Drawing Exact Samples: Rejection Sampling, Density Fusion and Constrained Disaggregation

Submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

by

Shenggang Hu

Department of Mathematical Sciences
University of Essex

October, 2022

DECLARATION

I here by declare that the thesis entitled “Drawing Exact Samples: Rejection Sampling, Density Fusion and Constrained Disaggregation” submitted by me, for the award of the degree of *Doctor of Philosophy in Statistics* to University of Essex is a record of bonafide work carried out by me under the supervision of Prof. Hongsheng Dai, Department of Mathematical Sciences, University of Essex.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Date: 05/10/2022 **Signature of the Candidate**

ABSTRACT

Sampling is an important topic in the area of computational statistics. Being able to draw samples from a designated distribution allows one to numerically compute various statistics without the need to solve for solutions analytically. A popular branch of the sampling method generates samples by evolving a stationary Markov chain that admits the target distribution as its stationary distribution. The problem, however, is that one does not have a universal criterion to assess whether the chain is stationary.

On the other hand, exact simulation methods, being the focus of this thesis, always produce samples that precisely follow the target distribution. We first begin with the path-space rejection sampling for the exact simulation of diffusion bridges and show how this rejection scheme can be further set up into an exact simulation method for sampling product densities. We provide guidance on how to tune the algorithm parameters in order to attain a near-optimal performance and introduce the construction of an importance sampler/particle filter based on the same theoretical result for better efficiency. Finally, we show a variant of the sampler that deals with linear constraints which render most of the target distributions intractable. Two application studies are conducted in the end to demonstrate the effectiveness of the algorithm.

Keywords: *constrained imputation*, *diffusion process*, *particle filter*, *perfect sampling*, *time series oversampling*.

ACKNOWLEDGEMENT

With immense pleasure and deep sense of gratitude, I wish to express my sincere thanks to my supervisor Prof. Hongsheng Dai. Without his motivation and continuous encouragement, this research would not have been successfully completed.

The lecture notes from "Advanced Simulation Methods" course written by Arnaud Doucet provided good inspiration on how to lay down the essential results from measure theory without getting too technical. My thanks to all the people who contributed to the polishing of the lecture notes (Pierre E. Jacob, Rémi Bardenet, George Deligiannidis, Lawrence M. Murray, Tigran Nagapetyan), and also to Patrick Rebeschini who lectured me the course in 2018.

I am more than lucky to be involved in the BIAS project and many thanks to all the members of the team, especially Lei Ding, Jabir A. Al-Ani and Yang Hu for your support and many productive discussions on the project.

Finally, I wish to express my gratitude to my family for all the sacrifices they made during my research and also providing me with moral support and guidance whenever I need.

CONTENTS

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Sampling and Monte Carlo Integration

Stochastic models allow some of the variables in the model to be non-deterministic, which makes the model capable of capturing, to some extent, the randomness that is often exhibited in real-world problems. We see stochastic models in various active areas including finance (Glasserman, 2004), operational research (Fishman, 2013), statistical mechanics (Binder, 1997), engineering (Bird, 1981, Mazhdrakov et al., 2018) and statistics (Robert et al., 1999). Stochastic models can be hard to solve analytically, especially when the dimension of the problem gets large.

Analyzing a stochastic model often results to computing the value of some function f that depends on a (perhaps d-dimensional) random variable $\boldsymbol{x}$ that follows a distribution $\pi(\boldsymbol{x})$. Since $\boldsymbol{x}$ is random, we often seek the expected value of $f(\boldsymbol{x})$ which is an integral of form

$$\mathbb{E}_\pi[f(\boldsymbol{x})] := \int_{\mathbb{X}} f(\boldsymbol{x})\pi(\boldsymbol{x})\mathtt{d}\boldsymbol{x}. \tag{1.1}$$

For simple choices of f and π, the integral will have an analytical answer. However, the problem can become intractable easily as the model complexity increases. In the occasions that f and π are analytically known, it is possible to numerically approximate the integral using quadrature methods (Davis and Rabinowitz, 2007) which work well in low-dimensions. However, traditional quadrature methods scale badly as the number of dimensions d climbs, with an approximation error in $\mathcal{O}(n^{-1/d})$.

In contrast, Monte Carlo methods for approximating (1.1) is based on drawing samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ from the distribution π and computing the average

$$I = \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i)$$

as the approximated value for the integral (1.1). Due to the Law of Large numbers, the approximation error is always in $\mathcal{O}(n^{-1/2})$ regardless of the dimension d. Due to this advantage, Monte Carlo methods may be favored even if traditional numerical methods are applicable to the problem.

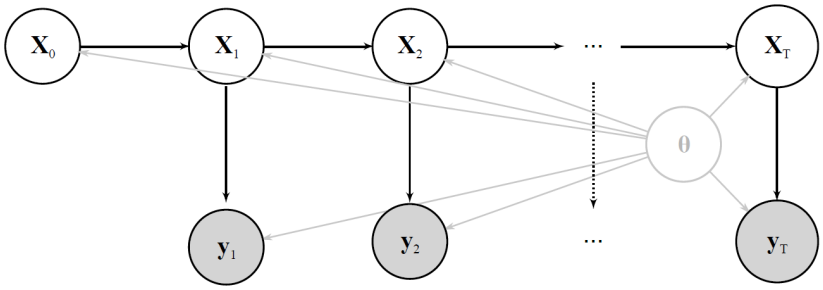


Fig. 1.1 Graphical representation of a typical Hidden Markov Model, Figure from Murray (2013).

In order for Monte Carlo methods to be implemented, one must be able to draw samples from the target distribution π which relies on the use of sampling algorithms. Sampling algorithms in computational statistics is the family of algorithms that generates random values such that the collection of generated values follows a certain distribution that one is interested in. In Monte Carlo methods, samples from distribution are used as a proxy for the true distribution with the advantage of working with concrete and deterministic values. Using these samples, one can visualize the distribution and approximate various statistics determined by that distribution which is often useful in complex models.

1.2 Applications of Monte Carlo Methods

1.2.1 Bayesian Inference

The hidden Markov model encompasses a sequence of evolving states X_t with prior $\pi(X_0)$ and transition density $p(X_t|X_{t-1})$, and the true values of X_t are unknown. The known values y_t are observations of Y_t conditioned on X_t modeled by likelihood function $f(y_t|X_t)$. Thus, the joint distribution of the system takes the form of

$$p(X_{0:t}, y_{1:t}) = \pi(X_0) \prod_{i=1}^{t} p(X_i|X_{i-1}) f(y_i|X_i)$$

and the posterior distribution of the hidden state is

$$p(X_{0:t}|y_{1:t}) = \frac{p(X_{0:t}, y_{1:t})}{\int_{\mathbb{X}} p(X_{0:t}, y_{1:t}) \mathtt{d}X_{0:t}}.$$

The posterior distribution quickly becomes intractable even for simple setups and thus Monte Carlo methods are required in order to numerically compute the normalizing constant and the posterior density.

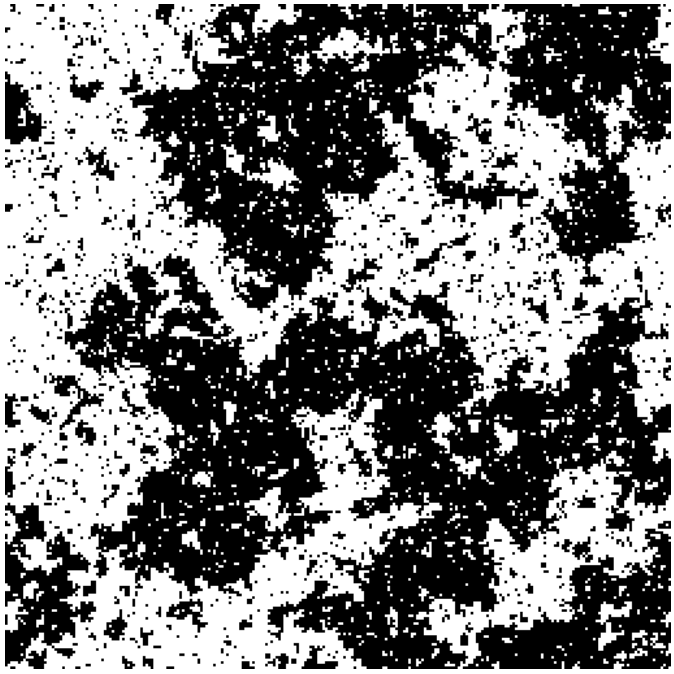


Fig. 1.2 A sample from the Ising model distribution. Figure generated at this website.

1.2.2 Ising Model

Ising model is a well-studied model in statistical mechanics that models the behavior of a magnet. The model consists of a lattice hosting particles at each site (crossing) and every particle has either a $+1$ or -1 spin. Consider a 2-d lattice with coordinates (i, j), $i, j \in \{1, \ldots, m\}$. Then the probability distribution on the state of all particles is defined on the state space $\mathbb{X} := \{-1, 1\}^{m^2}$ with distribution

$$\pi_\beta(\boldsymbol{x}) = \frac{1}{Z_\beta} \exp\left[-\beta \cdot J\left(\sum_{\sigma \sim \sigma'} x_\sigma x_{\sigma'}\right)\right], \quad \boldsymbol{x} \in \mathbb{X}$$

where

$$U(x) = J\left(\sum_{\sigma \sim \sigma'} x_\sigma x_{\sigma'}\right)$$

is the potential energy of the system and the summation is taken over all adjacent pairs of coordinates σ, σ' ($\sigma \sim \sigma'$), and Z_β is the normalizing constant. In order to compute the expected energy of the system $\mathbb{E}_{\pi_\beta}[U(X)]$ or the normalizing constant Z_β, one needs to sum over the 2^{m^2} possible states which is analytically very difficult. In most cases, simulation-based methods are used to approximate the sums.

Sampling Method	Exact	Generality	Info beyond pdf	Implementation
Inversion	exact	only in 1d	need cdf	distribution dependent
Transformation	exact	not generic	N/A	hard to design
Rejection	exact	generic	bound estimate M	hard for complex distributions
MCMC	non-exact	generic	not crucial	easy to implement
MCF	exact	generic	2nd derivative of pdf	easy with support of packages

Table 1.1 List of some most common sampling methods and their properties compared with MCF

1.3 Families of Sampling Methods

Usually, there is an efficient and exact sampling algorithm for every common distribution we work with on a daily basis, e.g., the uniform distribution, Gaussian distribution, binomial distribution, Poisson distribution, etc. These algorithms are designed in a way such that all samples generated by the algorithm are independent draws from the specified distribution, and usually, these algorithms are achieved through a combination of inversion, transformation, or rejection sampling. Such algorithms usually impede low computation costs due to the careful design of the execution. However, these algorithms are not generic, usually only work for one distribution family, and are usually hard to design for complicated target distributions.

Markov Chain Monte Carlo (MCMC) methods are perhaps the most used family of sampling methods when it comes to complex target distributions. The goal of an MCMC algorithm is to simulate a stochastic process called Markov Chain. The marginal distribution of the current state of the Markov chain will incrementally approach the target distribution. Since the iterative nature, the design/derivation of the update steps is not as challenging. A good example is the application of the Gibbs sampler to the Ising model. Gibbs sampler only relies on the conditional distribution of each X_σ given all other states $X_{-\sigma}$ which is easy to derive in the Ising model where every particle is only correlated to the particles adjacent to it.

MCMC methods also have their drawbacks. Firstly, the samples are generated from a distribution that asymptotically approaches the target distribution but not the target distribution itself. Moreover, each sample depends on the previous sample generated from the algorithm. Therefore, making use of the generated samples requires the user to distinguish whether the Markov chain has approached well enough to its converged state, unlike using exact algorithms where the samples can be directly used. To make the problem worse, it is not easy to spot whether the chain has converged or not, and assessments are usually done empirically without a universal criterion. Thus, it becomes apparent that there is a need for a generic exact sampling algorithm that is versatile enough to be applied on a wide family of distributions and is easy enough to implement

while, hopefully, being efficient enough to be practically implementable.

The Coupling From the Past (CFTP) algorithm (Propp and Wilson, 1996) received significant attention with various works on extending the algorithm (Huber, 1998, Murdoch and Green, 1998, Haggstrom and Nelander, 1999, Wilson, 2000, Mitha and Huber, 2012), see Djuric et al. (2002) for a more detailed review on CFTP or more recently Dai (2019). However, the algorithm is not very efficient when the state space is large and is significantly harder to implement on continuous state space. The diffusion simulation algorithm introduced in Beskos et al. (2006, 2008) demonstrates a design different from the previously existing rejection sampling framework which inspired the Monte Carlo Fusion (MCF) algorithm which is the main focus of this thesis. In later chapters, we will discuss various extensions of the MCF algorithm including a constrained variant. Constrained simulation is a difficult problem by itself and there is hardly any exact sampling algorithm that is applicable to a general distribution on a general constraint.

In this thesis, we will begin Chapter 2 by introducing, in different levels of detail, the first four methods listed in Table 1.1 accompanied by two other exact simulation methods (Propp and Wilson, 1996, Beskos et al., 2006). Then in Chapter 3, we will present the MCF algorithm from the work of Dai et al. (2019), followed by some discussion on tuning the parameters of the algorithm. In Chapter 4, we extend the MCF algorithm to deal with linear constraints, deriving a novel algorithm for sampling from a linearly constrained distribution. Following the constrained algorithm, we discuss two potential cases of application in Chapter 5 and put the algorithm to the test. Finally, we conclude in Chapter 6 with some future works.

1.4 Contributions

- Section 3.1.3 on the MCF algorithm taking different time parameters for different fusion components;
- Section 3.2 on some theoretical and simulation results on tuning the time parameters;
- Section 4.2 on deriving the constrained exact sampling algorithm;
- Section 4.3 on introducing an importance sampling/particle filter variant of the constrained algorithm based on the Poisson estimator (Beskos et al., 2006).
- Section 4.4 on analysing the effect of applying a linear constraint to prediction models compared with unconstrained models;
- Chapter 5 on two application studies of using the constrained algorithm to simulate samples from models with linear constraints.

CHAPTER 2

Perfect Sampling

Sampling, in this thesis, will be referring to the operation/algorithm inside a computer that obtains realizations of random variables that follow certain desired distributions. As the key step of Monte Carlo methods, sampling in a computer provides us the flexibility to work with distributions numerically rather than analytically. Indeed, Monte Carlo methods are often used when the target distribution is intractable analytically so that numerical approximations are inevitable, e.g., in Bayesian inference.

Perfect sampling, as mentioned in the chapter title, refers to sampling algorithms that produce exact samples from the target distribution. Perfect sampling and exact simulation are used interchangeably in this thesis. In the literature, the two terms may have been used to refer to the same goal but in different setups.

In order to talk about "Perfect Sampling", we need first to introduce in what sense some sampling methods are "imperfect". In this chapter, We will begin with some commonly used methods for generating random variables while comparing their differences, leading to the problem of "imperfectness". Finally, we will talk about a perfect sampling method that works on diffusion processes[1] rather than random variables, which will become the key ingredient for the rest of the works in this thesis.

2.1 Random Variables from Random Numbers

To keep the introduction concise, we shall avoid the discussion on the *source of randomness* in computers and assume here that we have access to a *(pseudo-)random number generator* that is good enough to produce samples that resemble realizations of an *uniformly distributed* random variable. In practice, we usually use such generators to produce draws from the Uniform distribution $\mathcal{U}[0,1]$, which leads to our first method for generating random variables.

[1] Since this is not a thesis in Measure theory, use of measure theory terminology and results, although inevitable, will be limited to minimal.

2.1.1 Inversion Method

Suppose our target is to generate samples for the random variable X that follows certain distribution $\mathcal{D}$ with cumulative distribution function (cdf) given by $F : \mathbb{R} \to [0, 1]$. The name of the "Inversion Method" comes from the operation of finding the inverse of the cdf $F^{-1}(u)$ to generate a sample of $X \sim \mathcal{D}$.

Algorithm 1: Inversion Method

Input : Inverse cumulative function $F^{-1} : [0, 1] \to \mathbb{R}$;

1 Generate $u \sim \mathcal{U}[0, 1]$;

2 Return $F^{-1}(u)$ as the sample.

Lemma 2.1.1. *Let X be a 1-d random variable follows certain distribution $\mathcal{D}$ with probability density function (pdf) f and cdf $F(x) := \int_{-\infty}^{x} f(t)\mathrm{d}t$. Let $F^{-1} : [0, 1] \to \mathbb{R}$ denote the inverse of F, then given $u \sim \mathcal{U}[0, 1]$,*

$$F^{-1}(u) \sim \mathcal{D}.$$

Proof. To prove that $F^{-1}(u) \sim \mathcal{D}$, we just need to verify that $\mathbb{P}(F^{-1}(u) \leq x) = F(x)$.

$$\begin{aligned} \mathbb{P}\left(F^{-1}(u) \leq x\right) &= \mathbb{P}\left(u \leq F(x)\right) \\ &= F(x), \end{aligned}$$

since $F(x) \in [0, 1]$ by definition and $u \sim \mathcal{U}[0, 1]$. □

Example 2.1.1 (Inversion method for Exponential distribution)**.** *Let $X \sim Exp(\lambda)$, then*

$$F(x) = \begin{cases} 1 - e^{-\lambda x}, & x \geq 0; \\ 0, & \textit{otherwise} \end{cases}$$

and

$$F^{-1}(u) = -\frac{\log(1 - u)}{\lambda}.$$

Since $1 - u$ also follows $\mathcal{U}[0, 1]$, we might as well treat $1 - u$ as u giving

$$x = -\frac{\log(u)}{\lambda}$$

as the output.

The Inversion method is a special case of the Transformation method where the process transforms independent samples of one distribution into independent samples of another distribution, in this case from Uniform to any target distribution with known

F^{-1}. Another example for the transformation method is we can transform a set of K Gamma distributions into a sample from the Dirichlet distribution.

The benefit of this type of method is that as long as our batches of input random variables are independent, we are guaranteed to obtain independent samples from the desired distribution. However, the problem is these algorithms are problem specific and for many complicated distributions, it is not easy to construct such algorithms through transformation.

2.1.2 Rejection Sampling

The Inversion method requires the knowledge of the cdf, namely the integral of pdf. In many cases, such an integral is not analytically known, not to say the method only works on 1-dimensional distributions. In contrast, rejection sampling only requires the knowledge of pdf and works on multidimensional distributions.

Let $\mathcal{D}$ be the target distribution with density function $f(x)$. Suppose that we cannot sample directly from $\mathcal{D}$, but we can sample from a proposal distribution with density function $q(x)$ such that there exists a constant M that $Mq(x) \geq f(x), \forall x \in \text{supp}(f)$. Then we can formulate the rejection sampling algorithm as follows:

Algorithm 2: Rejection Sampling

1 Sample from the proposal distribution $x \sim q(\cdot)$;
2 Sample $u \sim \mathcal{U}[0,1]$;
3 **if** $u \leq \frac{f(x)}{Mq(x)}$ **then**
4 | Accept and return x.
5 **else**
6 | Reject x and go back to Step 1.

Proposition 2.1.2. *Let $f : \mathbb{R}^d \to \mathbb{R}^{\geq 0}$ be the density function for the target d-dimensional distribution and let $g : \mathbb{R}^d \to \mathbb{R}^{\geq 0}$ be another density function to be used as proposal. Suppose there exists $M > 0$ such that $Mq(\boldsymbol{x}) \geq f(\boldsymbol{x})$, $\forall \boldsymbol{x} \in$ supp(f), then the random result $\boldsymbol{X}$ produced from the above rejection sampling algorithm follows the target density function f.*

Proof. It is enough to check the distribution of $\boldsymbol{X}$ given it is accepted. For any arbitrary $A \subset \mathbb{R}^d$,

$$\begin{aligned}
\mathbb{P}(\boldsymbol{X} \in A | \boldsymbol{X} \text{ accepted}) &= \frac{\mathbb{P}(\boldsymbol{X} \in A, \boldsymbol{X} \text{ accepted})}{\mathbb{P}(\boldsymbol{X} \text{ accepted})} \\
\mathbb{P}(\boldsymbol{X} \in A, \boldsymbol{X} \text{ accepted}) &= \int_A \int_0^{\frac{f(\boldsymbol{x})}{Mg(\boldsymbol{x})}} g(\boldsymbol{x}) \mathtt{d}u \mathtt{d}\boldsymbol{x} \\
&= \int_A \frac{f(\boldsymbol{x})}{M} \mathtt{d}\boldsymbol{x}
\end{aligned}$$

and the acceptance probability

$$\mathbb{P}\left(\boldsymbol{X} \text{ accepted}\right) = \int_{\mathbb{R}^d} \int_0^{\frac{f(\boldsymbol{x})}{Mg(\boldsymbol{x})}} g(\boldsymbol{x}) \mathtt{d}u \mathtt{d}\boldsymbol{x} = \frac{1}{M}.$$

Thus

$$\mathbb{P}\left(\boldsymbol{X} \in A | \boldsymbol{X} \text{ accepted}\right) = \int_A f(\boldsymbol{x}) \mathtt{d}\boldsymbol{x}$$

which ends the proof. □

Drawing on the elegant proof of Flury (1990), we may conclude the rejection sampling process as uniformly generating dots from the carved product space

$$\left\{(\boldsymbol{X}, y) \in \mathbb{R}^d \times [0, \infty) \middle| \, y \leq Mg(\boldsymbol{X})\right\}$$

and the dots are accepted if and only if they land inside the acceptance region

$$\left\{(\boldsymbol{X}, y) \in \mathbb{R}^d \times [0, \infty) \middle| \, y \leq f(\boldsymbol{X})\right\}$$

and return the $\boldsymbol{X}$ component of the dot. Similar to the outcomes of transformation methods, the samples obtained from rejection sampling are also independent and identically distributed (i.i.d.). Although more flexible than the transformation method, rejection sampling relies on the existence of the bounding constant M. More importantly, the acceptance rate is $1/M$ on average. In other words, the efficiency is highly dependent on the choice of proposal distribution, and finding a suitable proposal can be hard in some situations.

