic.

Bibliography

[1] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. 40th Annu. ACM Symp. Theory Comput.*, 2008, p. 197–206.

[2] C. Ling and J. Belfiore, "Achieving AWGN channel capacity with lattice Gaussian coding," *IEEE Trans. Inf. Theory*, vol. 60, no. 10, pp. 5918–5929, 2014.

[3] S. Vatedka, N. Kashyap, and A. Thangaraj, "Secure compute-and-forward in a bidirectional relay," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2531–2556, 2015.

[4] D. Aggarwal, D. Dadush, O. Regev, and N. Stephens-Davidowitz, "Solving the Shortest Vector Problem in 2^n time using discrete Gaussian sampling: Extended abstract," in *Proc. STOC*, 2015, p. 733–742.

[5] S. Liu, C. Ling, and D. Stehle, "Decoding by sampling: A randomized lattice algorithm for bounded distance decoding," *IEEE Trans. Inf. Theory*, vol. 57, no. 9, pp. 5933–5945, 2011.

[6] Z. Wang and C. Ling, "On the geometric ergodicity of Metropolis-Hastings algorithms for lattice Gaussian sampling," *IEEE Trans. Inf. Theory*, vol. 64, no. 2, pp. 738–751, 2018.

[7] Z. Wang, S. Lyu, and L. Liu, "Learnable Markov Chain Monte Carlo sampling methods for lattice Gaussian distribution," *IEEE Access*, vol. 7, pp. 87 494–87 503, 2019.

[8] S. Anaswara, "Sampling from multidimensional distributions supported on a lattice," Master's thesis, Indian Institute of Science, Bengaluru, 2020.

[9] P. Klein, "Finding the closest lattice vector when it's unusually close," in *Proc. ACM-SIAM Symp. Discrete Algorithms*, 2000, p. 937–941.

[10] C. Peikert, "An efficient and parallel Gaussian sampler for lattices," in *Advances in Cryptology*, T. Rabin, Ed. Springer Berlin Heidelberg, 2010, pp. 80–97.

[11] T. Prest, "Gaussian sampling in lattice-based cryptography," Ph.D. dissertation, Ecole normale supérieure - ENS PARIS, 2015.

[12] R. M. Neal, "MCMC using Hamiltonian dynamics," in *Handbook of Markov chain Monte Carlo*, S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, Eds. Chapman and Hall/CRC, 2011, p. 113–162.

[13] A. B. Tsybakov, *Introduction to Nonparametric Estimation*. Springer, 2008.

[14] S. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability*, 2nd ed. Cambridge University Press, 2009.

[15] G. O. Roberts and J. S. Rosenthal, "General state space Markov chains and MCMC algorithms," *Probab. Surveys*, vol. 1, pp. 20–71, 2004.

[16] R. Dwivedi, Y. Chen, M. J. Wainwright, and B. Yu, "Log-concave sampling: Metropolis-Hastings algorithms are fast," *J. Mach. Learn. Res*, vol. 20, no. 183, pp. 1–42, 2019.

[17] G. O. Roberts, A. Gelman, and W. R. Gilks, "Weak convergence and optimal scaling of random walk Metropolis algorithms," *Ann. Appl. Probab.*, vol. 7, no. 1, pp. 110–120, 1997.

[18] G. O. Roberts and J. S. Rosenthal, "Complexity bounds for MCMC via diffusion limits," 2014. [Online]. Available: https://arxiv.org/pdf/1411.0712.pdf

[19] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, Eds., *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, 2011.

[20] G. O. Roberts and R. L. Tweedie, "Exponential convergence of Langevin distributions and their discrete approximations," *Bernoulli*, vol. 2, no. 4, pp. 341–363, 1996.

[21] K. L. Mengersen and R. L. Tweedie, "Rates of convergence of the Hastings and Metropolis algorithms," *Ann. Statist.*, vol. 24, no. 1, pp. 101–121, 1996.

[22] S. Livingstone, M. Betancourt, S. Byrne, and M. Girolami, "On the geometric ergodicity of Hamiltonian Monte Carlo," *Bernoulli*, vol. 25, no. 4A, pp. 3109–3138, 2019.

[23] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, 3rd ed. Springer-Verlag, 1999.

[24] W. Ehm, T. Gneiting, and D. Richards, "Convolution roots of radial positive definite functions with compact support," *Trans. Am. Math. Soc*, vol. 356, no. 11, pp. 4655–4685, 2004.

[25] L. Tierney, "Markov chains for exploring posterior distributions," *Ann. Statist.*, vol. 22, no. 4, pp. 1701–1728, 1994.

[26] S. Chatterjee, P. Diaconis, and A. Sly, "Properties of uniform doubly stochastic matrices," 2010. [Online]. Available: https://arxiv.org/pdf/1010.6136.pdf

[27] Y. Chen, R. Dwivedi, M. J. Wainwright, and B. Yu, "Fast MCMC sampling algorithms on polytopes," *Journal of Machine Learning Research*, vol. 19, no. 55, pp. 1–86, 2018.

[28] A. Smith, "Some analyses of Markov chains by the coupling method," Ph.D. dissertation, Stanford University, 2015.

[29] R. Sinkhorn and P. Knopp, "Concerning nonnegative matrices and doubly stochastic matrices," *Pacific Journal of Mathematics*, vol. 21, no. 2, pp. 343 – 348, 1967.

[30] P. A. Knight, "The Sinkhorn–Knopp algorithm: Convergence and applications," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 1, pp. 261–275, 2008.

[31] M. Idel, "A review of matrix scaling and Sinkhorn's normal form for matrices and positive maps," 2016. [Online]. Available: https://arxiv.org/pdf/1609.06349.pdf

[32] M. Krishnapur and N. Kashyap, "Iteration to generate doubly stochastic matrices," *preprint.*

[33] R. Sinkhorn and P. Knopp, "Problems involving diagonal products in nonnegative matrices," *Transactions of the American Mathematical Society*, vol. 136, pp. 67–75, 1969.

[34] V. Cappellini, H. Sommers, W. Bruzda, and K. Życzkowski, "Random bistochastic matrices," *Journal of Physics A: Mathematical and Theoretical*, vol. 42, no. 36, 2009.

[35] B. Lemmens and R. Nussbaum, *Nonlinear Perron–Frobenius Theory*, ser. Cambridge Tracts in Mathematics. Cambridge University Press, 2012.

[36] A. Beskos, N. S. Pillai, G. O. Roberts, J. M. Sanz-Serna, and A. M. Stuart, "Optimal tuning of the hybrid Monte-Carlo algorithm," 2010.