2.2 Asymptotically Exact — Markov Chain Monte Carlo Methods

The previous section lists two sampling methods that produce i.i.d. samples from the target distribution. However, this section is introduced only to make contrast with the previous "exact" algorithms to give the reader some idea of the "imperfectness" of MCMC algorithms. Therefore, we will only briefly introduce the concept of Markov chains and list some properties and results that make it suitable to be used in sampling algorithms. Meyn and Tweedie (2012) would be recommended for a more detailed introduction to Markov Chain. We will see later that MCMC methods are designed and executed differently from algorithms we have seen before, with the advantage of being even more flexible.

2.2.1 Basic Notions of Markov Chain

Definition 2.2.1 (Markov Chain)**.** *A discrete-time Markov chain is a collection of random variables $\{X_t \in \mathbb{X} | t \in \mathbb{N}\}$, where the measurable space $(\mathbb{X}, \mathcal{X})$ is the state space, that follows the Markov property*

$$\forall x \in \mathbb{X}, A \in \mathcal{X}, \text{ and time indices } s_1 < \cdots < s_n < t_0 < t$$
$$\mathbb{P}(X_t \in A | X_{s_1}, \ldots, X_{s_n}, X_{t_0} = x) = \mathbb{P}(X_t \in A | X_{t_0} = x) \quad (2.1)$$

For the purpose of this thesis, it suffices to consider only the time-homogeneous Markov chains, i.e.,

$$\mathbb{P}(X_{t+n} \in A | X_t = x) = \mathbb{P}(X_n \in A | X_0 = x), \ \forall x \in \mathbb{X}, A \in \mathcal{X}, t, n \in \mathbb{N}. \quad (2.2)$$

For time-homogeneous Markov chains, the evolution of the states X_t follows the distribution of X_{t+1} conditioned on $X_t = x$, which is completely governed by its ***transition kernel*** $K(\cdot, \cdot)$.

Definition 2.2.2 (Transition Kernel)**.** *A **transition kernel** $K : \mathbb{X} \times \mathcal{X} \to \mathbb{R}$ is a function such that*

1. *$\forall x \in \mathbb{X}$, $K(x, \cdot)$ is a measure on the state space $(\mathbb{X}, \mathcal{X})$*
2. *$\forall A \in \mathcal{X}$, $K(\cdot, A)$ is a measurable function.*

In the case where $\mathbb{X}$ is a continuous space, we use $K(x, x')$ to denote the transition density such that

$$\mathbb{P}(X_{t+1} \in A | X_t = x) = K(x, A) = \int_A K(x, x') \mathtt{d}x'.$$

2.2.2 Some Properties and Results

Definition 2.2.3 (Stationary Distribution)**.** *A probability measure π is said to be the **stationary distribution** of (the Markov chain equipped with) transition kernel K if*

$$\pi(A) = \int_{\mathbb{X}} \pi(x) K(x, A) \mathtt{d}x, \ \forall A \in \mathcal{X} \quad (2.3)$$

When a Markov chain admits π as its stationary distribution, we also call this Markov chain π-invariant.

A Markov chain is π-invariant implies when we start with $X_0 \sim \pi$, we will get $X_1 \sim \pi$, $X_2 \sim \pi$ and so forth. Such a chain can continuously produce samples from the fixed

distribution π. Under certain conditions, the π-invariant Markov chain can reach stationary state, i.e., the marginal distribution of X_n converges to π, as we evolve the chain. MCMC algorithms work based on the design that the target distribution π is attained as the invariant distribution of the Markov chain while the conditions for converges are also met (ergodic).

Definition 2.2.4 (Irreducibility)**.** *Let π be a measure on $(\mathbb{X}, \mathcal{X})$, a Markov chain with kernel K is **π-irreducible** if,*

$$\forall A \in \mathcal{X} \text{ such that } \pi(A) > 0, \exists\, n > 0 \text{ such that} K^n(x, A) > 0,\ \forall x \in \mathbb{X}.$$

Here K^n denotes the n-step transition function. A Markov chain is strongly π-irreducible if $n = 1$ for any $A \in \mathcal{X}$.

For a chain on a discrete state space to be irreducible, every state in the space must be accessible from every other state. In fact, how often a state (or a measurable set) is visited in the long run is a statistic for characterizing the Markov chain. Here we define the notion of η_A to denote the number of visits to set A by the chain X_t

$$\eta_A = \sum_{t=1}^{\infty} \mathbb{I}_A(X_t).$$

Definition 2.2.5 (Harris Recurrence)**.** *A set $A \in \mathcal{X}$ is **Harris recurrent** if $\mathbb{P}_x(\eta_A = \infty) = 1$ for all $x \in \mathbb{X}$. A Markov chain is **Harris recurrent** if it is π-irreducible and every set $A \in \mathcal{X}$ such that $\pi(A) > 0$ is Harris recurrent.*

Harris recurrence and π-invariant is enough to give a simple Ergodic Theorem, i.e., an MCMC equivalent of "Law of Large Numbers", for the convergence of Monte Carlo approximation to the true integral

$$\lim_{t\to\infty} \frac{1}{t} \sum_{k=1}^{t} h(X_t) \overset{\text{a.s.}}{=} \int_{\mathbb{X}} h(x)\pi(\mathtt{d}x).$$

However, for the convergence towards stationary distribution, we also need the chain to be aperiodic. Since the definition of periodicity in continuous space Markov chain is more involved than in discrete space, we will only give an abridged version of the definition similar to Roberts and Rosenthal (2004).

Definition 2.2.6 (Periodicity)**.** *A π-irreducible Markov chain with transition kernel K is periodic with period $d > 1$, if the state space $\mathbb{X}$ can be partitioned into d disjoint sets*

$\mathbb{X}_1, \ldots, \mathbb{X}_d$, *such that*

$$\forall i, j, t, s: \ \mathbb{P}(X_{t+s} \in \mathbb{X}_j | X_t \in \mathbb{X}_i) = \begin{cases} 1, & j \equiv i + s \mod d \\ 0, & \textit{otherwise} \end{cases}$$

Otherwise, the Markov chain is said to be aperiodic.

Note that it can be shown that if the Markov chain is periodic on a set $A \in \mathcal{X}$ with period d, then the state space can be partitioned into d disjoint sets leading to the definition above (Meyn and Tweedie, 2012, Robert et al., 1999). A Markov chain is periodic with period d means the state of the chain cycles through the d disjoint sets in a specific order.

Theorem 2.2.1 (Convergence to Stationary)**.** *If a Markov chain X_t with transition kernel K is π-invariant, Harris recurrent and aperiodic, then for every initiaion distribution λ*

$$\left|\left| \int \lambda(\mathtt{d}x) K^t(x, \cdot) - \pi \right|\right|_{TV} \to 0, \qquad t \to \infty.$$

This theorem was presented in Meyn and Tweedie (2012), Theorem 13.3.3. The result ensures that the produced samples, although they might be correlated, can be treated as samples from the target distribution π as $t \to \infty$.

2.2.3 Metropolis-Hastings Algorithm

Let f denote the target distribution, a generic Metropolis-Hastings algorithm operates on a proposal distribution $q(\cdot|\cdot)$ where $\forall x' \in \mathbb{X}$, $q(\cdot|x')$ is a density function on $\mathbb{X}$. Like the rejection sampling algorithm, at step $t + 1$ we generate a new proposal X from the proposal density and compute an acceptance probability

$$\alpha(X, X_t) = \min \left\{ 1, \frac{f(X) q(X_t|X)}{f(X_t) q(X|X_t)} \right\}. \tag{2.4}$$

The difference is when we reject the proposal, we will record again the previous value X_t, i.e., a new sample X_{t+1} is generated and equal to X_t.
We can prove the Markov chain in the Metropolis-Hastings algorithm admits the target distribution f as its invariant distribution, provided that the support of proposal q is good enough, by checking a condition called ***detailed balance***, though this part of the detail is omitted. Two primary concerns often arise when using MCMC algorithms like the Metropolis-Hastings:

- When will the Markov chain reach (close enough to) its stationary distribution, i.e., generating samples from the target distribution?

Algorithm 3: Generic Metropolis-Hastings Algorithm

Given starting point X_0;
for $t = 1, 2, \ldots$ **do**
  Sample X from $q(\cdot|X_{t-1})$;
  Compute acceptance probability $\alpha(X, X_{t-1})$ in (2.4);
  Generate $u \sim \mathcal{U}[0, 1]$;
  if $u < \alpha(X, X_{t-1})$ **then**
    Accept and set $X_t = X$;
  else
    Reject and set $X_t = X_{t-1}$.

- How many steps does it take for the correlation between samples to diminish, i.e., the value k such that X_t and X_{t+k} are almost independent?

The latter might be easier to answer and less problematic when the goal is to approximate some integral since correlation can be examined by checking the autocorrelation plot and the Ergodic theorems provide a convergence guarantee without requiring the samples to be independent. The former, however, has no generic answer to it. The need to alleviate such "imperfectness" leads to the search for perfect simulation methods based on MCMC techniques, e.g., the Coupling from the Past method.

2.2.4 Coupling from the Past

Algorithm 4: Coupling from the Past

Set $t = 0$;
Set $F_t^0 =$ identity map;
while $F_t^0(\cdot)$ *is not constant* **do**
  Simulate one-step transition mapping F_{t-1}^t;
  Compute $F_{t-1}^0 = F_{t-1}^t \circ F_t^0$;
  $t = t - 1$;
Return the singleton range of F_t^0.

The Coupling from the Past (CFTP) method (Propp and Wilson, 1996) keeps track of the Markov chain's evolution differently so that it is possible to realize when the chain has converged. The intuition is, if one could simulate the Markov chain for an infinite number of steps, then the chain must have converged (provided the ergodic conditions are met). In that case, the resulting state can be seen as an i.i.d. draw from the target distribution.

Here, for simplicity, we restrict ourselves to Markov chains defined on finite discrete space $\mathbb{X}$ of cardinality $|\mathbb{X}| := n$. Instead of evolving a single state X_t, the CFTP algorithm keeps track of the full transition mapping $F_{-t}^0 : \mathbb{X} \to \mathbb{X}$ from time $-t$ to 0.

Moving from time $-t$ to $-t-1$, we simulate a new transition mapping $F_{-t-1}^{-t} : \mathbb{X} \to \mathbb{X}$ by simulating the one-step Markov chain evolution for every state $x \in \mathbb{X}$ and record the images. As a matter of convenience, we sample and apply each new transition step from the left, so the time is counted backward.

$$F_{-t-1}^{0} := F_{-t-1}^{-t} \circ F_{-t}^{0}, \qquad F_{0}^{0} := \text{ identity map.}$$

When the Markov chain is f-irreducible and aperiodic, there exists T large enough such that F_{-T}^{0} is a constant mapping, i.e., $F_{-T}^{0}(x_1) = F_{-T}^{0}(x_2), \forall x_1, x_2 \in \mathbb{X}$. Then continuing the evolution backward no longer make any difference since any $M > T$, F_{-M}^{0} will also be a constant mapping. In this case, we call the chain has *coalesced* or *coalescence* has occurred and the singleton range of the coalesced mapping is an i.i.d. sample from the target distribution.

Remark 2.2.1. *The example algorithm only works on finite state space. For uncountable state spaces, coalescence would never happen under the current strategy. Extension to CFTP on uncountable state space requires implementing more sophisticated couplers, please see for instance, Murdoch and Green (1998), Mitha and Huber (2012).*

2.3 Perfect Sampling for Diffusion Processes

In the previous section we talked about Markov-chain-based sampling methods where the goal is to construct a Markov chain that admits the target distribution as its stationary distribution and aims to draw samples from the stationary chain. We ended that section with a perfect sampling method called CFTP.

In this section, we will move away from Markov chains and talk about perfect sampling for diffusion processes based on the rejection sampling algorithm introduced in Section 2.1.2. This section is based on the works of Beskos et al. (2005, 2006, 2008), we will go through the problem setup, theoretical proofs and give the algorithm framework before addressing the implementation difficulties.

Due to the length limit, it is impractical to include a rigorous introduction to diffusion process building from the basics of measure theory. We will select a few definitions and results that are necessary to lay down the theoretical basis for the rejection algorithm and keep the terminologies to a minimum, much like the introduction to Markov chains in the previous section.

2.3.1 Diffusion Processes, Itô's Lemma and Girsanov Theorem

A diffusion process is a Markov process, think of it as a continuous-time Markov chain, that satisfies some stochastic differential equations. Unlike what we have seen before,

the time indexing is now from an uncountable set. To start with, we shall give the definition of a broader family called ***stochastic processes***

Definition 2.3.1 (Stochastic process)**.** *Let $(\mathbb{X}, \mathcal{X})$ be a measurable space where the process will be defined on. A **stochastic process** is a collection of random variables*

$$X = \{X_t : 0 \leq t < \infty\} \text{ on } (\mathbb{X}, \mathcal{X}),$$

where for each t, $X_t : (\mathbb{X}, \mathcal{X}) \to (\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$.

Definition 2.3.2 (Brownian Motion)**.** *A d-dimensional **Brownian motion** $B = (B_t)_{t \geq 0}$ is a stochastic process that has the following properties:*

1. *For $0 \leq t_0 < t_1 < \cdots < t_n < \infty$, the increments $B_{t_0}, B_{t_1} - B_{t_0}, \cdots, B_{t_n} - B_{t_{n-1}}$ are independent;*
2. *For any $0 \leq s < t$, the increment $B_t - B_s$ follows a d-dimensional standard normal distribution, i.e., $B_t - B_s \sim \mathcal{N}_d(\mathbf{0}, \boldsymbol{I}_d)$;*
3. *The sample path $t \to B_t$ is continuous with probability one (almost surely).*

Here we quickly introduce the idea of induced measure. Let $X = (X_t)$ be a stochastic process defined on the measure space $(\mathbb{X}, \mathcal{X}, \mathbb{P})$. We may see X itself as a mapping $\mathbb{X} \to \mathbb{F}$ where $\mathbb{F}$ is the set of mappings on $[0, \infty) \to \mathbb{R}^d$ with a Borel algebra $\mathcal{B}(\mathbb{F})$. If X is a measurable mapping, then we may define the measure $\mathbb{Q}$ induced by the stochastic process X by $\mathbb{Q}(A) := \mathbb{P}(\{\omega : X(\omega) \in A\}), \forall A \in \mathcal{B}(\mathbb{F})$.

Definition 2.3.3 (Diffusion Process)**.** *A **diffusion process** $X = (X_t)_{t \geq 0}$ is a Markov process with a continuous sample path and satisfies some stochastic differential equations of the form*

$$\mathtt{d}X_t = b(X_t, t)\mathtt{d}t + \sigma(X_t, t)\mathtt{d}B_t$$

where B_t is a Brownian motion. When B_t is d-dimensional, $b(X_t, t)$ is a d-dimensional vector and $\sigma(X_t, t)$ is a $d \times d$ matrix.

Although stochastic integrals $\int \mathtt{d}B_t$ can be less trivial than the Riemann integrals $\int \mathtt{d}x$ we usually encounter, the notation $\mathtt{d}B_t$ also comes from the intuition of taking the limit of infinite sums to approximate an integral using summation. For the purpose of this thesis, it is enough to think the symbol $\mathtt{d}B_t$ to be analogous to the symbol $\mathtt{d}x$, except for seeing $(\mathtt{d}B_t)^2$ which becomes $\mathtt{d}t$.

Our target is to sample (the skeleton of) the diffusion process X from time 0 to T, driven by the equation

$$\mathtt{d}X_t = \alpha(X_t)\mathtt{d}t + \mathtt{d}B_t. \tag{2.5}$$

In order to set up a rejection sampling framework for this process, we need some tools from measure theory and stochastic integral.

Theorem 2.3.1 (Itô (1951))**.** *Let B_t be a 1-dimensional Brownian motion and $f(B_t, t)$ to be a twice-differentiable deterministic function. Then $X_t = f(B_t, t)$ satisfies*

$$\mathtt{d}X_t = \left(\frac{\partial f}{\partial t} + \frac{1}{2}\frac{\partial^2 f}{\partial B_t^2}\right)\mathtt{d}t + \frac{\partial f}{\partial B_t}\mathtt{d}B_t.$$

For simplicity, the formula is given in 1-d, but the extension to the multidimensional case comes naturally by considering the same Taylor expansion.

Another result we will use is the "change of measure" formula for diffusion processes (Girsanov, 1960). Like in the case of density functions, change of variable involves the knowledge of the derivative between the original variable and its reparameterization.

Theorem 2.3.2 (Radon-Nikodym Derivative)**.** *Let μ, ν be two σ-finite measures on the measurable space $(\mathbb{X}, \mathcal{X})$ such that ν is abusolutely continuous with respect to μ, i.e., $\mu(A) = 0 \implies \nu(A) = 0, \forall A \in \mathcal{X}$. Then there exists a measurable function $f : \mathcal{X} \to [0, \infty)$ such that*

$$\nu(A) = \int_A f \mathtt{d}\mu.$$

We call $\frac{\mathtt{d}\nu}{\mathtt{d}\mu} = f$ the Randon-Nikodym derivative between measures ν and μ.

See Durrett (2019) (p179, 406).

Theorem 2.3.3 (Girsanov's Formula)**.** *Consider two stochastic systems with equations*

$$\mathtt{d}X_t = \sigma(X_t, t)\mathtt{d}B_t + b(X_t, t)\mathtt{d}t$$

and

$$\mathtt{d}\tilde{X}_t = \sigma(X_t, t)\mathtt{d}B_t + \tilde{b}(X_t, t)\mathtt{d}t$$

where $b, \tilde{b}, \sigma$ are Lipschitz with respect to X_t. Let X_t and $\tilde{X}_t$, $t \in [0, T]$ be the corresponding solution to the SDEs and $\mathbb{Q}$, $\tilde{\mathbb{Q}}$ denote the measure induced by X and $\tilde{X}$ respectively. Then, $\tilde{\mathbb{Q}}$ is absolutely continuous with respect to $\mathbb{Q}$ with the Radon-Nikodym derivative between the measures given by

$$\frac{\mathtt{d}\tilde{\mathbb{Q}}}{\mathtt{d}\mathbb{Q}} = \exp\left[\int_0^T \phi(t)\mathtt{d}B_t - \frac{1}{2}\int_0^T |\phi(t)|^2 \mathtt{d}t\right] \tag{2.6}$$

where $\phi(t) = \sigma^{-1}(X_t, t)\left[\tilde{b}(X_t, t) - b(X_t, t)\right]$.

For a rigorous description of the result, see Theorem 3.4 in Chapter 7 of Friedman (1975) and the corresponding section.

Corollary 2.3.4. *Let $X = (X_t)_{t\in[0,T]}$ be a diffusion process driven by the Brownian motion B_t satisfying*

$$\mathtt{d}X_t = f(X_t, t)\mathtt{d}t + \mathtt{d}B_t,$$

Let $\mathbb{Q}$ denote the measure induced by the diffusion process X and $\mathbb{W}$ denote the measure induced by the Brownian motion B, then

$$\frac{\mathtt{d}\mathbb{Q}}{\mathtt{d}\mathbb{W}}(\omega) = \exp\left[\int_0^T f(\omega(t), t)\mathtt{d}B_t - \frac{1}{2}\int_0^T |f(\omega(t), t)|^2\mathtt{d}t\right] \tag{2.7}$$

where ω is a typical element of $C([0, T], \mathbb{R})$, the set of continuous mapping from $[0, T]$ to $\mathbb{R}$.

Proof. Apply Theorem 2.3.3 to the case where $\tilde{b} = f$, $b = 0$ and $\sigma = \mathbf{1}$. □

2.3.2 Rejection Sampling for Diffusion Process

The rejection sampling algorithm (Algorithm 2) we introduced earlier operates on density functions. Naturally, we see the generalization from finite-dimensional distributions to stochastic processes where we now identify the distribution based on the measure induced by the stochastic process. Firstly, we shall check that when we operate on measures instead of density functions, the acceptance probability will depend on the Radon-Nikodym derivative between the measures, analogous to the quotient between the density functions.

Lemma 2.3.5 (Rejection with Radon-Nikodym derivative)**.** *Let target measure $\mathbb{Q}$ and proposal measure $\mathbb{Z}$ be two probability measures defined on the same measurable space $(\mathbb{X}, \mathcal{X})$ with $\mathbb{Q}$ absolutely continuous with respect to $\mathbb{Z}$. Suppose that, for some finite constant $M > 0$*

$$\frac{\mathtt{d}\mathbb{Q}}{\mathtt{d}\mathbb{Z}}(\omega) = \frac{f(\omega)}{M} \qquad \mathbb{Z} - \textit{almost surely}.$$

with $f(\omega) \leq$, $\forall\omega \in \mathbb{X}$. Then the following rejection sampling algorithm produces samples from $\mathbb{Q}$.

Algorithm 5: Base Rejection Sampling

```
Draw sample ω ~ ℤ;
Draw sample u ~ 𝒰[0, 1];
Compute f(ω);
if u < f(ω) then
    accept ω.
else
    Reject ω.
```

Proof. It is enough to check that

$$\mathbb{P}(\omega \in A|\text{accepted}) = \mathbb{Q}(A).$$

$$\begin{aligned}\mathbb{P}(\omega \in A|\text{accepted}) &= \frac{\mathbb{P}(\omega \in A, u \leq f(\omega))}{\mathbb{P}(\text{accepted})} \\ &= \frac{\int_A f(\omega)\mathtt{d}\mathbb{Z}}{\int_{\mathbb{X}} f(\omega)\mathtt{d}\mathbb{Z}} \\ &= \frac{\int_A M\mathtt{d}\mathbb{Q}}{\int_{\mathbb{X}} M\mathtt{d}\mathbb{Q}} = \mathbb{Q}(A).\end{aligned}$$

□

2.3.2.1 Radon-Nikodym Derivative

Recall our target is to simulate a sample path for the time-homogeneous diffusion process X_t, $t \in [0, T]$ driven by the equation (2.5)

$$\mathtt{d}X_t = \alpha(X_t)\mathtt{d}t + \mathtt{d}B_t.$$

For simplicity, we consider the target diffusion to be 1-dimensional with unit diffusion.

Remark 2.3.1. *For a diffusion process with non-unit diffusion coefficient*

$$\mathtt{d}V_t = b(V_t, t)\mathtt{d}t + \sigma(V_t, t)\mathtt{d}B_t,$$

the transformed process $X_t = \int_0^t \frac{\mathtt{d}s}{\sigma(V_s,s)}$ *will have unit diffusion.*

In order for the Radon-Nikodym derivative (2.7) to exist, we assume the following conditions hold

Condition 1. *1.* $\alpha(\cdot)$ *is continuously differentiable;*

2. $\alpha^2 + \alpha'$ *is bounded below;*

3. Expression (2.7) is a martingale with respect to B_t.

Remark 2.3.2. *The third point in the above condition is just a technical requirement for applying Girsanov's formula (Theorem 2.3.3). Let*

$$M_T(\omega) := \exp\left[\int_0^T f(\omega(t), t)\mathtt{d}B_t - \frac{1}{2}\int_0^T |f(\omega(t), t)|^2\mathtt{d}t\right].$$

The condition is essentially requiring: $\mathbb{E}[|M_t|] \leq \infty$, $\forall t$, *and* $\mathbb{E}[M_t|\{B_\tau, \tau \leq s\}] = M_s$.

Fixing the starting point x and ending point y, we let $\mathbb{Q}_{T,x,y}$ denote the measure induced by the diffusion process X conditioned on $X_0 = x$ and $X_T = y$ and $\mathbb{W}_{T,x,y}$ be the measure induced by the corresponding Brownian bridge. Let $p_T(x,y)$ denote the transition density of X from $X_0 = x$ to $X_T = y$ and let $A_t = A(B_t) = \int_0^t \alpha(B_s)\mathtt{d}B_s$.

Theorem 2.3.6. *Let ω be a typical element of $\mathcal{C}([0,T],\mathbb{R})$. Under condition 1, the Radon-Nikodym derivative between $\mathbb{Q}_{T,x,y}$ and $\mathbb{W}_{T,x,y}$ is proportional to*

$$\frac{\mathtt{d}\mathbb{Q}_{T,x,y}}{\mathtt{d}\mathbb{W}_{T,x,y}}(\omega) = \frac{\mathcal{N}(y;x,T)}{p_T(x,y)} \cdot \exp\left\{A(y) - A(x) - \int_0^T \frac{1}{2}\left(\alpha^2 + \alpha'\right)(\omega(t))\mathtt{d}t\right\} \tag{2.8}$$

$$\propto \exp\left\{-\int_0^T [\phi(\omega(t))]\,\mathtt{d}t\right\} \leq 1 \tag{2.9}$$

where $\mathcal{N}(y;x,T)$ is the density function of normal variable $Y \sim \mathcal{N}(x,T)$ evaluated at $Y = y$ and

$$\phi(u) = \frac{1}{2}(\alpha^2(u) + \alpha'(u)) - l \tag{2.10}$$

for some lower bound l of function $\frac{1}{2}(\alpha^2(u) + \alpha'(u))$, $\forall u$.

Proof. Let $\mathbb{Q}$ and $\mathbb{W}$ be the unconditioned measure of $\mathbb{Q}_{T,x,y}$ and $\mathbb{W}_{T,x,y}$ respectively, then by Corollary 2.3.4

$$\begin{aligned}\frac{\mathtt{d}\mathbb{Q}_{T,x,y}}{\mathtt{d}\mathbb{W}_{T,x,y}}(\omega) &= \frac{\mathtt{d}\mathbb{Q}}{\mathtt{d}\mathbb{W}}(\omega) \cdot \frac{\mathcal{N}(y;x,T)}{p_T(x,y)} \\ &= \frac{\mathcal{N}(y;x,T)}{p_T(x,y)} \cdot \exp\left\{\int_0^T \alpha(\omega(t))\mathtt{d}B_t - \frac{1}{2}\int_0^T \alpha^2(\omega(t))\mathtt{d}t\right\}\end{aligned}$$

By Itô's Lemma (Thm 2.3.1)

$$\begin{aligned}\mathtt{d}A_t &= \left(\frac{\partial A}{\partial t} + \frac{1}{2}\frac{\partial^2 A}{\partial B_t^2}\right)\mathtt{d}t + \frac{\partial A}{\partial B_t}\mathtt{d}B_T \\ &= \frac{1}{2}\alpha'(B_t)\mathtt{d}t + \alpha(B_t)\mathtt{d}B_t \\ \implies \int_0^T \mathtt{d}A_t &= \int_0^T \frac{1}{2}\alpha'(B_t)\mathtt{d}t + \int_0^T \alpha(B_t)\mathtt{d}B_t \\ \implies \int_0^T \alpha(B_t)\mathtt{d}B_t &= A_T - A_0 - \int_0^T \frac{1}{2}\alpha'(B_t)\mathtt{d}t.\end{aligned}$$

Under the conditioned measure, $A_T = A(B_T) = A(y)$ and $A_0 = A(B_0) = A(x)$. Hence, with x, y fixed

$$\begin{aligned}\frac{\mathtt{d}\mathbb{Q}_{T,x,y}}{\mathtt{d}\mathbb{W}_{T,x,y}}(\omega) &= \frac{\mathcal{N}(y;x,T)}{p_T(x,y)} \cdot \exp\left\{A(y) - A(x) - \int_0^T \frac{1}{2}\left(\alpha^2 + \alpha'\right)(\omega(t))\mathtt{d}t\right\} \\ &\propto \exp\left\{-\int_0^T \left[\frac{1}{2}\left(\alpha^2 + \alpha'\right)(\omega(t)) - l\right]\mathtt{d}t\right\} \leq 1.\end{aligned}$$

for some lower bound l of $\alpha^2 + \alpha'$. □

2.3.2.2 Poisson Process Rejection

Given Theorem 2.3.6, we know how to construct, in theory, a rejection sampler for the diffusion process (2.5) with its two ends fixed using Algorithm 5. However, the function ϕ is intractable, for Algorithm 5 to be implementable on (2.9), we need a way to simulate the event $U \leq \exp\left\{-\int_0^T \phi(\omega(t))\mathtt{d}t\right\}$.

Definition 2.3.4 (Poisson Point Process)**.** *Let λ be a σ-finite measure on measurable space $(\mathbb{X}, \mathcal{X})$. A Poisson Point Process of intensity λ is a random countable subset Π of $\mathbb{X}$ such that*

(i) for any disjoint measurable sets $A_1, A_2, \ldots, A_n \in \mathcal{X}$, the random variables $N(A_1), \ldots, N(A_n)$ are independent;

(ii) for any $A \in \mathcal{X}$, the random variable $N(A)$ follows the Poisson distribution with intensity $\lambda(A)$.

Under suitable separability conditions on the space $(\mathbb{X}, \mathcal{X})$, the point process can be seen as a random collection of points $\{X_1, \ldots, X_\kappa\} \subset \mathbb{X}$ such that for any $A \in \mathcal{X}$, $N(A)$ denotes the number of points landing within the set A. More specifically, the probability of no point landing in A is $\exp(-\lambda(A))$. Using this property, we can simulate an event of (2.9).

Proposition 2.3.7. *Let $\omega \in C([0,T], \mathbb{R})$, Φ be a Poisson point process of intensity 1 on the space $[0,T] \times [0,M]$, where M is the upper bound of the function ϕ. Let $A := \{(t,u) \in [0,T] \times [0,M] : u \leq \phi(\omega(t))\}$ denote the region under the curve $\phi(\omega(t))$, then*

$$\mathbb{P}(N(A) = 0|\omega) = \exp\left\{-\int_0^T \phi(\omega(t))\mathtt{d}t\right\}$$

Proof. By the definition of Φ, the random variable $N(A)|\omega$ is Poisson with intensity $\int_A \mathtt{d}\lambda$ with λ is the Lebesgue measure on $[0,T] \times [0,M]$. Thus

$$\mathbb{P}(N(A) = 0|\omega) = \exp\left\{-\int_A \mathtt{d}\lambda\right\} = \exp\left\{-\int_0^T \phi(\omega(t))\mathtt{d}t\right\}.$$

□

From the proof, we require that ϕ is bounded both above and below. ϕ is bounded below is required in the condition 1 and is usually satisfied in practice. However, ϕ is often not bounded above. We will address this problem later in this chapter, so for now we assume that ϕ is also bounded above. Then we can rewrite Algorithm 5.

Algorithm 6: Rejection Sampling for Bounded ϕ

Input: Starting point x, ending point y, total time T;

1 Given T, M, simulate $\kappa \sim \text{Poi}(TM)$;
2 **for** $i = 1, \ldots, \kappa$ **do**
3 Simulate $t_i \sim \mathcal{U}[0, T]$;
4 Simulate $u_i \sim \mathcal{U}[0, M]$;
5 Store Poisson point process $\Phi := \{(t_1, u_1), \ldots, (t_\kappa, u_\kappa)\}$;
6 Simulate the skeleton of a Brownian bridge ω connecting $B_0 = x$ and $B_T = y$ at time points $0, t_1, \ldots, t_\kappa, T$;
7 compute the indicator

$$I := \prod_{i=1}^{\kappa} I[\phi(\omega(t_i)) < u_i]$$

8 **if** $I = 1$ **then**
9 Accept ω as a realization of $\mathbb{Q}_{T,x,y}$.
10 **else**
11 Otherwise reject ω.

Note that we cannot simulate the full Brownian bridge since its is a continuous stochastic process. However, to compute the indicator at step 3, it is enough to know the values of the Brownian bridge at the desired time points determined by the Poisson process Φ. Since the process Φ is independent of the realization of the Brownian bridge ω, we can simulate Φ before ω.

2.3.2.3 Properties of Brownian Motion/Bridge

Since it is impossible to simulate and store the full trajectory of a continuous stochastic process, it suffices to design algorithms that simulate the trajectory at requested time points $0 < t_1 < \cdots < t_n < T$.

Brownian Motion

Since Brownian motion satisfies Markov property and every increment $B_{t_j} - B_{t_{j-1}}$ are Gaussian random variables, simulating the Brownian motion on the skeleton points $0, t_1, \ldots, t_n, T$ can be done by sampling the step sizes incrementally.

Algorithm 7: Brownian Motion

Input: Time points $t_0 = 0, t_1, \ldots, t_n, t_{n+1} = T$;

1 Set $B_0 := 0$;
2 **for** $i = 1, \ldots, n+1$ **do**
3 Sample $h_i \sim \mathcal{N}(0, t_i - t_{i-1})$;
4 Set $B_i = B_{i-1} + h_i$;
5 Return the skeleton of Brownian motion $B := \{(t_i, B_i)_{i=0}^{n+1}\}$.

Brownian Bridge

Definition 2.3.5. *A standard Brownian bridge of time T is a Gaussian stochastic process $W = (W_t)_{t\in[0,T]}$ such that $\mathbb{E}[W_t] = 0, \forall t \in [0, T]$ and $\mathbb{E}[W_s W_t] = s - (ts/T)$, for $0 \leq s < t \leq T$.*

Lemma 2.3.8. *It is equivalent to seeing a standard Brownian bridge of time T as a Brownian motion B conditioned on $B_T = 0$.*

See, e.g., Theorem 1 Chow (2009).

Lemma 2.3.9. *A drifted Brownian bridge $W^{x,y}$ of time T is a Gaussian stochastic process such that $\mathbb{E}[W_t] = (1 - t/T)x + ty/T, \forall t \in [0, T]$ and $\mathbb{E}[W_s W_t] = s - (ts/T)$, for $0 \leq s < t \leq T$.*

Proof. By relocation invariant property of Brownian motion, we can see $W_t^{x,y}$ as the result of shifting the Brownian motion B_t to $B_t + x$ and condition on $B_T + x = y$. Using the Gaussian conditioning formula, we can derive the definition for a Brownian bridge of time T connecting x and y, i.e., compute $\mathbb{E}[W_t|W_T = y]$ and $\text{Cov}(W_s, W_t|W_T = y)$, the calculation is omitted. □

Proposition 2.3.10. *Let $(B_t)_{t\in[0,T]}$ be a Brownian motion, then*

$$W_t^{0,0} = B_t - \frac{t}{T} B_T$$

is a standard Brownian bridge of time T. More generally,

$$W_t^{x,y} = B_t - \frac{t}{T} B_T + (1 - t/T)x + (t/T)y$$

gives a Brownian bridge connecting x and y.

Proof. The first equality can be verified by checking the mean and covariance of $B_t - \frac{t}{T} B_T$, see, e.g., Theorem 5 of Chow (2009). The second equality is obtained by adding the mean function to the first equality. □

Algorithm 8: Brownian Bridge

Input: Starting point x, ending point y;
Time points $t_0 = 0, t_1, \ldots, t_n, t_{n+1} = T$;

1 Simulate a Brownian motion B of time T at time points $(t_i)_{i=1}^n$;
2 Set $W_0 := x$, $W_T := y$;
3 **for** $i = 1, \ldots, n$ **do**
4 $\quad W_{t_i} = B_{t_i} - \frac{t}{T} B_T + (1 - t/T)x + (t/T)y$;
5 Return $W := \{(t_i, W_{t_i})_{i=0}^{n+1}\}$.

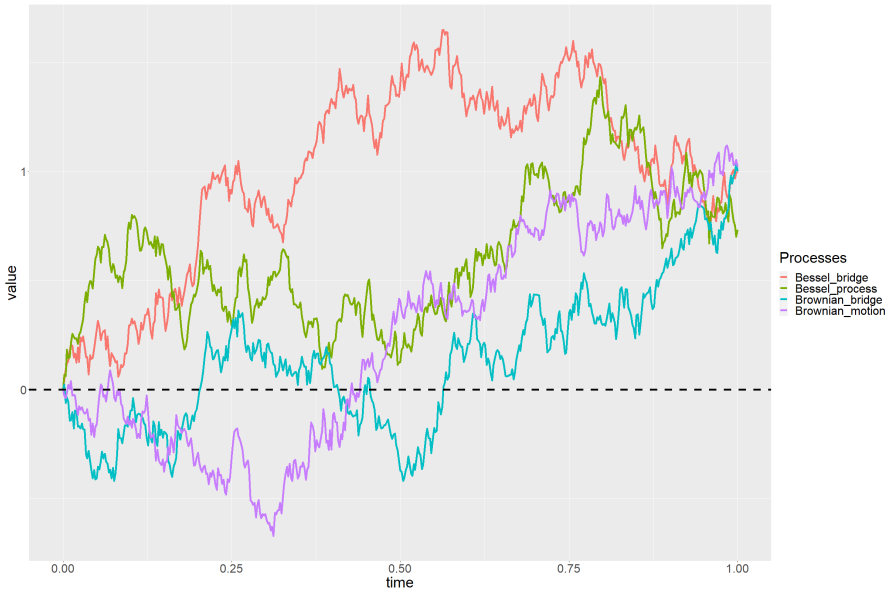


Fig. 2.1 Plots of the four different processes mentioned in this chapter. The Bessel bridge (in red) and the Brownian bridge (in cyan) are set to move from zero to one, while the Bessel process (in green) and the Brownian motion (in purple) only have the starting point set to zero. It is clear that the Bessel process and the Bessel bridge always stay above zero.

Bessel Bridge

Definition 2.3.6. *A d-dimensional **Bessel process** denoted by $R^d = \{(R_t)_{t\in[0,T]}\}$ is defined as*

$$R_t^{(d)} = \sqrt{\sum_{i=1}^{d}(B_t^{(i)})^2}$$

where $B^{(i)}$, $i = 1, \ldots, n$ are n independent 1-dimensional Brownian motions.

Informally, a 3-dimensional Bessel process is a Brownian motion constrained to be positive (Pitman, 1975) and a Bessel bridge R^y is a Bessel process R conditioned on the end point $R_T = y$.

Proposition 2.3.11. *A 3-dimensional Bessel bridge $BB^{T,0,y} = (BB_t^{T,0,y})_{t\in[0,T]}$ connecting 0 and $y > 0$ can be obtained by a transformation of three independent standard Brownian bridges $W^{(1)}, W^{(2)}, W^{(3)}$ of time T through*

$$BB_t^{T,0,y} = \sqrt{(yt/T + W_t^{(1)})^2 + (W_t^{(2)})^2 + (W_t^{(3)})^2}.$$

See, e.g., Bertoin and Pitman (1994).

Algorithm 9: Bessel Bridge

Input: Ending point y;
Time points $t_0 = 0, t_1, \ldots, t_n, t_{n+1} = T$;
Simulate three independent standard Brownian bridges, denoted W^1, W^2, W^3;
Set $X_0 = 0$;
for $i = 1, \ldots, n$ **do**
  $X_{t_i} = \sqrt{(yt_i/T + W^{(1)}_{t_i})^2 + (W^{(2)}_{t_i})^2 + (W^{(3)}_{t_i})^2}$;
Return Bessel bridge $X := \{(t_i, X_{t_i})_{i=0}^{n+1}\}$

2.3.2.4 Decomposition of Brownian Bridge at Its Minimum

Proposition 2.3.12. *Let $W^{x,y}$ be a Brownian bridge of time T connecting x and y. Let $\tau \in [0, T]$ be the time at which $W^{x,y}$ attains its minimum m. Then conditioned on τ, m, the process $(W^{x,y}_{\tau - t} - m)_{0 \le t \le \tau}$ is a Bessel bridge $BB^{\tau,0,x-m}$ and is independent of the process $(W^{x,y}_{t-\tau} - m)_{\tau < t \le T}$ which is a Bessel bridge $BB^{T-\tau,0,y-m}$.*

See, e.g., Proposition 2 of Asmussen et al. (1995).

Proposition 2.3.13. *The joint distribution of (τ, m) given x, y, T is*

$$\pi(m, \tau | x, y, T) \propto \frac{(m-x)(m-y)}{\sqrt{\tau^3 (T-\tau)^3}} \exp\left\{ -\frac{(m-x)^2}{2\tau} - \frac{(m-y)^2}{2(T-\tau)} \right\} \tag{2.11}$$

for $m \le \min\{x, y\}$ and $\tau \in [0, T]$.

This result appears many times in the literature on passage time of Brownian motion, see, e.g., Shepp (1979), Karatzas and Shreve (2012).

By Proposition 2.3.12 and 2.3.13, we come to a different algorithm (Algorithm 10) for simulating a Brownian bridge by first deciding its minimum point. The same routine can be reused for constructing a Brownian bridge with its maximum point decided first, see Algorithm 12 and 13.

Algorithm 10: Brownian bridge through its minimum

Input: Starting point x, ending point y;
Time points $t_0 = 0, t_1, \ldots, t_n, t_{n+1} = T$;

Simulate $(m, \tau) \sim \pi(m, \tau)$ from (2.11) using Algorithm 11;
Partition the time points into $\mathcal{T}_1 = \{t_1, \ldots, t_k\}$ and $\mathcal{T}_2 = \{t_{k+1}, \ldots, t_n\}$ such that $t_k < \tau < t_{k+1}$;
Simulate a Bessel bridge $BB^{\tau,0,x-m}$ on time points $\{\tau - t_k, \tau - t_{k-1}, \ldots, \tau - t_1\}$, denoted $(L_t)_{t\in[0,\tau]}$;
Simulate a Bessel bridge $BB^{T-\tau,0,y-m}$ on time points $\{t_{k+1} - \tau, \ldots, t_n - \tau\}$ denoted $(R_t)_{t\in[0,T-\tau]}$;
Construct the Brownian bridge W by setting

$$W_{t_i} = \begin{cases} L_{\tau - t_i}, & 0 \le i \le k \\ R_{t_i - \tau}, & k < i \le n+1 \end{cases}$$

set $W_\tau = m$;
return W.

Algorithm 11: Simulate time and minimum attained by a Brownian bridge

Input: Starting point x, ending point y, total time T;

Set $a = y - x$;
Draw $\lambda \sim \text{Exp}(1)$;
$b = (a - \sqrt{2T\lambda + a^2})/2$;
$c_1 = (a - b)^2/2T$;
$c_2 = b^2/2T$;
Draw $u \sim \mathcal{U}[0, 1]$;
if $u < (1 + \sqrt{c_1/c_2})^{-1}$ **then**
  Draw $V \sim \text{IGau}(\sqrt{c_1/c_2}, 2c_1)$;
else
  Draw $V \sim 1/\text{IGau}(\sqrt{c_2/c_1}, 2c_2)$;
Return pair $(b, T/(1 + V))$.

Algorithm 12: Brownian bridge through its maximum

Input: Starting point x, ending point y;
Time points $t_0 = 0, t_1, \ldots, t_n, t_{n+1} = T$;

Simulate $W^{-x,-y}$ on times points $\{t_i\}_{i=1,\ldots,n}$ using Algorithm 10;
Return the trajectory of $-W$.

Algorithm 13: Simulate time and maximum attained by a Brownian bridge

Input: Starting point x, ending point y, total time T;

Run Algorithm 11 with input $(-x, -y, T)$, denote the return values (m, τ);
Return $(-m, \tau)$.

2.3.3 Layered Construction for Brownian Bridges

Now we shall address the case where ϕ is not bounded above. The idea here is to decide first the compact interval $[a, b]$ where the Brownian bridge will stay and then compute the upper bound of ϕ given the interval, i.e., $\sup\{\phi(x) : x \in [a, b]\}$.

After the interval is decided, we may simulate the position of the maximum/minimum point of the Brownian bridge and recover the trajectory by decomposing the bridge into two Bessel bridges using either Algorithm 12 or 10.

Let $\mathtt{p}(T, x, y, K)$ denote the probability under $\mathbb{W}_{T,x,y}$ (the probability law induced by a Brownian bridge connecting $B_0 = x$ and $B_T = y$) that a trajectory does not leave the interval $[-K, K]$, $K > \max\{|x|, |y|\}$.

2.3.3.1 Some Auxiliary Results

For ease of explanation, we define the notion of ***alternating monotone***.

Definition 2.3.7. *A series $S = \sum b_k$ is **alternating monotone** if its partial sum S_n can be expressed as the sum of a positive sequence a_n*

$$S_n = a_0 + \sum_{r=1}^{n} (-1)^r a_r$$

where a_n is strictly decreasing, i.e., $0 < a_{r+1} < a_r$, $\forall r \geq 1$.

*A sequence is **alternating monotone** if it is the partial sum of an alternating monotone series.*

Lemma 2.3.14. *A sequence $(S_i)_{i\geq 1}$ is alternating monotone if and only if S_i is converging and $S_{2j-1} < S_{2j+1} < S_{2j+2} < S_{2j}$, $\forall j \geq 1$.*

Proof. The forward direction is trivial. Suppose that $(S_i)_{i\geq 1})$ is a converging sequence satisfying $S_{2j-1} < S_{2j+1} < S_{2j+2} < S_{2j}$, $\forall j \geq 1$. Then the subsequences $(S_{2j-1})_{j\geq 1}$ and $(S_{2j})_{j\geq 1}$ are strictly monotone converging sequences.

Let $S_0 = 0$, $a_{2j} = S_{2j} - S_{2j-1} > 0$ and $a_{2j-1} = S_{2j-2} - S_{2j-1} > 0$, $j \geq 1$. Then

$$S_n = 0 + \sum_{r=1}^{n} (-1)^r a_r.$$

Note that

$$a_{2j} - a_{2j-1} = S_{2j} - S_{2j-2} < 0, \qquad a_{2j+1} - a_{2j} = S_{2j-1} - S_{2j+1} < 0.$$

and the sequences $(d_j)_{j\geq 1} = (a_{2j})$, $(e_j)_{j\geq 1} = (a_{2j-1})$ both converge to zero. Hence, $(a_j)_{j\geq 1}$ is a positive monotone decreasing sequence converging to zero. □

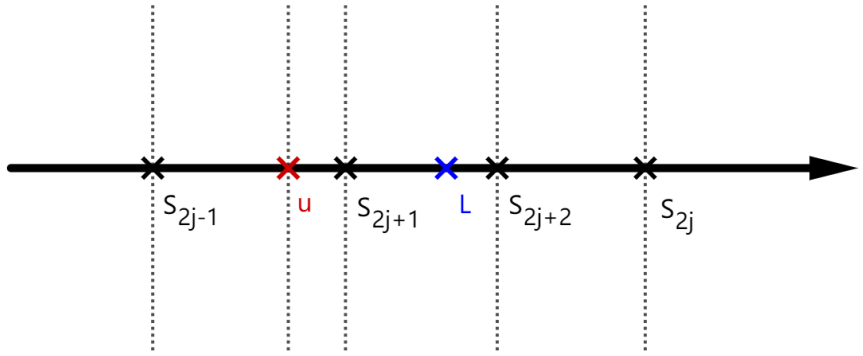


Fig. 2.2 The event where $u < L$ by identifying the fact that $U < S_{2j+1}$.

Remark 2.3.3. *An alternating monotone series is always convergent due to Leibniz's test.*

Proposition 2.3.15. *Let $(A_i)_{i\geq 1}$, $(B_i)_{i\geq 1}$ be two alternating monotone sequences, then the sequence $(C_i)_{i\geq 1}, C_i = \frac{A_i}{B_{i+1}}$ is also alternating monotone.*

Proof. Firstly, $(A_i)_{i\geq 1}$, $(B_i)_{i\geq 1}$ are both converging sequences, their quotient $(C_i)_{i\geq 1}$ is also convergent. By definition,

$$A_{2j-1} < A_{2j+1} < A_{2j+2} < A_{2j}, \quad B_{2j+1} < B_{2j+3} < B_{2j+2} < B_{2j}.$$

Therefore

$$\frac{1}{B_{2j}} < \frac{1}{B_{2j+2}} < \frac{1}{B_{2j+3}} < \frac{1}{B_{2j+1}}.$$

Grouping with A gives

$$\frac{A_{2j-1}}{B_{2j}} < \frac{A_{2j+1}}{B_{2j+2}} < \frac{A_{2j+2}}{B_{2j+3}} < \frac{A_{2j}}{B_{2j+1}}.$$

Hence, $C_{2j-1} < C_{2j+1} < C_{2j+2} < C_{2j}$, $\forall j \geq 1$ and Lemma 2.3.14 concludes the proof. □

Proposition 2.3.16. *Let S_n be alternating monotone with $S_n \to L$, as $n \to \infty$. Draw a uniform random variable $U \sim \mathcal{U}[0,1]$, knowing the sequence $S_1, S_2, S_3 \ldots$ is enough to compute the indicator function $I_{[U<L]}$ without knowing the exact value of L.*

Proof. By definition, there exists a positive monotone decreasing sequence a_n such that

$$S_n = a_0 + \sum_{r=1}^{n} (-1)^r a_r.$$

Thus

$$S_1 < S_3 < S_5 < \cdots < S_{2j-1} < \cdots < L < \cdots < S_{2j} < \cdots < S_6 < S_4 < S_2.$$

Given $U \neq L$, one of the two following cases will happen:

1. $\exists j > 0$, such that $U > S_{2j} > L$
2. $\exists j > 0$, such that $U < S_{2j-1} < L$.

Knowing the value of the sequence $S_1, S_2, \ldots$ is enough to evaluate $I_{[U<L]}$ and thus concludes the proof. □

From the above proposition, we can write down an algorithm (see also Fig. 2.2) for simulating the event of probability L where $L \in [0, 1]$ is the limit of an alternating monotone sequence S_n. The simulation strategy will become useful later when we will see that the crossing probability $\mathtt{p}(T, x, y, K)$ can only be expressed as an alternating monotone series.

Algorithm 14: Alternating Simulation of event $u < L$

Input : Function for computing $S_1, S_2, \ldots$;

```
Simulate u ~ U[0,1];
j = 1;
while True do
    Compute S_{2j-1}, S_{2j};
    if u < S_{2j-1} then
        Return True.
    if u > S_{2j} then
        Return False.
```

Output: The result of event with probability $L = \lim_{n\to\infty} S_n$.

2.3.3.2 Layered Partitioning of Path Space

Given a sequence of strictly increasing positive real numbers $\{a_i\}_{i\geq 1}$ and conditioned $X_0 = x$ and $X_T = y$ are known. Let $a_0 = 0$, $\bar{x} = \min\{x, y\}$ and $\bar{y} = \max\{x, y\}$, we define the following events:

$$U_i = \left\{ \sup_{0\leq t\leq T} X_t \in [\bar{y} + a_{i-1}, \bar{y} + a_i) \right\} \cap \left\{ \inf_{0\leq t\leq T} X_t > \bar{x} - a_i \right\},$$

$$L_i = \left\{ \inf_{0\leq t\leq T} X_t \in (\bar{x} - a_i, \bar{x} - a_{i-1}] \right\} \cap \left\{ \sup_{0\leq t\leq T} X_t < \bar{x} + a_i \right\},$$

$$D_i = U_i \cup L_i,\ i \geq 1.$$

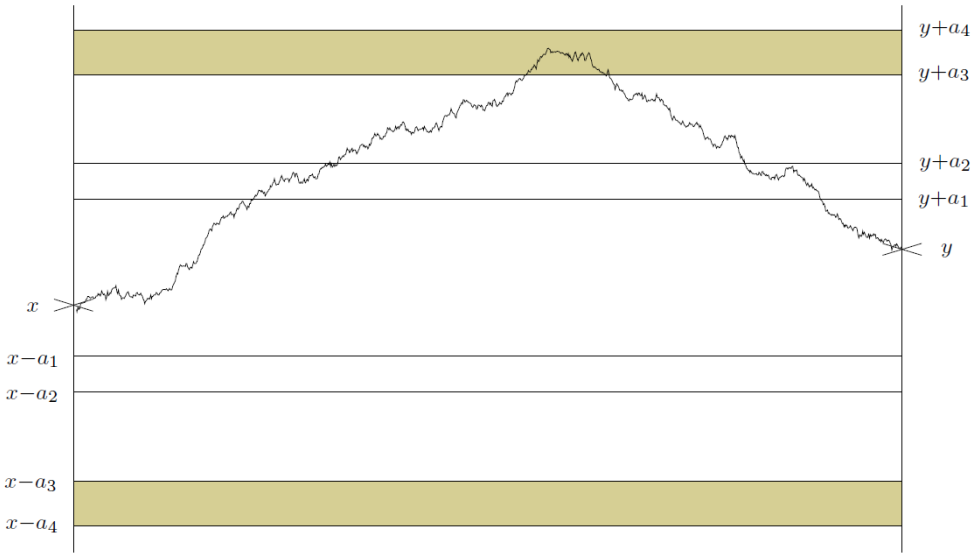


Fig. 2.3 A sample path from $X_0 = x$ to $X_T = y$ with $x < y$. The trajectory lands in the fourth layer, i.e., the event D_4 happened. (Beskos et al., 2008)

It is easy to check that $D_i \cap D_j = \emptyset$, $i \neq j$ and $\mathbb{W}_{T,x,y}(\cup_{i\geq 1} D_i) = 1$. Let I be the random variable such that $D_i = \{I = i\}$. Then

$$\{I \leq i\} = \{\bar{x} - a_i < X_t < \bar{y} + a_i,\ \forall t \in [0, T]\}.$$

Let $\mathrm{p}(T, x, y, K)$ denote the probability under $\mathbb{W}_{T,x,y}$ (the probability law induced by a Brownian bridge connecting $B_0 = x$ and $B_T = y$) that a trajectory does not leave the interval $[-K, K]$, $K > \max\{|x|, |y|\}$. Suppose that $\mathrm{p}(T, x, y, K)$ can be simulated, which in turn can be extended into an algorithm for simulating the event $\{I \leq i\}$ by using the relocation invariance of Brownian bridge. Then it is clear that given the ends x and y, we can sample the random variable I that determines which layer (or interval) our target trajectory will live in and derive an upper bound on ϕ.

Remark 2.3.4. *The original paper (Beskos et al., 2008) did not specify a searching algorithm for the layers. Here, we propose to use Algorithm 15 which implements a binary search that is more efficient than an ordinary sweep search.*

Remark 2.3.5. *The equal signs in the algorithm blocks (e.g., in Algorithm 15) always indicate the assignment of value on the right-hand side to the variable (in the computer memory) on the left-hand side. They* ***do not*** *represent equality, at least not before the value is assigned.*

Remark 2.3.6. *For ϕ that is unbounded, it is usually the case that a slight change to the interval will result in an exponential change to the upper bound M. Hence the interval increments $a_{i+1} - a_i$ should be tuned to a suitable magnitude, to keep the increment*

Algorithm 15: Binary Search on Layer I

Input: Starting point x, ending point y, total time T;
Layer step sequence $(a_i)_{i\geq 1}$;
Draw $u \sim \mathcal{U}[0,1]$;
head $= 1$, tail $= 2$;
while $u > \mathbb{P}(I \leq \textit{tail})$ **do**
    head $=$ tail;
    tail $= 2 \times$ tail;
while *head* $<$ *tail* $- 1$ **do**
    mid $= \lfloor (\text{head} + \text{tail})/2 \rfloor$;
    if $u < \mathbb{P}(I \leq \textit{mid})$ **then**
        tail $=$ mid;
    else
        head $=$ mid
Return tail.

in upper bound M suitably small. This could lead to the increase of $\mathbb{E}[I]$ adding extra computational cost.

2.3.3.3 Layer Probabilities

Layer Probability for Brownian Bridge

Recall that we use $\mathtt{p}(T,x,y,K)$ to denote the probability under $\mathbb{W}^{T,x,y}$ that the trajectory of the Brownian bridge does not leave interval $[-K,K]$, $K > \max\{|x|,|y|\}$.

Lemma 2.3.17. *Define for $j \geq 1$,*

$$\bar{\sigma}_j(T,x,y,K) = \exp\left\{-\frac{2}{T}[2jK-(K+x)][2jK-(K+y)]\right\}$$
$$\bar{\tau}_j(T,x,y,K) = \exp\left\{-\frac{2j}{s}[4jK^2+2K(x-y)]\right\}$$

and

$$\sigma_j(T,x,y,K) = \bar{\sigma}_j(T,x,y,K) + \bar{\sigma}_j(T,-x,-y,K)$$
$$\tau_j(T,x,y,K) = \bar{\tau}_j(T,x,y,K) + \bar{\tau}_j(T,-x,-y,K).$$

Then

$$\mathtt{p}(T,x,y,K) = 1 - \sum_{j=1}^{\infty}\{\sigma_j(T,x,y,K) - \tau_j(T,x,y,K)\}. \tag{2.12}$$

See, e.g., Theorem 3 of Pötzelberger and Wang (2001).

Proposition 2.3.18. *Omitting the function parameters,* p *is a alternating monotone se-*

ries of form

$$\mathtt{p} = 1 + \sum_{r=1}^{\infty} (-1)^r a_r$$

where

$$a_r = \begin{cases} \sigma_j & r = 2j-1 \\ \tau_j & r = 2j. \end{cases}$$

Proof. By (2.12), p indeed takes the form above, we just need to check $S_{2j} < S_{2j+2} < S_{2j+1} < S_{2j-1}$. With straight calculation,

$$\begin{aligned} \frac{\bar{\sigma}_j}{\bar{\tau}_j} &= \exp\left[\frac{2}{T}(K+x)(4jK-K-y)\right] > 1 \\ \frac{\bar{\sigma}_{j+1}}{\bar{\tau}_j} &= \exp\left[\frac{2j}{T}4K(x-K) - \frac{2}{T}(K-x)(K-y)\right] < 1. \end{aligned}$$

Thus, $\sigma_j > \tau_j > \sigma_{j+1}$ which leads to the result. □

Moreover, the probability of a Brownian bridge $W_t^{x,y}, t \in [0,T]$ not leaving the interval $[\min\{x,y\} - a_i, \max\{x,y\} + a_i]$ can be obtained by translating the path by $-\frac{1}{2}(x+y)$ and use the formula for p

$$F(i) := \mathbb{P}(I \le i) = \mathtt{p}\left(T, \frac{x-y}{2}, \frac{y-x}{2}, \frac{|x-y|}{2} + a_i\right), \quad i \ge 1. \tag{2.13}$$

With Algorithm 14, we can now simulate the probability of the path belonging to layer $I \le i$. Given $I = i$, the rest of the construction will roughly follow the steps below:

1. Decide whether the path belongs to layer i due to its maximum or its minimum;
2. Simulate the maximum (or minimum) and the time it is attained;
3. Simulate the skeleton of the path given the maximum (or minimum) and attained time using Algorithm 10 (or 12);
4. Verify the path between the skeleton points, which are Bessel bridges, indeed does not leave layer i.

For the last step, we also need to establish the boundary hitting probability for Bessel bridges.

Layer Probability for Bessel Bridge

To derive the boundary hitting probabilities, we use the property that the probability law of a Bessel bridge $BB_t^{x,y}, t \in [0,T]$ is identical to the probability law of a Brownian bridge $W^{x,y}$ conditioned on $W_t^{x,y} > 0, t \in [0,T]$, see Proposition 1.1 of Pitman (1975) for a more rigorous setup.

Remark 2.3.7. *This set of results is needed since we need to verify that the whole trajectory of the Brownian bridge we generated, effectively two connecting Bessel bridges, is contained in the interval decided at the start. Since we only see a finite set of points in trajectory, we can only verify the containment by computing boundary-hitting probability.*

Let $\mathtt{q}(T, x, y, K)$ denote the probability of a Bessel bridge $BB^{x,y}$ connecting $BB_0 = x \geq 0$, $BB_T = y > 0$ does not leave interval $(0, K)$ and $\mathtt{q}(T, x, y, K; L)$, $K < L$, denote the probability of $BB^{x,y}$ does not leave $(0, K)$ conditioned on it not leaving $(0, L)$. Then with simple derivation, we have

$$\mathtt{q}(T, x, y, K; L) = \frac{\mathbb{P}(BB_t^{x,y} \text{ does not leave } (0, K)}{\mathbb{P}(BB_t^{x,y} \text{ does not leave } (0, L)} = \frac{\mathtt{p}(T, x - K/2, y - K/2, K/2)}{\mathtt{p}(T, x - L/2, y - L/2, L/2)} \tag{2.14}$$

$$\mathtt{q}(T, x, y, K) = \lim_{L\to\infty} \mathtt{q}(T, x, y, K; L) = \frac{\mathtt{p}(T, x - K/2, y - K/2, K/2)}{1 - \exp\{-2xy/T\}}. \tag{2.15}$$

Note that $\mathtt{p}(T, 0, y, K) = 0$, though the limit $\mathtt{q}(T, 0, y, K; L)$ exists, by l'Hospital's rule,

$$\begin{aligned}\mathtt{q}(T, 0, y, K; L) = \lim_{x\to 0} \mathtt{q}(T, x, y, K; L) &= \lim_{x\to 0} \frac{\frac{\partial \mathtt{p}}{\partial x}(T, x - K/2, y - K/2, K/2)}{\frac{\partial \mathtt{p}}{\partial x}(T, x - L/2, y - L/2, L/2)} \\ &= \frac{y - \sum_{j=1}^{\infty} \{\zeta_j(T, y, K) - \xi_j(T, y, K)\}}{y - \sum_{j=1}^{\infty} \{\zeta_j(T, y, L) - \xi_j(T, y, L)\}}\end{aligned} \tag{2.16}$$

$$\begin{aligned}\mathtt{q}(T, 0, y, K) = \lim_{x\to 0} \mathtt{q}(T, x, y, K) &= \lim_{L\to\infty} \lim_{x\to 0} \frac{\frac{\partial \mathtt{p}}{\partial x}(T, x - K/2, y - K/2, K/2)}{\frac{\partial \mathtt{p}}{\partial x}(T, x - L/2, y - L/2, L/2)} \\ &= 1 - \frac{1}{y} \sum_{j=1}^{\infty} \{\zeta_j(T, y, K) - \xi_j(T, y, K)\}\end{aligned} \tag{2.17}$$

where

$$\zeta_j(T, y, K) = (2jK - y) \exp\left\{-\frac{2}{T} jK(jK - y)\right\}, \quad \xi_j(T, y, K) = \zeta_j(T, -y, K)$$

Proposition 2.3.19. *Fixing T, y, K. If $K > y > 0$ and $3K^2 - T > 0$, then, omitting the function arguments, the series $y - \sum_j \{\zeta_j - \xi_j\}$ is alternating monotone.*

Proof. Let

$$S_n(y) = y - \sum_{j=1}^{\infty} \{\zeta_j - \xi_j\} = y + \sum_{r=1}^{\infty} (-1)^r a_r(y)$$

where

$$a_{2j}(y) = \xi_j(T, y, K), a_{2j-1}(y) = \zeta_j(T, y, K).$$

$(S_n)_{n\geq 1}$ is alternating monotone if a_r is also monotone decreasing to 0. Obviously, $a_r \to 0$ as $r \to \infty$, we only need to check it is monotone decreasing.

Recall that $y \in (0, K)$.

1. Let
$$d_j(y) := \frac{a_{2j-1}}{a_{2j}}(y) = \frac{2jK - y}{2jK + y} \exp\left\{\frac{4}{T} jKy\right\}.$$
Note that $d_j(0) = 1$. The derivative is
$$\frac{\mathtt{d}d_j}{\mathtt{d}y}(y) = \frac{4jK}{(2jK + y)^2} \exp\left\{\frac{4}{T} jKy\right\} ((4j^2K^2 - y^2)/T - 1).$$
Thus if $3K^2 - T > 0$, we have $\frac{\mathtt{d}d_j}{\mathtt{d}y}(y) > 0$ and hence, $d_j(y) > 1, y \in (0, K)$.

2. Let
$$e_j(y) := \frac{a_{2j}}{a_{2j+1}}(y) = \frac{2jK + y}{2(j + 1)K - y} \exp\left\{\frac{2}{T}(2j + 1)K(K - y)\right\}.$$
Note that $e_j(K) = 1$. The derivative is
$$\frac{\mathtt{d}e_j}{\mathtt{d}y}(y) = \frac{2(2j + 1)K}{(2(j + 1)K - y)^2} \exp\left\{\frac{2}{T}(2j + 1)K(K - y)\right\} \times (1 - (2jK + y)(2(j + 1)K - y)/T).$$
If $8K^2 > T$, then $\frac{\mathtt{d}e_j}{\mathtt{d}y}(y) < 0$ and hence $e_j(y) > 1$.

□

Usually, T is chosen to be quite small in practise, so $3K^2 > T$ is easily achievable. Therefore by Proposition 2.3.19 and 2.3.15, the probability $\mathtt{q}$ in all four cases (2.14) to (2.17) are alternating monotone and thus can be simulated by Algorithm 14.

2.3.3.4 Layered Brownian Bridge

Putting the results we have so far, we can describe in detail how we can simulate a Brownian bridge conditioned on its layer I. Recall the sketched algorithm for simulating a layered Brownian bridge:

Algorithm 16: Algorithm Overview for Layered Brownian Bridge

Input: Starting point x, ending point y, total time T;
Layer step sequence $(a_i)_{i\geq 1}$, layer $I = i$;

1 Decide whether the path belongs to layer i due to its maximum or its minimum;
2 Simulate the maximum (or minimum) and the time it is attained;
3 Simulate the skeleton of the path given the maximum (or minimum) and attained time using Algorithm 10 (or 12);
4 Verify the path between the skeleton points, which are Bessel bridges, indeed does not leave layer i.

The reason for step 2 in Algorithm 16 is that simply simulating an unconditioned Brownian bridge W and rejecting it if $W \notin D_I$ is not efficient enough. In fact, the expected number of proposals to get an acceptance is infinite (see Beskos et al. (2008)). Thus, we opt to simulate first the maximum or minimum point of the Brownian bridge, which its range is decided by the layer I, then using the Bessel decomposition of the Brownian bridge (Step 4).

Let $\mathbb{W}_{D_I}$ denote the law of the Brownian bridge $\mathbb{W}_{T,x,y}$ restricted to the event D_I happened, i.e, the Brownian bridge $W^{x,y}$ belongs to layer I, which is our target measure for now. Consider the following events:

$$\bar{M}_i = \left\{ \sup_{0\leq t\leq T} W_t \in [\bar{y} + a_{i-1}, \bar{y} + a_i) \right\}, \underline{M}_i = \left\{ \inf_{0\leq t\leq T} W_t \in (\bar{x} - a_i, \bar{x} - a_{i-1}] \right\}, i \geq 1.$$

Similarly, let $\mathbb{W}_{\bar{M}_i}$, $\mathbb{W}_{\underline{M}_i}$ be the law $\mathbb{W}_{T,x,y}$ restricted to the corresponding event. Now, we rewrite Step 2 to wrap Algorithm 16 in a rejection sampling scheme.

Consider the following proposal measure $\mathbb{Q}_{D_I} = \frac{1}{2}\mathbb{W}_{\bar{M}_i} + \frac{1}{2}\mathbb{W}_{\underline{M}_i}$, i.e., with probability a half, we will simulate the Brownian bridge given its maximum point and with the other half probability, a Brownian bridge conditioned on its minimum will be generated.

Remark 2.3.8. *Note that $\mathbb{Q}_{D_I} \neq \mathbb{W}_{D_I}$, since the events U_I and L_I are not disjoint, so the event $(U_I \cap L_I) \subset D_I$ is counted twice under the scheme. Thus in execution, we need to recognize the case where $I_{U_I \cap L_I} = 1$ and reject the proposal bridge with probability 0.5.*

Algorithm 17: Refined Algorithm for Layered Brownian Bridge

Input: Starting point x, ending point y, total time T;
Layer step sequence $(a_i)_{i\geq 1}$, layer $I = i$;
1 Sample $u \sim \mathcal{U}[0,1]$;
2 **if** $u < 0.5$ **then**
3 Simulate the maximum and the time it is attained conditioned on layer $I = i$, i.e., from interval $[\bar{y} + a_{i-1}, \bar{y} + a_i)$;
4 Simulate the skeleton of the path X given the maximum and attained time using Algorithm 12;
5 Verify that $X \in D_I$, if not then reject;
6 **else**
7 Simulate the minimum and the time it is attained conditioned on layer $I = i$, i.e., from interval $(\bar{x} - a_i, \bar{x} - a_{i-1}]$;
8 Simulate the skeleton of the path X given the minimum and attained time using Algorithm 10;
9 Verify that $X \in D_I$, if not then reject;
10 **if** $X \in U_I \cap L_I$ **then**
11 Accept X with probability 0.5.
12 **else**
13 Accept X.

Steps 4,8 of Algorithm 17

Since simulating the maximum point of $W^{x,y}$ restricted to interval $[m, M]$ is equivalent to simulating the minimum point of $W^{-x,-y}$ restricted to interval $[-M, -m]$, thus it suffices to modify Algorithm 11 only. Note that from Algorithm 11, the minimum point $b = (a - \sqrt{2T\lambda + a^2})/2$ where $\lambda \sim \text{Exp}(1)$. The condition $b \in [m, M]$ is hence equivalent to $\lambda \in [2M(M-a)/T, 2m(m-a)/T]$. Since the cdf of Exponential distribution is known, we can use the inversion method to draw from the distribution restricted to some interval $[u, v]$, see also Example 2.1.1.

Example 2.3.1. *Let $\lambda \sim Exp(1)$, then the cdf $F(x) = 1 - e^{-x}$. To draw λ from interval $[c, d]$, we can draw an uniform random variable $u \sim \mathcal{U}[1 - e^{-c}, 1 - e^{-d}]$ and apply the preimage to get $\lambda = F^{-1}(u) = -\log(u)$.*

For numerical stability, we may consider drawing $u \sim \mathcal{U}[0,1]$ and get

$$\begin{aligned}\lambda &= -\log\left(e^{-c} - (e^{-c} - e^{-d})u\right) \\ &= -d + \log\left(e^{d-c} - (e^{d-c} - 1)u\right).\end{aligned}$$

Usually, the interval range $d - c$ will not be very large in simulation. For large $d - c$, one might need to use series expansion to approximate the preimage.

By replacing step 2 of Algorithm 11 with Algorithm 18, we can simulate the minimum point and time attained for a Brownian bridge conditioned on the layer it lies in. Then

Algorithm 18: Draw $\lambda \sim$Exp(1) from $[m, M)$

Input: Starting point x, ending point y, total time T, layer interval $[m, M)$

$a = y - x$;
$c = 2M(M - a)/T$;
$d = 2m(m - a)/T$;
Draw $u \sim \mathcal{U}[0, 1]$;
$\lambda = -d + \log\left(e^{d-c} - (e^{d-c} - 1)u\right)$;
Return λ.

based on the position of the minimum, the skeleton of the Brownian bridge can be simulated by piecing together two Bessel bridges based on Algorithm 11, the same applies to the maximum point case. Thus, so far, we have generated a path skeleton for the Brownian bridge where all the skeleton points obey the layer condition. Next, we need to check the unobserved trajectory, i.e., the trajectories between the skeleton points indeed obey the layer condition and the rejection scheme.

Verify $X \in D_I$

Since the two proposals (from maximum or from minimum) are quite symmetric, we will only derive the assertions for the minimum case and write down the maximum case directly. Recall that the event for the minimum point in layer I is

$$L_I = \left\{ \inf_{0 \le t \le T} X_t \in (\bar{x} - a_I, \bar{x} - a_{I-1}] \right\} \cap \left\{ \sup_{0 \le t \le T} X_t < \bar{x} + a_I \right\},$$

where $\bar{x} = \min\{x, y\}$, $\bar{y} = \max\{x, y\}$. Now, the paths between the skeleton points are all Bessel bridges that does not go below the minimum $m \in n(\bar{x} - a_I, \bar{x} - a_{I-1}]$. Thus, to verify the path $X \in D_i$, we only need to check that each Bessel bridge does not exceed $\bar{y} + a_I$. Suppose the path skeleton are generated on the time points $\mathcal{T} := \{0, t_1, \ldots, t_k, \tau, t_{k+1}, \ldots, t_n, T\}$, where τ is the time when the minimum m is attained. For simplicity, relabel the index set $\mathcal{T}$ into the ordered sequence $s_0, \ldots, s_{n+2}$, with $s_0 = 0$, $s_{n+2} = T$. With relocation invariance and boundary hitting probability $\mathtt{q}$, the probability of $X \in D_I$ is equal to

$$\prod_{i=1}^{n+2} \mathtt{q}(s_i - s_{i-1}, X_{s_{i-1}} - m, X_{s_i} - m, \bar{y} + a_I - m) \tag{2.18}$$

where each $\mathtt{q}(s_i - s_{i-1}, X_{s_{i-1}} - m, X_{s_i} - m, \bar{y} + a_I - m)$ is alternating monotone and can be simulated separately. Since we reject X if $X \notin D_i$, the path is rejected immediately if $\mathtt{q}(s_i - s_{i-1}, X_{s_{i-1}} - m, X_{s_i} - m, \bar{y} + a_I - m)$ failed for some i.

For the case of maximum, the inputs should be reflected with respect to zero which

gives

$$\prod_{i=1}^{n+2} \mathtt{q}(s_i - s_{i-1}, M - X_{s_{i-1}}, M - X_{s_i}, M - \bar{x} + a_I) \tag{2.19}$$

where M is the simulated maximum of the Brownian bridge.

Verify $X \in U_I \cap L_I$

Finally, if $X \in D_I$, we need to reject X with probability 0.5 if $X \in U_I \cap L_I$ is true to avoid double counting. In the case where the minimum m is simulated, the event $X \in L_I$ already happened due to $X \in D_I$. To check $X \in U_I$, we need to verify that

$$\sup_{t\in[0,T]} X_t \geq \bar{y} + a_{i-1}, \qquad \text{given that } X_t \leq \bar{y} + a_i.$$

Thus, the probability that $X \notin U_I$, i.e., $X \notin U_I \cap L_I$ is given by

$$\prod_{i=1}^{n+2} \mathtt{q}(s_i - s_{i-1}, X_{s_{i-1}} - m, X_{s_i} - m, \bar{y} + a_{I-1} - m; \bar{y} + a_I - m). \tag{2.20}$$

If $\mathtt{q}(s_i - s_{i-1}, X_{s_{i-1}} - m, X_{s_i} - m, \bar{y} + a_{I-1} - m; \bar{y} + a_I - m)$ failed for some i, i.e., $X \in U_I \cap L_I$, we can stop checking the remaining terms and accept X with probability 0.5. Otherwise, X is accepted.

For the case of maximum, $X \notin L_I$ happens with probability

$$\prod_{i=1}^{n+2} \mathtt{q}(s_i - s_{i-1}, M - X_{s_{i-1}}, M - X_{s_i}, M - \bar{x} + a_{I-1}; M - \bar{x} + a_I). \tag{2.21}$$

Thus, every step of Algorithm 17 can now be implemented, and Algorithm 6 can be extended to lift the restriction on ϕ being bounded. For a general time-homogeneous diffusion process driven by equation (2.5)

$$\mathtt{d}X_t = \alpha(X_t)\mathtt{d}t + \mathtt{d}B_t. \tag{2.5}$$

that satisfies condition 1 can now be simulated with Algorithm 19.

Algorithm 19: Path-Space Rejection Sampling

Input: Starting point x, ending point y, total time T;
Layer step sequence $(a_i)_{i\geq 1}$;

Decide the layer $I = i$ the path belongs to using Algorithm 14 and 15 based on (2.13);
Compute the upperbound $M > \phi(u)$, for $u \in [\bar{x} - a_i, \bar{y} + a_i]$;
Simulate $\kappa \sim \text{Poi}(TM)$;
for $i = 1, \ldots, \kappa$ **do**
    Simulate $t_i \sim \mathcal{U}[0, T]$;
    Simulate $u_i \sim \mathcal{U}[0, M]$;
Store Poisson point process $\Phi := \{(t_1, u_1), \ldots, (t_\kappa, u_\kappa)\}$;
Simulate the skeleton of a Brownian bridge ω connecting $B_0 = x$ and $B_T = y$ at time points $0, t_1, \ldots, t_\kappa, T$ conditioned on layer $I = i$ using Algorithm 17;
compute the indicator

$$I := \prod_{i=1}^{\kappa} I[\phi(\omega(t_i)) < u_i]$$

if $I = 1$ **then**
    Accept ω as a realization of $\mathbb{Q}_{T,x,y}$.
else
    Otherwise reject ω.

CHAPTER 3

Monte Carlo Fusion

At the end of the last chapter, we introduced an exact sampling algorithm for generating diffusion processes. In this chapter, we will look at the problem of sampling from a product density, where the target distribution is of form

$$f(\boldsymbol{y}) \propto \prod_{i=1}^{m} f_i(\boldsymbol{y}). \tag{3.1}$$

If the $\boldsymbol{y}^{(i)}$ are allowed to be different and each f_i is easy to sample from, it is also easy to obtain a sample from the product density f. Therefore, if we can construct a stochastic process $(X_t^{(i)})_{t\in[0,T]}$ such that one can maintain the process at its stationary state and, at the same time, the marginal distribution of the endpoint $X_T^{(i)} \sim f_i$. If we somehow apply the condition that all endpoints coalesce, we can obtain a sample from the target distribution $f(\boldsymbol{y})$. Fortunately, the Langevin process is a family of diffusion processes that admits stationary distributions, which will almost act as a continuous-time analog of a Markov chain, with the difference that the target distribution will not become its stationary distribution. The latter step is possible since we can decide the starting and ending points prior to the generation of the path skeleton in path-space rejection sampling (Algorithm 19). Due to Dai et al. (2019), we will use the name ***Monte Carlo Fusion*** to call the family of algorithms that samples from distributions like (3.1) through simulation of diffusion process such that the marginal distribution of the end point is the target distribution.

3.1 Langevin Diffusion and Density Fusion

Unless indicated otherwise, the subscripts s are usually reserved for the time index, and the bracketed superscripts (i) are usually reserved for the component index.

3.1.1 Basic Notions

Definition 3.1.1 (Langevin Diffusion)**.** *Let $A : \mathbb{R}^m \to \mathbb{R}$ be a differentiable function. The **Langevin diffusion** $(\boldsymbol{X}_t)_{t\geq 0}$ is a diffusion process satisfying the Langevin equation*

$$\mathtt{d}\boldsymbol{X}_t = \boldsymbol{\nabla} A(\boldsymbol{X}_t)\mathtt{d}t + \sigma \mathtt{d}\boldsymbol{B}_t,$$

where $(\boldsymbol{B}_t)_{t\geq 0}$ is an m-dimensional Brownian motion and $\sigma > 0$.

The Langevin diffusions appear as the continuous-time analog of ergodic Markov chains, in the sense that, under suitable conditions, these processes also admit stationary distributions (Hansen, 2003).

Theorem 3.1.1. *Let $\pi : \mathbb{R}^m \to \mathbb{R}$ is a proper density function and suppose that $\boldsymbol{\nabla} \log \pi(\boldsymbol{x})$ is continuously differentiable. Then the Langevin diffusion $(\boldsymbol{X}_t)t \geq 0$ satisfying*

$$\mathtt{d}\boldsymbol{X}_t = \frac{1}{2}\boldsymbol{\nabla} \log \pi(\boldsymbol{X}_t)\mathtt{d}t + \mathtt{d}\boldsymbol{B}_t$$

is π-invariant.

See Theorem 2.1 of Roberts and Tweedie (1996).

Using Langevin diffusion, we can construct a stationary process much like the stationary Markov chain that the MCMC algorithms are pursuing. However, due to the extra terms in (2.8) in Theorem 2.3.6, we do not simulate a process that is f^*-invariant.

Proposition 3.1.2. *Let $(\boldsymbol{X}_t)_{t\in[0,T]}$ be a stochastic process that is f_i^2-invariant, with transition function $p_i(\boldsymbol{y}|\boldsymbol{x})$ from $\boldsymbol{X}_0 = \boldsymbol{x}$ to $\boldsymbol{X}_T = \boldsymbol{y}$. Consider a biased process $(\bar{\boldsymbol{X}}_t)_{t\in[0,T]}$ such that the probability law induced by $\bar{\boldsymbol{X}}_t$ and $\boldsymbol{X}_t$ are the same conditioned on their starting and ending points $(\boldsymbol{x}, \boldsymbol{y})$. The bias is imposed so that the joint distribution of $(\bar{\boldsymbol{X}}_0, \bar{\boldsymbol{X}}_T)$ is*

$$\bar{p}_i(\boldsymbol{x}, \boldsymbol{y}) = f_i^2(\boldsymbol{x})p_i(\boldsymbol{y}|\boldsymbol{x})\frac{1}{f_i(\boldsymbol{y})}. \tag{3.2}$$

Then, the marginal distribution of $\bar{\boldsymbol{X}}_T$ is $f_i(\cdot)$.

Proof. Since $\boldsymbol{X}_t$ is f_i^2-invariant,

$$\int_{\mathcal{X}} f_i^2(\boldsymbol{x})p_i(\boldsymbol{y}|\boldsymbol{x})\mathtt{d}\boldsymbol{x} = f_i^2(\boldsymbol{y}),$$

and hence the result. □

Consider the following d-dimensional Langevin diffusion processes $(\boldsymbol{X}_t^{(i)})_{t\in[0,T]}$, $i = 1, \ldots, m$ driven by equations

$$\mathtt{d}\boldsymbol{X}_t^{(i)} = \boldsymbol{\nabla} A_i(\boldsymbol{X}_t^{(i)})\mathtt{d}t + \mathtt{d}\boldsymbol{B}_t^{(i)},$$

where $A_i(\boldsymbol{x}) = \log f_i(\boldsymbol{x})$. Now, the process $\boldsymbol{X}_t^{(i)}$ is f_i^2-invariant. Furthermore, if we construct a biased version of $\boldsymbol{X}_t^{(i)}$, denoted $(\bar{\boldsymbol{X}}_t^{(i)})_{t\in[0,T]}$, as in Proposition 3.1.2. Then the marginal distribution of $\bar{\boldsymbol{X}}_T$ is $f_i(\cdot)$. Similarly, the joint distribution of $\left(\bar{\boldsymbol{X}}_0^{(1:m)}, \bar{\boldsymbol{X}}_T^{(1:m)}\right)$ conditioned on $\bar{\boldsymbol{X}}_T^{(i)} = \bar{\boldsymbol{X}}_T^{(j)} = \boldsymbol{y}, \forall i \neq j$ has density

$$g(\bar{\boldsymbol{X}}_0^{(1)} = \boldsymbol{x}^{(1)}, \ldots, \bar{\boldsymbol{X}}_0^{(m)} = \boldsymbol{x}^{(m)},\ \boldsymbol{y}) = \prod_{i=1}^{m}\left[f_i^2(\boldsymbol{x}) p_i(\boldsymbol{y}|\boldsymbol{x}) \frac{1}{f_i(\boldsymbol{y})}\right] \tag{3.3}$$

and the marginal distribution of $\boldsymbol{y}$ is exactly the target distribution f in (3.1).

3.1.2 Construction of Proposal Diffusion

In path-space rejection sampling, we used a Brownian bridge as the proposal diffusion conditioned on the two ends of the bridge, we computed the Radon-Nikodym derivative between the measure induced by the target diffusion bridge and the Brownian bridge. We seek to do the same in this case. Reusing the notations in Chapter 1, we use $\mathbb{Q}_{T,\boldsymbol{x},\boldsymbol{y}}^{(i)}$ to denote the measure induced by the unbiased diffusion bridge $\boldsymbol{X}^{(i)}$ connecting $\boldsymbol{X}_0 = \boldsymbol{x}$ and $\boldsymbol{X}_T = \boldsymbol{y}$ where $\boldsymbol{x}, \boldsymbol{y}$ are fixed values. Similarly, $\mathbb{W}_{T,\boldsymbol{x},\boldsymbol{y}}^{(i)}$ denote the measure induced by a d-dimensional Brownian bridge connecting $\boldsymbol{B}_0 = \boldsymbol{x}$ and $\boldsymbol{B}_T = \boldsymbol{y}$.

Condition 2. *Let $\boldsymbol{\alpha}_i(\boldsymbol{u}) = \boldsymbol{\nabla} A_i(\boldsymbol{u})$.*

1. $$M_T := \exp\left\{\int_0^T \boldsymbol{\alpha}_i(\boldsymbol{\omega}_t)\cdot \mathtt{d}\boldsymbol{\omega}_t - \int_0^T \frac{1}{2}\|\boldsymbol{\alpha}_i(\boldsymbol{\omega}_t)\|^2 \mathtt{d}t\right\}$$
   is a martingale with respect to the Brownian motion $\boldsymbol{B}_t$.

2. *$\boldsymbol{\alpha}_i$ is continuously differentiable in all its arguments.*

3. *The function*
   $$\phi_i(\boldsymbol{u}) = \frac{1}{2}\left[\|\boldsymbol{\alpha}_i(\boldsymbol{u})\|^2 + \boldsymbol{\nabla}\cdot\boldsymbol{\alpha}_i(\boldsymbol{u})\right] - l_i \geq 0,$$
   for some l_i and for any $\boldsymbol{u}$, where $\boldsymbol{\nabla}\cdot$ is the divergence operator.

Proposition 3.1.3. *Under Condition 2, the transition function $p_i(\boldsymbol{y}|\boldsymbol{x})$ can be expressed with respect to the measure $\mathbb{W}_{T,\boldsymbol{x},\boldsymbol{y}}^{(i)}$*

$$p_i(\boldsymbol{y}|\boldsymbol{x}) = \frac{f_i(\boldsymbol{y})}{f_i(\boldsymbol{x})}\cdot \mathcal{N}_d(\boldsymbol{y}; \boldsymbol{x}, T\boldsymbol{I}_d)\cdot \mathbb{E}_{\mathbb{W}_{T,\boldsymbol{x},\boldsymbol{y}}^{(i)}}\left[\exp\left\{-\int_0^T (\phi_i(\boldsymbol{\omega}_t) + l_i)\,\mathtt{d}t\right\}\right], \tag{3.4}$$

where $\mathcal{N}_d$ represents the density function of a multivariate normal distribution of dimension d.

Proof. Recall from Theorem 2.3.6, (2.8) that

$$\begin{aligned}\frac{\mathtt{d}\mathbb{Q}_{T,\boldsymbol{x},\boldsymbol{y}}}{\mathtt{d}\mathbb{W}^{(i)}_{T,\boldsymbol{x},\boldsymbol{y}}}(\omega) =& \frac{\mathcal{N}_d(\boldsymbol{y};\boldsymbol{x},T\boldsymbol{I}_d)}{p_i(\boldsymbol{y}|\boldsymbol{x})}\exp\left\{A_i(\boldsymbol{y})-A_i(\boldsymbol{x})-\int_0^T\frac{1}{2}\left(\|\boldsymbol{\alpha}_i\|^2+\boldsymbol{\nabla}\cdot\boldsymbol{\alpha}\right)(\omega(t))\mathtt{d}t\right\}\\ =& \frac{\mathcal{N}_d(\boldsymbol{y};\boldsymbol{x},T\boldsymbol{I}_d)}{p_i(\boldsymbol{x},\boldsymbol{y})}\frac{f_i(\boldsymbol{y})}{f_i(\boldsymbol{x})}\exp\left\{-\int_0^T(\phi_i(\boldsymbol{\omega}_t)+l_i)\mathtt{d}t\right\}\end{aligned}$$

Rearrange and integrate both sides gives the expression we want. □

Now simplifying the expression of (3.3) gives

$$\begin{aligned}g(\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(m)},\boldsymbol{y}) =& \prod_{i=1}^m\left[f_i^2(\boldsymbol{x}^{(i)})p_i(\boldsymbol{y}|\boldsymbol{x}^{(i)})\frac{1}{f_i(\boldsymbol{y})}\right]\\ =& \left[\prod_{i=1}^m f_i(\boldsymbol{x}^{(i)})\right]\cdot(\sqrt{2\pi T})^{-md}\exp\left(-\frac{1}{2T}\sum_{i=1}^m\|\boldsymbol{y}-\boldsymbol{x}^{(i)}\|^2\right)\\ &\cdot\mathbb{E}_{\boldsymbol{x}^{(i)},\boldsymbol{y}}\left[\prod_{i=1}^m\exp\left\{-\int_0^T(\phi_i(\omega_t^{(i)})+l_i)\mathtt{d}t\right\}\right]\end{aligned}$$

The third term is the Radon-Nikodym derivative we dealt with in the previous chapter, which corrects the Brownian bridge proposal into the target diffusion bridge. Although the proposal has to be a Brownian bridge, we have the freedom to choose the distribution of $\boldsymbol{x}^{(i)}$ and $\boldsymbol{y}$. The first term above tells us to draw $\boldsymbol{x}^{(i)}\sim f_i(\cdot)$, so consider the following proposal distribution for $(\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(m)},\boldsymbol{y})$

$$h(\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(m)},\boldsymbol{y}) = \prod_{i=1}^m f_i(\boldsymbol{x}^{(i)})\mathcal{N}(\boldsymbol{y}|\bar{\boldsymbol{x}},(T/m)\boldsymbol{I}_d)$$

where $\bar{\boldsymbol{x}}=\frac{1}{m}\sum_{i=1}^m\boldsymbol{x}^{(i)}$. The quotient between g and h is proportional to

$$\frac{g(\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(m)},\boldsymbol{y})}{h(\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(m)},\boldsymbol{y})}\propto\exp\left[-\frac{\sum_{i=1}^m\|\boldsymbol{x}^{(i)}-\bar{\boldsymbol{x}}\|^2}{2T}\right]\cdot\mathbb{E}_{\boldsymbol{x}^{(i)},\boldsymbol{y}}\left[\prod_{i=1}^m\exp\left\{-\int_0^T\phi_i(\omega_t^{(i)})+l_i\mathtt{d}t\right\}\right], \tag{3.5}$$

where both terms can be corrected through rejection sampling, see Algorithm 20.

Remark 3.1.1 (Tuning parameter T)**.** *Note the two terms in (3.5) varies differently with T. The first rejection term*

$$\exp\left[-\frac{\sum_{i=1}^m\|\boldsymbol{x}^{(i)}-\bar{\boldsymbol{x}}\|^2}{2T}\right]\to 0 \quad \textit{as} \quad T\to 0$$

but the second term

$$\exp\left\{-\sum_{i=1}^m\int_0^T\phi_i(\omega_t^{(i)})+l_i\mathtt{d}t\right\}\to 1 \quad \textit{as} \quad T\to 0.$$

Algorithm 20: Monte Carlo Fusion

Input: Time parameter T;
for $i = 1, \dots, m$ **do**
  Simulate $\boldsymbol{x}^{(i)} \sim f_i(\cdot)$;
Compute $\bar{\boldsymbol{x}} = \frac{1}{m}\sum_i \boldsymbol{x}^{(i)}$;
Simulate $\boldsymbol{y} \sim \mathcal{N}\left(\bar{\boldsymbol{x}}, \frac{T}{m}\boldsymbol{I}_d\right)$;
Simulate $u_1 \sim \mathcal{U}[0,1]$;
if $\log u_1 \leq -\sum_i \|\boldsymbol{x}^{(i)} - \bar{\boldsymbol{x}}\|^2/2T$ **then**
  for $i = 1, \dots, m$ **do**
    Simulate, with one attempt only, the path skeleton for the diffusion bridge corresponding to measure $\boldsymbol{Q}^{(i)}_{T,\boldsymbol{x}^{(i)},\boldsymbol{y}}$ using Algorithm 19;
    if *failed* **then**
      Go back to Step 1;
  Accept and output $\boldsymbol{y}$.

In practice, the choice of T can drastically affect the efficiency of the algorithm and it is always worth the time tuning the parameter carefully before running a simulation. Usually, a binary search or grid search is enough for choosing a single T.

3.1.3 Fusion with Variable Time

An intuitive example where the fusion density problem appears is in meta-analysis (Fleiss, 1993, Smith et al., 1995, Scott et al., 2016), where multiple studies on a common subject, e.g., the effectiveness of a treatment, under different experimental conditions, i.e., results not directly fusible, are aggregated. Typically, each study will give a belief on the subject, in the form of a density f_i, and the aggregated belief will be the product density taking the form of (3.1). Naturally, some beliefs f_i will be closer to the final product than the other, which means the samples $\boldsymbol{x}^{(i)} \sim f_i(\cdot)$ will be closer to the final sample $\boldsymbol{y} \sim f(\cdot)$ on average. Therefore, the tuning parameter T for those components should be smaller for better efficiency. In this section, we will extend the original algorithm and show that using different time parameters for each component is indeed possible. Due to the increased complexity of tuning the parameters, we also propose an adaptive way of choosing the parameters later in this chapter.

For each density function f_i define a d-dimensional Langevin diffusion process $\boldsymbol{X}^{(i)} := \{\boldsymbol{X}_t^{(i)} : t \in [0, T_i]\}$ given by

$$\mathtt{d}\boldsymbol{X}_t^{(i)} = \boldsymbol{\nabla} A_i(\boldsymbol{X}_t^{(i)})\mathtt{d}t + \mathtt{d}\boldsymbol{B}_t^{(i)},$$

where T_i is now a random variable with probability density $\tau_i(t)$ independent of $\boldsymbol{X}^{(i)}$. Again, $\boldsymbol{X}^{(i)}$ has invariant distribution $\propto f_i^2(\boldsymbol{x})$ over $[0, T_i]$, given T_i being independent of $\boldsymbol{X}^{(i)}$. Let $p_i(\boldsymbol{y}|\boldsymbol{x}, t_i)$ denote the transition density from $\boldsymbol{X}_0 = \boldsymbol{x}$ to $\boldsymbol{X}_{t_i} = \boldsymbol{y}$.

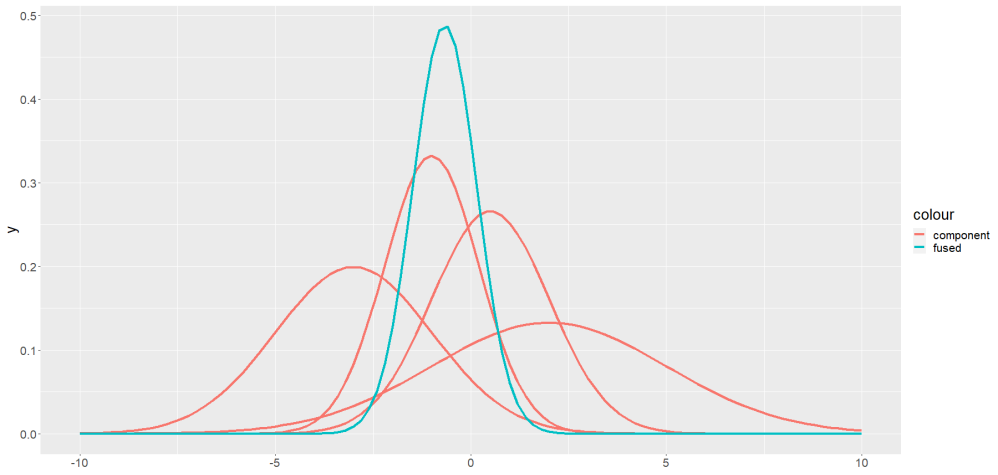


Fig. 3.1 Fusing four individual densities and the resulting product density.

Proposition 3.1.4. *With $\boldsymbol{T} = (T_1, \cdots, T_m)$ taking values $\boldsymbol{t} = (t_1, \cdots t_m)$, the following conditional density*

$$g\left(\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(m)}, \boldsymbol{y} \middle| \boldsymbol{t}\right) \propto \prod_{i=1}^{C} f_i^2(\boldsymbol{x}^{(i)}) p_i(\boldsymbol{y}|\boldsymbol{x}^{(i)}, t_i) \frac{1}{f_i(\boldsymbol{y})} \tag{3.6}$$

admits the marginal density f for $\boldsymbol{y}$ with any given $\boldsymbol{t}$. Further, with any density function $\tau(\boldsymbol{t})$, the joint density $g(\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(m)}, \boldsymbol{y}|\boldsymbol{t})\tau(\boldsymbol{t})$ admits the marginal density f for $\boldsymbol{y}$.

Corollary 3.1.5. *Under Condition 2, the transition density from $\boldsymbol{x}$ at 0 to $\boldsymbol{y}$ at t_i is given by*

$$p_i(\boldsymbol{y}|\boldsymbol{x}^{(i)}, t_i) = \frac{f_i(\boldsymbol{y})}{f_i(\boldsymbol{x}^{(i)})} \cdot \left(\frac{1}{\sqrt{2\pi t_i}}\right)^d \exp\left(-\frac{\|\boldsymbol{y} - \boldsymbol{x}^{(i)}\|^2}{2t_i}\right) \cdot \mathbb{E}_{\mathbb{W}^{(i)}_{T,\boldsymbol{x},\boldsymbol{y}}}\left[-\int_0^{t_i} (\phi_i(\boldsymbol{\omega}_t) + l_i)\, \mathrm{d}s\right].$$

Proof. Since t_i are fixed, the result follows directly from Proposition 3.1.3. □

By taking

$$\begin{aligned} h(\boldsymbol{x}^{(1)}, \cdots \boldsymbol{x}^{(m)}, \boldsymbol{y}|\boldsymbol{t}) &= (2\pi)^{-\frac{d}{2}} \left(\sum_{i=1}^{m} t_i^{-1}\right)^{\frac{d}{2}} \cdot \prod_{i=1}^{m} h_i(\boldsymbol{x}^{(i)}, \boldsymbol{y}|t_i) \\ h_i(\boldsymbol{x}, \boldsymbol{y}|t_i) &= f_i(\boldsymbol{x}) \exp\left[-\frac{\|\boldsymbol{y} - \boldsymbol{\mu}\|^2}{2t_i}\right] \end{aligned} \tag{3.7}$$

where $\boldsymbol{\mu}$ is a function of $\boldsymbol{x}$, we will have the following theorem.

Theorem 3.1.6. *Under Condition 2, with given $T_i = t_i, i = 1, \cdots, C$, we have*

$$\begin{aligned} &\frac{g(\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(m)}, \boldsymbol{y}|\boldsymbol{t})}{h(\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(m)}, \boldsymbol{y}|\boldsymbol{t}) \exp(-\sum_{i=1}^m l_i t_i)} \\ = &\left(\prod_{i=1}^m t_i \sum_{j=1}^m t_j^{-1}\right)^{-\frac{d}{2}} \underbrace{\prod_{i=1}^m \left(\frac{\exp\left[-\frac{\|\boldsymbol{x}^{(i)}-\boldsymbol{y}\|^2}{2t_i}\right]}{\exp\left[-\frac{\|\boldsymbol{y}-\boldsymbol{\mu}\|^2}{2t_i}\right]}\right)}_{\rho(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{t})} \cdot \mathbb{E}\left[\exp\left(-\sum_{i=1}^m \int_0^{t_i} \phi_i(\boldsymbol{\omega}_s^{(i)})\mathrm{d}s\right)\right] \end{aligned} \tag{3.8}$$

Proof. The Theorem follows directly from Corollary 3.1.5. □

Corollary 3.1.7. *By choosing*

$$\boldsymbol{\mu} = \frac{1}{W}\sum_{i=1}^m \frac{1}{t_i}\boldsymbol{x}^{(i)}, \tag{3.9}$$

where

$$W = \sum_{i=1}^m \frac{1}{t_i}.$$

The term $\rho(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{t})$ in equation (3.8) simplifies into

$$\prod_{i=1}^m \exp\left[-\frac{1}{2t_i}\|\boldsymbol{x}^{(i)} - \boldsymbol{\mu}\|^2\right] \tag{3.10}$$

Proof. Note that

$$\log \rho(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{t}) = -\frac{1}{2}\sum_{i=1}^m \frac{1}{t_i}\left[\|\boldsymbol{x}^{(i)} - \boldsymbol{y}\|^2 - \|\boldsymbol{y} - \boldsymbol{\mu}\|^2\right]$$

The result holds directly from the following identities.

$$\begin{aligned} &\sum_{i=1}^m \frac{1}{t_i}\left[\|\boldsymbol{x}^{(i)} - \boldsymbol{y}\|^2 - \|\boldsymbol{y} - \boldsymbol{\mu}\|^2\right] \\ = &\sum_{i=1}^m \frac{1}{t_i}\|\boldsymbol{x}^{(i)}\|^2 - 2\boldsymbol{y}^\top\left(\sum_{i=1}^m \frac{1}{t_i}\boldsymbol{x}^{(i)}\right) + 2\boldsymbol{y}^\top(W\boldsymbol{\mu}) - W\|\boldsymbol{\mu}\|^2 \\ = &\sum_{i=1}^m \frac{1}{t_i}\|\boldsymbol{x}^{(i)}\|^2 - W\|\boldsymbol{\mu}\|^2 \\ = &\sum_{i=1}^m \frac{1}{t_i}\|\boldsymbol{x}^{(i)}\|^2 - 2\boldsymbol{\mu}^\top\left(\sum_{i=1}^m \frac{1}{t_i}\boldsymbol{x}^{(i)}\right) + W\|\boldsymbol{\mu}\|^2 \\ = &\sum_{i=1}^m \frac{1}{t_i}\|\boldsymbol{x}^{(i)} - \boldsymbol{\mu}\|^2 \end{aligned}$$

□

Corollary 3.1.8. *For any distribution densities of $\boldsymbol{T}$, say $\tau(\boldsymbol{t})$. Let*

$$\tau'(\boldsymbol{t}) \propto \tau(\boldsymbol{t}) \left(\prod_{i=1}^{m} t_i \sum_{j=1}^{m} t_j^{-1} \right)^{-\frac{d}{2}} \exp(-\sum_{i=1}^{m} l_i t_i) \tag{3.11}$$

we have , for $\boldsymbol{\mu}$ in (3.9),

$$\begin{aligned} & \frac{g(\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(m)}, \boldsymbol{y}|\boldsymbol{t}) \cdot \tau(\boldsymbol{t})}{h(\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(m)}, \boldsymbol{y}|\boldsymbol{t}) \cdot \tau'(\boldsymbol{t})} \\ \propto & \prod_{i=1}^{m} \left\{ \exp\left[-\frac{1}{2t_i} \|\boldsymbol{x}^{(i)} - \boldsymbol{\mu}\|^2 \right] \cdot \mathbb{E}\left[\exp\left(-\int_0^{t_i} \phi_i(\boldsymbol{\omega}_s^{(i)}) \mathrm{d}s \right) \right] \right\} \end{aligned} \tag{3.12}$$

Algorithm 21: MCF with variable t_i

Input: Time parameter $t_1, \ldots, t_m$;
1 **for** $i = 1, \ldots, m$ **do**
2 Simulate $\boldsymbol{x}^{(i)} \sim f_i(\cdot)$;
3 Compute $W = \sum_i \frac{1}{t_i}$;
4 Compute $\boldsymbol{\mu} = \frac{1}{W} \sum_i \frac{1}{t_i} \boldsymbol{x}^{(i)}$;
5 Simulate $\boldsymbol{y} \sim \mathcal{N}_d(\boldsymbol{\mu}, W^{-1}\boldsymbol{I}_d)$;
6 Simulate $u_1 \sim \mathcal{U}[0,1]$;
7 **if** $\log u_1 \leq -\sum_i \|\boldsymbol{x}^{(i)} - \boldsymbol{\mu}\|^2 / 2t_i$ **then**
8 **for** $i = 1, \ldots, m$ **do**
9 Simulate, with one attempt only, the path skeleton for the diffusion bridge corresponding to measure $\boldsymbol{Q}^{(i)}_{t_i, \boldsymbol{x}^{(i)}, \boldsymbol{y}}$ using Algorithm 19;
10 **if** *failed* **then**
11 Go back to Step 1;
12 Accept and output $\boldsymbol{y}$.

3.2 Acceptance Guided Parameter Tuning

The previous section presents the algorithm that allows the time parameters T_i to be random variables. Usually, it is more practical to run the algorithm such that T_i is fixed before the proposal step. In which case, the rejection step of equation (3.12) may ignore terms that does not depend on $\boldsymbol{x}^{(i)}, \boldsymbol{y}$ as long as it is upper bounded by 1.

From (3.12), it is obvious that when t_i is too large or too small, the product probability will fall towards zero. Ideally one would like to directly analyze (3.12) to get an optimal choice. However, since the second term of the function is intractable, directly optimizing the equation is usually not possible.

Moreover, the rejection sampling is two-staged, by checking acceptance for the first and second term separately, the time cost of being rejected at the first stage (Step 6)

is far less than at the second stage. Thus the choice of t_i should favor the acceptance probability for the second stage while keeping the overall acceptance rate acceptable.

Choosing t_i usually relies on trial and error since solving numerically for the optimal choice is not straightforward. We propose here to simulate a search grid from the acceptance probability, and thus avoid the need to manually guess a set of random choices to test on.

3.2.1 Acceptance Probability as Density

From equation (3.12), every t_i are independent with each other. In other words, changing t_i will not affect the choice of t_j for $i \neq j$. Thus we only consider one component of the equation

$$\underbrace{\exp\left[-\frac{1}{2t_i}\|\boldsymbol{x}^{(i)}-\boldsymbol{\mu}\|^2\right]}_{AP_1}\cdot\underbrace{\mathbb{E}\left[\exp\left\{-\int_0^{t_i}\phi_i(\boldsymbol{\omega}_s^{(i)})\mathtt{d}s\right\}\right]}_{AP_2} \tag{3.12}$$

AP_2 is an expectation with respect to the Brownian measure $\boldsymbol{W}$ and both the expectation and the integral inside it are intractable. In practise, the rejection step of AP_2 is done by first simulating a path instance $\omega_{[0,t_i]}$ and then simulating the outcome of

$$U \leq \exp\left\{-\int_0^{t_i}\phi_i(\boldsymbol{\omega}_s^{(i)})\mathtt{d}s\right\}$$

where $U \sim \mathcal{U}[0,1]$. Since AP_2 is intractable, we can instead work with the lower bound of AP_2

$$\exp\left[-\frac{1}{2t_i}\|\boldsymbol{x}^{(i)}-\boldsymbol{\mu}\|^2 - M_i t_i\right] \tag{3.13}$$

where

$$M_i := \sup_{\boldsymbol{u}\in D_i}\phi_i(\boldsymbol{u}) \tag{3.14}$$

for some $D_i \subset \mathbb{R}^d$ such that $\boldsymbol{X}^{(i)}_{[0,t_i]}$ does not leave D_i.

One does not need a choice of D_i if ϕ_i is bounded over $\mathbb{R}^d$. However, for most cases ϕ_i is unbounded, and one can simulate the layer intervals D_i for each dimension before simulating the skeleton of the bridge.

Let $d_i = \|\boldsymbol{x}^{(i)}-\boldsymbol{\mu}\|$. Equation (3.13) can be split into a product of two improper density functions as follows

$$\underbrace{t_i^{-\frac{3}{2}}\exp\left[-\frac{1}{2t_i}d_i^2-(M_i-\beta)t_i\right]}_{L_1}\cdot\underbrace{t_i^{\frac{3}{2}}\exp(-\beta_i t_i)}_{L_2} \tag{3.15}$$

where

$$L_1 \sim \text{IG}\left(\sqrt{\frac{d_i^2}{2(M_i-\beta)}}, d_i^2\right), \quad L_2 \sim \Gamma\left(\frac{5}{2}, \beta_i\right), \quad 0 < \beta_i < M_i.$$

The choice of β_i will affect the efficiency of simulation. In this case, we would like to make the distributions L_1 and L_2 match each other as closely as possible. One simple way is to choose β_i such that the mean of two distributions match, i.e.

$$\frac{5}{2\beta_i} = \sqrt{\frac{d_i^2}{2(M_i-\beta_i)}}$$

and

$$\beta_i = \frac{-25 + \sqrt{25^2 + 200 d_i^2 M_i}}{4 d_i^2}.$$

It is trivial to verify that β_i always lies in $(0, M_i)$. The layered diffusion bridge approach simulates the layer $\boldsymbol{U}_i$ given the two ends of the diffusion bridge which required the length t_i to be decided before the simulation. In this case, an iterative approach can be applied as in Algorithm 22

Algorithm 22: Simulating t_i w.r.t acceptance probability

Result: A choice for T_i

Set $t_i^{(0)}, i = 1, \ldots, m$ to some value of your choice;
for *j = 1,2,...* **do**
  Sample the starting points $\boldsymbol{x}^{(i)} \sim f_i(\cdot)$ for each i;
  Compute the weighted mean $\boldsymbol{\mu}$ by (3.9);
  Simulate endpoint $\boldsymbol{y} \sim \mathcal{N}_d\left(\boldsymbol{\mu}, \left(\sum t_i^{-1}\right)^{-1} I_d\right)$ with respect to $t_i^{(j-1)}$;
  for *i = 1,2,...,m* **do**
    Simulate the layer I_i (Algorithm 15) starting at $\boldsymbol{x}^{(i)}$ ending at $\boldsymbol{y}$ with time $t_i = t_i^{(j-1)}$;
    Compute M_i;
    Simulate $t_i^{(j)}$ from the product distribution (3.15) using MCF (Algorithm 20);
  end
end

3.2.2 Same t across all components

When all t_is are equal, then the mean $\boldsymbol{\mu}$ is a simple average and the lower bound of acceptance probability becomes

$$\exp\left\{-\frac{C\sigma^2}{2t}\right\}\cdot\exp\left[-Mt\right]$$

where

$$\sigma^2 = C^{-1}\sum_{i=1}^{C}\|\boldsymbol{x}^{(i)}-\boldsymbol{\mu}\|^2, \quad M := \sum_{i=1}^{C} M_i$$

and M_i is defined the same way as previously. The same split can be applied which gives

$$L_1 \sim \text{IG}\left(\sqrt{\frac{C\sigma^2}{2(M-\beta)}}, C\sigma^2\right), \qquad L_2 \sim \Gamma\left(\frac{5}{2}, \beta\right)$$

and β can be chosen to be

$$\beta = \frac{-25+\sqrt{25^2+200C\sigma^2M}}{4C\sigma^2}$$

3.2.3 Search grid from drawn samples

It is clear that the acceptance probability (3.12) is positively skewed, meaning that the mode is smaller than the mean. The difference in rejection cost means that we would favor a higher pass rate in the second rejection stage and thus smaller T.

Therefore, one way to set the search grid is by taking various quantiles of the sample set with denser points near 0 and quickly testing the time efficiency for each case. In this way, the choice of T can be close to optimal. For the case where T_i are chosen differently, a multi-dimensional search grid might not be time efficient to test when C is large, in which case a one-dimensional search grid can be implemented where the quantile point for each T_i are set to be the same.

3.2.4 Simulation Studies

In this section, two cases will be presented showing the performance of T sampling in two different views.

3.2.4.1 Case study 1

In this case, we consider to sample from the following density function

$$f(x) \propto \exp\left[\alpha x - \beta e^x\right]\cdot\exp\left[-\frac{1}{2}(x+\frac{\lambda}{\mu^2}e^x+\lambda e^{-x})\right] \tag{3.16}$$

Quantile	(t_1, t_2)	Rate	Sim. time	t_{mean}	Rate	Sim. time
0.01%	(0.018, 0.009)	10.8%	151.67s	0.013	10.8%	151.53s
0.1%	(0.026, 0.015)	13.0%	142.42s	0.020	13.0%	143.75s
0.5%	(0.052, 0.032)	18.0%	138.50s	0.042	17.7%	142.01s
1%	(0.069, 0.041)	20.1%	140.90s	0.055	19.7%	145.52s
5%	(0.155, 0.079)	26.1%	166.10s	0.117	25.2%	180.10s
10%	(0.225, 0.115)	28.2%	197.60s	0.170	26.7%	222.64s
15%	(0.297, 0.147)	29.2%	236.61s	0.222	27.1%	273.00s
20%	(0.381, 0.177)	29.2%	285.48s	0.279	26.7%	343.92s
30%	(0.584, 0.238)	28.1%	420.42s	0.411	23.6%	579.06s
40%	(0.854, 0.303)	24.9%	686.33s	0.578	18.8%	1141.4s
50%	(1.221, 0.382)	19.8%	1382.7s	0.801	13.0%	3329.4s

Table 3.1 Case 1, fusing transformed Gamma and transformed inverse Gaussian.

Note that by performning a change of variable $x = \exp(y)$, the first component becomes a Gamma distribution $\Gamma(\alpha, \beta)$ and the second becomes an Inverse Gaussian IG(μ, λ). In specific, $\Gamma(1, 2)$ and IG$(1, 3)$ are fused together. Algorithm 22 is used to sample 5000 samples and different quantiles are chosen in each dimension for each component. For a more stable simulation, instead of using the previous T, the median of the current simulated Ts is used. Then for each set of (T_1, T_2), MCF algorithm is used to draw 100,000 samples from (3.16). For comparison between the case using the same T, the mean $(T_1 + T_2)/2$ of each set previously is used and 100,000 samples are drawn. The performance is displayed in both time and acceptance rate in table 3.1.

Both the acceptance rate and total simulation time are listed in table 3.1. For the fusion algorithm, a higher acceptance rate does not always lead to faster simulation due to the two separate rejection stages. To avoid ambiguity, we here use simulation time as the eThere are several observations one can make from table 3.1. Firstly, the optimal T seems to be located close to the 0.5% quantile at about 0.04. Secondly, a higher overall acceptance rate does not mean a more efficient simulation as the cost of being rejected at the first stage is much less than being rejected at the second stage. Moreover, when the difference between T_1 and T_2 is large, using different T out-performs using the mean of T_1 and T_2.

Recall that the optimality of T relates to the equation (3.13), more specifically to M_i. In this case, both components do not have an upper bound for the function ϕ_i in equation (3.14), and M_i, the upper bound within $\boldsymbol{U}_i$, can vary significantly with the layer $\boldsymbol{U}_i$. Moreover, the growth rate also differs for the two components, and thus the simulated T_i.

Quantile	(T_1, T_2, T_3)	Rate	Sim. time	T_{mean}	Rate	Sim. time
0.01%	(0.053, 0.044, 0.059)	0.5%	970.91s	0.052	0.5%	969.43s
0.1%	(0.095, 0.087, 0.099)	0.8%	653.81s	0.094	0.8%	650.99s
0.5%	(0.154, 0.155, 0.198)	1.3%	525.36s	0.169	1.3%	523.16s
1%	(0.193, 0.226, 0.269)	1.6%	521.48s	0.229	1.6%	516.81s
5%	(0.391, 0.474, 0.503)	2.1%	759.33s	0.456	2.1%	747.86s
10%	(0.557, 0.644, 0.683)	2.1%	1114.53s	0.628	2.1%	1104.7s
15%	(0.681, 0.785, 0.830)	2%	1525.72s	0.765	1.9%	1512.31s
20%	(0.799, 0.909, 0.958)	1.8%	2033.09s	0.889	1.8%	2005.89s
30%	(1.03, 1.14, 1.19)	1.4%	3394.68s	1.12	1.4%	3367.82s
40%	(1.27, 1.37, 1.41)	1.1%	5625.12s	1.35	1.1%	5593.73s
50%	(1.51, 1.61, 1.64)	0.8%	9198.64s	1.59	0.8%	9166.02s

Table 3.2 Case 2, fusing three Student's t-distribution.

3.2.4.2 Case study 2

In contrast to the previous case, when the components have better behavior and optimal T for each component is close to each other, using different T can hinder the performance due to the difficulty in perfectly tuning T for each component.

In this case, three translated Student's t-distributions are fused. Let $T(\mu, \nu)$ denote the translated Student's t-distribution with mean μ and ν degrees of freedom. The target distribution is

$$f \propto T(0, 3) \cdot T(1, 5) \cdot T(-2, 7). \tag{3.17}$$

The choosing strategy of T is the same as previous case, and again 100,000 samples are drawn from the target distribution. The result is shown in table 3.2. From the table, we can see that using different T for each component does not always improve the simulation efficiency, and in many cases using the mean actually have better performance (slightly). The underlying reason is that the optimal T for each component might be at different quantiles and it is hard to perfectly tune it due to the combinatorial complexity. Instead, since the optimal Ts are very close, it is easier to tune when the T is the same for each component the one can have more luck getting close to the optimal choice.

The Student's t-distribution is "well-behaved" because there exists an upper bound for function ϕ over the whole real axis and the maximum is attained usually not very far from the mean. In addition, the maximum is quite close for the three components in this case. Thus the optimal T is also close for each component.

3.2.4.3 Interpretation of optimal T_i

From equation (3.12) and the case studies, we may conclude that the choice of T_i is affected by two main factors:

- Similarity between the component i and the target distribution;
- Growth rate of $\sup \phi_i$.

When a component is closer to the target distribution, it is assigned a greater weight in $\boldsymbol{\mu}$ and thus a smaller T_i and vice versa. This is shown in both simulation cases where the component with the smallest T_i is the one that is closest to the target distribution.

The absolute size of T_i also depends on the similarity between the components and the target. If none of the components are similar to the target, then it is less likely for their proposals $_i$ to be close to each other, and larger T_i is assigned to compensate for the first rejection stage.

In contrast, if the upper bound ϕ_i grows quickly, then a smaller T_i is preferred, even though smaller T_i is always preferred, pulling the optimal T_i towards 0. The last two situations are conflicting, thus the algorithm performance will be low despite having the optimal T_i if both conditions are present.

CHAPTER 4

Monte Carlo Fusion under Linear Constraints

While in the previous chapter we considered the density fusion problem where the target distribution of form

$$f(\boldsymbol{y}) \propto \prod_{i=1}^{m} f_i(\boldsymbol{y}). \tag{3.1}$$

Alternatively, the above equation can be viewed as a constrained distribution

$$f(\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(m)}) \propto f^*(\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(m)}) \mathbb{I}_{\boldsymbol{y}^{(1)}=\cdots=\boldsymbol{y}^{(m)}},$$
$$f^*(\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(m)}) = \prod_{i}^{m} f_i(\boldsymbol{y}^{(i)}).$$

In this chapter, we consider a more general ***linearly*** constrained fusion distribution

$$f_{\mathcal{H}}(\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(m)}) \propto f_1(\boldsymbol{y}^{(1)}) f_2(\boldsymbol{y}^{(2)}) \cdots f_m(\boldsymbol{y}^{(m)}) \mathbb{1}_{\mathcal{H}}(\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(m)}), \tag{4.1}$$

where every $f_i(\cdot)$ is a proper density function, corresponding to the distribution of $\boldsymbol{y}^{(i)}$. The notation $\mathbb{1}_{\mathcal{H}}(\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(m)})$ asserts the constraint that $(\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(m)}) \in \mathcal{H} \subset \mathbb{R}^{dm}$. Here, $\mathcal{H}$ will be a linear equality constraint dependent only on $\boldsymbol{y}^{(i)}$.

The linear constraint in the target distribution is motivated by time series models where the constraint may be the sum of energy consumption. For instance, the total consumption in a day is equal to the sum of consumption in each hour of that day, in which case we have a constraint relating to 24 random variables for the hourly reading and a known constant for the total. We will show a few examples in the next chapter whereas this chapter will focus on the theoretical derivations for establishing the Monte Carlo Fusion method with respect to general linear constraints.

It is not straightforward to use classical methods to estimate the mean of $\boldsymbol{y}^{(i)}$ since its density $f_{\mathcal{H}}$ may not have an analytical solution. Even if one may be able to compute the mean analytically for density (4.1) using the classical approach, such analytical solutions may not exist for other estimates like confidence intervals (CI), as this requires solving equations with intractable integrals. This is not hard without constraint, due to the independence between $\boldsymbol{y}^{(i)}, i = 1, \cdots, m$ and the simplicity of f_i. However, when

a constraint is added to the model, the independence is lost, and thus not trivial to work out the estimators analytically even for simple density f_i.

4.1 Some Background in Constrained Simulation

In most cases, ordinary simulation methods, like the Metropolis-Hastings method or the Gibbs sampler, cannot be directly used on linear-equality-constrained problems even for simple distributions. Due to the reduction in dimensions of the sample space, the linear hyperplane has zero probability under the full original measure space, and hence, one has zero chance to draw a sample from the original distribution that lands exactly on the constraint. For this reason, the difficulty of implementing an MCMC algorithm on a constrained sampling problem lies in that the algorithm needs to generate proposals that satisfy the constraint. In practice, task-specific adaptations are usually applied to the base sampler where proposals are generated in such a way that the constraints are naturally met without violating the detailed balance (Asselin et al., 2010, Valenzuela and Mazumdar, 2001, Maggs, 2005).

For a generic setup, the constraint is usually satisfied by means of projection or transformation onto the constraint set. Zappa et al. (2018) and Chua (2020) both consider generating proposals by sampling from the tangential plane and then projecting onto the manifold. Chua (2020) presents a way to efficiently expand the base sample set to $n \times m$ weighted samples that are approximately distributed as the target distribution. Zappa et al. (2018) presents a modified Metropolis-Hastings(MH) algorithm where the proposals are generated through tangential projections onto the manifold. This algorithm resembles the usual random walk MH algorithm in that each proposal has a relatively low cost to generate and the rejection rate is directly related to the step size. Since the proposal generation depends on the result of an iterative solver, there is an additional rejection stage for reverse projection check to ensure every step is reversible, i.e., the iterative solver can also move from the proposal back to the current state.

The other branch builds upon the Hamiltonian Monte Carlo (HMC) method, where proposals are generated from a simulated Hamiltonian system. CHMC (Brubaker et al., 2012) extends HMC where samples are generated by including the constraint into the Hamiltonian system and the evolution of which is solved by a constrained integrator. The problem with CHMC is that the usual explicit integrator cannot be adopted as there is a constraint on the system, and the implicit integrator requires an iterative solver for each simulation step. A special case is Geodesic HMC (Byrne and Girolami, 2013) which splits the Hamiltonian system such that the integrator avoids the need for an iterative solver for simulating the Hamiltonian mechanics, given that the geodesic flow can be exactly computed. This approach can be applied in directional statistics where the

state space is usually an n-sphere for which the geodesics are explicitly known. Another problem of HMC is that the simulated Hamiltonian system needs to be reversible up to momentum reversal for detailed balance to hold. Although such reversibility is usually satisfied for sufficiently small time steps, this might be violated when the parameter is tuned for more efficient simulation.

Although MCMC may be adapted, its choice of proposal distributions is usually limited, e.g., mostly Gaussian-based distributions. In the case of HMC, one has the freedom to choose the guidance Hamiltonian which relies heavily on the user's experience to find a good guidance function. On the other hand, exact sampling under equality constraints is genuinely hard even for simple constraints like linear ones. The only equality-constrained exact simulation methods deal with Gaussian distribution (Cong et al., 2017, Vrins, 2018) or T-distributions (Kibria and Joarder, 2006) under linear constraints, which has a limited range of application.

In this chapter, we will tackle the problem from another point of view by extending the Monte Carlo Fusion algorithm to handle linear constraints. Recall that the MCF algorithm employs m Langevin diffusions, which start from a value at time 0 following the distribution f_i, with their ending points at time T following a Gaussian distribution with the required constraints. Therefore, the original non-Gaussian constrained problem becomes a Gaussian constrained problem. Finally, the outcomes based on Gaussian constraints will be adjusted according to a path-space rejection sampling for the Langevin diffusion processes. This adjusted ending point at T exactly satisfies the constraint and has the required target distribution.

4.2 From Unconstrained to Constrained

It is usually hard to find a proposal distribution that is appropriate for the target distribution and is supported only on the constraint, even for a linear constraint. Luckily, when the proposal distribution is Gaussian and the constraint is linear, we can easily generate proposals that land exactly on the constraint.

For this section, we start by restating the result from Chapter 2 with a slight change to accommodate the more general setup (section 4.2.1), then we move on to prove that the same Radon-Nikodym derivative can be derived in the constrained case (section 4.2.2) and finally show how the original algorithm needs to be altered in order to deal with the constrained problem (section 4.2.3).

4.2.1 Unconstrained Case Revisited

Recall that the general linearly constrained simulation problem allows each component to take different values

$$f_{\mathcal{H}}(\boldsymbol{y}^{(1)},\ldots,\boldsymbol{y}^{(m)}) \propto f_1(\boldsymbol{y}^{(1)})f_2(\boldsymbol{y}^{(2)})\cdots f_m(\boldsymbol{y}^{(m)})\mathbb{1}_{\mathcal{H}}(\boldsymbol{y}^{(1)},\ldots,\boldsymbol{y}^{(m)}). \tag{4.1}$$

To avoid the repetition, we shall directly state the target and proposal distributions which are similar to the ones before.

Proposition 4.2.1. *Consider a set of m diffusion processes of length T, with the transition kernel $p_i(\boldsymbol{X}_T|\boldsymbol{X}_0)$, $i=1,2,\cdots,m$ such that process i admits $f_i^2(\cdot)$ as its invariant distribution. Then the joint density*

$$g\left(\boldsymbol{x}^{(1)},\cdots,\boldsymbol{x}^{(m)},\boldsymbol{y}^{(1)},\cdots,\boldsymbol{y}^{(m)}\right) \propto \prod_{i=1}^{m} f_i^2(\boldsymbol{x}^{(i)})p_i(\boldsymbol{y}^{(i)}|\boldsymbol{x}^{(i)})\frac{1}{f_i(\boldsymbol{y}^{(i)})}. \tag{4.2}$$

admits the unconstrained target density

$$\prod_{i=1}^{m} f_i(\boldsymbol{y}^{(i)})$$

as the marginal distribution of the ending points $\left(\boldsymbol{y}^{(1)},\cdots,\boldsymbol{y}^{(m)}\right)$.

The above diffusion processes $(\boldsymbol{X}_t^{(i)})_{t\in[0,T]}$, $i=1,\ldots,m$ can be obtained by the same construction as in Section 3.1.1 driven by equations

$$\mathtt{d}\boldsymbol{X}_t^{(i)} = \boldsymbol{\nabla}A_i(\boldsymbol{X}_t^{(i)})\mathtt{d}t + \mathtt{d}\boldsymbol{B}_t^{(i)},$$

where $A_i(\boldsymbol{x}) = \log f_i(\boldsymbol{x})$.

Similarly, consider the proposal h as follows:

$$h(\boldsymbol{x}^{(1)},\cdots\boldsymbol{x}^{(m)},\boldsymbol{y}^{(1)},\cdots,\boldsymbol{y}^{(m)}) = \prod_{i=1}^{m} f_i(\boldsymbol{x}^{(i)})(2\pi T)^{-1/2}\exp\left[-\frac{||\boldsymbol{y}^{(i)}-\boldsymbol{x}^{(i)}||^2}{2T}\right], \tag{4.3}$$

which can be easily sampled from.

Theorem 4.2.2. *Under Condition 2,*

$$\frac{g\left(\boldsymbol{x}^{(1)},\cdots,\boldsymbol{x}^{(m)},\boldsymbol{y}^{(1)},\cdots,\boldsymbol{y}^{(m)}\right)}{h(\boldsymbol{x}^{(1)},\cdots,\boldsymbol{x}^{(m)},\boldsymbol{y}^{(1)},\cdots,\boldsymbol{y}^{(m))})} \propto \mathbb{E}\left[\exp\left(-\sum_{i=1}^{m}\int_0^T \phi_i(\boldsymbol{X}_s^{(i)})\mathtt{d}s\right)\right] \tag{4.4}$$

where $\mathbb{E}$ is taking expectation over the measure induced by Brownian bridges of length T connecting $(\boldsymbol{x}^{(1)},\cdots,\boldsymbol{x}^{(m)})$ and $(\boldsymbol{y}^{(1)},\cdots,\boldsymbol{y}^{(m)})$.

Proof. Using the equation (3.4) from Proposition 3.1.3 and simplify g/h gives the desired result. □

The above is basically the same as previously in the MCF algorithm, next we need to check that the constraint on the end point does not affect the Radon-Nikodym derivative.

4.2.2 Restricted Radon-Nikodym Derivative

Proposition 4.2.3 (Randon-Nikodym derivative on submanifold)**.** *Let $\mathbb{P}, \mathbb{O}$ be two probability measures on the Euclidean space $\left(\mathbb{R}^{md}, \mathcal{B}(\mathbb{R}^{md})\right)$, with density functions $f_{\mathbb{P}}$ and $f_{\mathbb{O}}$ respectively, such that $\mathbb{P} \ll \mathbb{O}$. Let*

$$\vec{\boldsymbol{h}} : \mathbb{R}^{md} \to \mathbb{R}^k,\ 0 < k < md$$

be a smooth function such that $\forall \boldsymbol{u} \in \mathcal{H}$, where

$$\mathcal{H} := \vec{\boldsymbol{h}}^{-1}(\mathbf{0}) = \left\{ \boldsymbol{u} \in \mathbb{R}^{md} : \vec{\boldsymbol{h}}(\boldsymbol{u}) = \mathbf{0} \right\},$$

the derivative of $\vec{\boldsymbol{h}}$ at each dimension, $\mathtt{D}\vec{\boldsymbol{h}}_{\boldsymbol{u}} : \mathbb{R}^{md} \to \mathbb{R}^{k \times md}$ is surjective, where $\mathtt{D}$ is the differentiation operator in each component. Then the following holds:

1. *There exists a canonical volume measure $\mathbb{V}^{\mathcal{H}}$ defined on $\mathcal{H}$.*
2. *If the integral of $f_{\mathbb{P}}$ is finite on $\mathcal{H}$, then there exists an unique measure $\mathbb{P}^{\mathcal{H}}$ defined on the measurable space induced by $\mathcal{H}$ such that its Radon-Nikodym derivative w.r.t to $\mathbb{V}^{\mathcal{H}}$ is proportional to its unconstrained counterpart w.r.t. the Lebesgue measure $\mathtt{d}\boldsymbol{u}$*
$$\mathtt{d}\mathbb{P}^{\mathcal{H}}/\mathtt{d}\mathbb{V}^{\mathcal{H}} \propto \mathtt{d}\mathbb{P}/\mathtt{d}\boldsymbol{u}$$
3. *$\mathbb{P}^{\mathcal{H}} \ll \mathbb{O}^{\mathcal{H}}$, and*
$$\mathtt{d}\mathbb{P}^{\mathcal{H}}/\mathtt{d}\mathbb{O}^{\mathcal{H}} \propto \mathtt{d}\mathbb{P}/\mathtt{d}\mathbb{O}.$$

Proof.

1. Suppose that constraint $\mathcal{H} \subset \mathbb{R}^n$ is defined to be the level set of a smooth function
$$\vec{\boldsymbol{h}} : \mathbb{R}^{md} \to \mathbb{R}^k$$
and
$$\mathcal{H} := \vec{\boldsymbol{h}}^{-1}(\mathbf{0}) = \left\{ \boldsymbol{u} \in \mathbb{R}^{md} : \vec{\boldsymbol{h}}(\boldsymbol{u}) = \mathbf{0} \right\}$$
Suppose also that $\forall \boldsymbol{u} \in \mathcal{H}$, the derivative of $\vec{\boldsymbol{h}}$ at $\boldsymbol{u}$, $\mathtt{D}\vec{\boldsymbol{h}}_{\boldsymbol{u}} : \mathbb{R}^{md} \to \mathbb{R}^{k \times md}$ is surjective. Then by Regular Value Theorem, $\mathcal{H}$ is an $(md-k)$-dimensional manifold. Since the manifold $\mathcal{H}$ is defined in $\mathbb{R}^{md}$, there exists a Riemannian metric

g defined on $\mathcal{H}$ induced by the Euclidean metric on $\mathbb{R}^{md}$. More importantly, the Canonical Volume measure $\mathbb{V}^{\mathcal{H}}$ can be defined on $(\mathcal{H}, g)$, such that $\int_{\mathcal{H}} \mathtt{d}\mathbb{V}^{\mathcal{H}}$ is exactly the volume of $\mathcal{H}$. (Section 1 of Chapter XII Amann and Escher (2009) is recommended for detailed explanation of canonical volume measure.)

2. Since the probability measure $\mathbb{P}$ is defined on the Euclidean space, it has a corresponding density function $f_{\mathbb{P}}$ and

$$\frac{\mathtt{d}\mathbb{P}}{\mathtt{d}\boldsymbol{u}} = f_{\mathbb{P}}(\boldsymbol{u})$$

Then define the probability measure $\mathbb{P}^{\mathcal{H}}$ that for any set $E \subset \mathcal{H}$ measurable by $\mathbb{V}^{\mathcal{H}}$

$$\mathbb{P}^{\mathcal{H}}(E) = \int_E Z_{\mathbb{P}}^{-1} f_{\mathbb{P}} \, \mathtt{d}\mathbb{V}^{\mathcal{H}}, \quad \text{where} \quad Z_{\mathbb{P}} = \int_{\mathcal{H}} f_{\mathbb{P}} \, \mathtt{d}\mathbb{V}^{\mathcal{H}} < \infty.$$

Uniqueness is obvious. Suppose that there exist another probability measure $\tilde{\mathbb{P}}^{\mathcal{H}}$ also defined on $\mathcal{H}$ such that

$$\frac{\mathtt{d}\tilde{\mathbb{P}}^{\mathcal{H}}}{\mathtt{d}\mathbb{V}^{\mathcal{H}}}(\boldsymbol{u}) = C f_{\mathbb{P}}(\boldsymbol{u}), \qquad C \neq 0.$$

Then

$$0 \;=\; \mathbb{P}^{\mathcal{H}}(\mathcal{H}) - \tilde{\mathbb{P}}^{\mathcal{H}}(\mathcal{H}) = \int_{\mathcal{H}} (Z_{\mathbb{P}}^{-1} - C) f_{\mathbb{P}} \mathtt{d}\, \mathbb{V}^{\mathcal{H}}.$$

Since $f_{\mathbb{P}}$ is non-negative and not zero everywhere on $\mathcal{H}$, we have $C = Z_{\mathbb{P}}^{-1}$ and thus $\mathbb{P}^{\mathcal{H}} = \tilde{\mathbb{P}}^{\mathcal{H}}$.

3. Since $\mathbb{P} \ll \mathbb{O}$, there exists non-negative function μ defined on $\mathbb{R}^{md}$ such that

$$\frac{\mathtt{d}\mathbb{P}}{\mathtt{d}\mathbb{O}} = \mu.$$

Clearly, $f_{\mathbb{P}}(\boldsymbol{u})/f_{\mathbb{O}}(\boldsymbol{u}) = \mu(\boldsymbol{u})$. Then

$$\mathbb{P}^{\mathcal{H}}(E) = \int_E Z_{\mathbb{P}}^{-1} f_{\mathbb{P}} \mathtt{d}\mathbb{V}^{\mathcal{H}} = \frac{Z_{\mathbb{O}}}{Z_{\mathbb{P}}} \int_E Z_{\mathbb{O}}^{-1} \mu f_{\mathbb{O}} \mathtt{d}\mathbb{V}^{\mathcal{H}}.$$

Thus $\frac{\mathtt{d}\mathbb{P}^{\mathcal{H}}}{\mathtt{d}\mathbb{O}^{\mathcal{H}}} = \frac{Z_{\mathbb{O}}}{Z_{\mathbb{P}}}\mu \propto \mu$ and clearly $\mathbb{P}^{\mathcal{H}} \ll \mathbb{O}^{\mathcal{H}}$.

□

Example 4.2.1. *Consider* $\vec{\boldsymbol{h}}(x_1, x_2) := x_1^2 + x_2^2$. *Then* $\mathtt{D}\vec{\boldsymbol{h}}_{(x_1,x_2)} = \begin{pmatrix} 2x_1 & 2x_2 \end{pmatrix}$. *Since* $(0,0) \in \vec{\boldsymbol{h}}^{-1}(0)$ *and* $\mathtt{D}\vec{\boldsymbol{h}}_{(0,0)} = \begin{pmatrix} 0 & 0 \end{pmatrix}$ *is degenerate. Thus* $\vec{\boldsymbol{h}}^{-1}(0)$ *is not a manifold. However,* $\vec{\boldsymbol{h}}^{-1}(c)$, $c > 0$ *is a manifold.*

Example 4.2.2. *If* $\vec{\boldsymbol{h}} : \mathbb{R}^n \to \mathbb{R}^m$ *is linear,* $\exists \boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\vec{\boldsymbol{h}}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$. *Then* $\mathtt{D}\vec{\boldsymbol{h}}_{\boldsymbol{x}} = \boldsymbol{A}$, *thus* $\vec{\boldsymbol{h}}^{-1}(c)$ *is a manifold of dimension* $n - m$ *if and only if* $\boldsymbol{A}$ *is full rank.*

Let $\mathbb{Q}^{(i)}_{T,\boldsymbol{x},\boldsymbol{y}}$ denote the probability measure induced by a Langevin bridge $\boldsymbol{X}^{(i)}$ of length T starting at $\boldsymbol{x}$ and ending at $\boldsymbol{y}$. Then we may define a biased Langevin process with probability law given by

$$\mathbb{Q}^{(i)} \propto f_i^2(\boldsymbol{x}) p_i(\boldsymbol{y}|\boldsymbol{x})(f_i(\boldsymbol{y}))^{-1} \cdot \mathbb{Q}^{(i)}_{T,\boldsymbol{x},\boldsymbol{y}}$$

which admits the joint marginal distribution for its two ends as $f_i^2(\boldsymbol{x}) p_i(\boldsymbol{y}|\boldsymbol{x})(f_i(\boldsymbol{y}))^{-1}$. Thus, the product measure $\mathbb{Q} := \mathbb{Q}^{(1)} \times \cdots \times \mathbb{Q}^{(m)}$ admits $g\left(\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(m)}, \boldsymbol{y}^{(1)}, \cdots, \boldsymbol{y}^{(m)}\right)$ defined in (4.2) as the joint distribution of the ends and admits the unconstrained product density as the marginal density for $\boldsymbol{y}$.

Let $\mathbb{W}^{(i)}_{T,\boldsymbol{x},\boldsymbol{y}}$ denote the probability measure induced by a Brownian bridge of length T starting at $\boldsymbol{x}$ ending at $\boldsymbol{y}$. Furthermore, let $\mathbb{Z}_i$ denote a biased Brownian process such that the product measure $\mathbb{Z} := \mathbb{Z}_1 \times \cdots \times \mathbb{Z}_m$ admits (4.3) as the joint distribution of its ends.

Now we can present the main result in the constrained space. Recall the constraint $\mathcal{H}$ in (4.1), which is our target distribution. Denote $\mathbb{Q}^{\mathcal{H}}$ as the measure induced by constraining the end-point $(\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(m)})$ within the set $\mathcal{H}$ and clearly $\mathbb{Q}^{\mathcal{H}}$ admits (4.1) as its ending point marginal distribution. The following Corollary tells us that we can have a proposal measure $\mathbb{Z}^{\mathcal{H}}$ having the same constraint and their Radon-Nikodym Derivative is the same as their unconstrained counterpart.

Corollary 4.2.4 (Constrained derivative)**.** *Let the constraint set $\mathcal{H} \subset \mathbb{R}^n$ satisfies the condition in Proposition 4.2.3. Then*

$$\frac{\mathtt{d}\mathbb{Q}^{\mathcal{H}}}{\mathtt{d}\mathbb{Z}^{\mathcal{H}}} \propto \frac{\mathtt{d}\mathbb{Q}}{\mathtt{d}\mathbb{Z}} \propto \exp\left(-\sum_{i=1}^{m} \int_0^T \phi_i(\boldsymbol{X}_s^{(i)}) \mathtt{d}s\right)$$

where the measure with superscript $^{\mathcal{H}}$ denotes the induced measure by constraining its end-point $(\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(m)})$ within the set $\mathcal{H}$.

Proof. By construction, measure $\mathbb{Q}$ has the conditional decomposition of

$$\mathtt{d}\mathbb{Q} = \mathtt{d}\mathbb{Q}_{T,\boldsymbol{x},\boldsymbol{y}} \cdot p(\mathtt{d}\boldsymbol{x}) \cdot q(\mathtt{d}\boldsymbol{y}|\mathtt{d}\boldsymbol{x}),$$

i.e., the diffusion bridge measure connecting $\boldsymbol{x}$ and $\boldsymbol{y}$, the distribution of end point $\boldsymbol{y}$ conditioned on $\boldsymbol{x}$, and the distribution of starting point $\boldsymbol{x}$. Here, we consider another conditional decomposition given by $\mathtt{d}\mathbb{Q} = \mathtt{d}\mathbb{Q}_{|Y} \cdot \mathtt{d}\mathbb{Y}_{\mathbb{Q}}$, where $\mathbb{Y}_{\mathbb{Q}}$ is a measure on $\mathbb{R}^m$ for the end point $\boldsymbol{y}$ and $\mathbb{Q}_{|Y}$ is the measure of the biased Langevin process conditioned on the end point. Similarly, we consider the same decomposition for $\mathbb{Z}$.

Since $\mathbb{Q}_{|Y}$ and $\mathbb{Z}_{|Y}$ are both independent of the end point, we have

$$\mathtt{d}\mathbb{Q}^{\mathcal{H}} \propto \mathtt{d}\mathbb{Q}_{|Y}\mathtt{d}\mathbb{Y}^{\mathcal{H}}_{\mathbb{Q}}, \qquad \mathtt{d}\mathbb{Z}^{\mathcal{H}} \propto \mathtt{d}\mathbb{Z}_{|Y}\mathtt{d}\mathbb{Y}^{\mathcal{H}}_{\mathbb{Z}}.$$

Thus the result follows by applying Proposition 4.2.3 to $\frac{\mathrm{d}\mathbb{Y}_{\mathbb{Q}}}{\mathrm{d}\mathbb{Y}_{\mathbb{Z}}}$. □

Since $\mathbb{Q}^{\mathcal{H}}$ admits (4.1) as the marginal distribution for its end-points $(\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(m)})$, we can use a similar rejection sampling strategy as in Algorithm 20 for sampling from the constrained density (4.1), given that we can generate proposals from $\mathbb{Z}^{\mathcal{H}}$. The detailed expressions of $\mathbb{Q}^{\mathcal{H}}$ and $\mathbb{Z}^{\mathcal{H}}$ are simply adding the constraint $\mathcal{H}$ to $\mathbb{Q}$ and $\mathbb{Z}$, respectively (see the proof of Corollary 1).

4.2.3 Simulation from the Constrained Proposal

Now the final problem is to generate samples from the proposal measure $\mathbb{Z}^{\mathcal{H}}$ which satisfies the constraint $\mathcal{H}$. Recall the proposal distribution h in (4.3) and apply the constraint, giving

$$\tilde{h}\left(\boldsymbol{x}^{(1:m)}, \boldsymbol{y}^{(1:m)}\right) \propto \left(\prod_{i=1}^{m} f_i(\boldsymbol{x}^{(i)})\right) f_{\boldsymbol{y}|\boldsymbol{x}}\left(\boldsymbol{y}^{(1:m)}|\boldsymbol{x}^{(1:m)}\right) \mathbb{1}_{\mathcal{H}}(\boldsymbol{y}^{(1:m)}), \tag{4.3c}$$

where

$$f_{\boldsymbol{y}|\boldsymbol{x}}\left(\boldsymbol{y}^{(1:m)}|\boldsymbol{x}^{(1:m)}\right) = \prod_{i=1}^{m} (2\pi T)^{-1/2} \exp\left[-\frac{||\boldsymbol{y}^{(i)} - \boldsymbol{x}^{(i)}||^2}{2T}\right]. \tag{4.5}$$

Algorithm 23: Linearly Constrained Langevin Rejection Sampler

input: Constraint $A\boldsymbol{y} = \boldsymbol{c}$; component distributions $f_i, i = 1, \ldots, C$; parameter T

1 Simulate, for each $1 \leq i \leq m$, $\boldsymbol{x}^{(i)} \sim f_i(\cdot)$;
2 Simulate $\boldsymbol{y} = (\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(m)}) \sim \mathcal{N}(\boldsymbol{x}^{(1:m)}, T\boldsymbol{I}_d)$ constrained on $\boldsymbol{y}^{(1:m)} \in \mathcal{H}$;
3 Simulate a uniform random variable $U_1 \in \mathcal{U}[0, 1]$;
4 **if** $U_1 \leq AP_1 = L\eta_{\vec{\boldsymbol{h}}}(\boldsymbol{x}^{(1:m)})$ **then**
5 **for** $i = 1, ..., m$ **do**
6 Simulate a layered Brownian Bridge of length T connecting $\boldsymbol{x}^{(i)}$ and $\boldsymbol{y}^{(i)}$;
7 **end**
8 Let $U_2 \in \mathcal{U}[0, 1]$ and simulate the event $U_2 < AP_2$ given by expression (4.4);
9 **if** $U_2 < AP_2$ *is true* **then**
10 Accept and return $\boldsymbol{y}^{(1:m)}$;
11 **else**
12 Go back to step 1;
13 **end**
14 **else**
15 Go back to step 1;
16 **end**

To simulate a process $\boldsymbol{X}$ from $\mathbb{Z}^{\mathcal{H}}$, we just need to simulate the end points from (4.3c) and then fill in the middle $\boldsymbol{X}_s, s \in (0,T)$ as Brownian bridges. Denote

$$\eta_{\vec{\boldsymbol{h}}}(\boldsymbol{x}^{(1:m)}) = \int f_{\boldsymbol{y}|\boldsymbol{x}}\left(\boldsymbol{y}^{(1:m)}|\boldsymbol{x}^{(1:m)}\right) \mathbb{1}_{\mathcal{H}}(\boldsymbol{y}^{(1:m)}) \mathrm{d}\boldsymbol{y}^{(1:m)}.$$

Then we can write the exact formula for $\tilde{h}\left(\boldsymbol{x}^{(1:m)}, \boldsymbol{y}^{(1:m)}\right)$, as

$$\tilde{h}\left(\boldsymbol{x}^{(1:m)}, \boldsymbol{y}^{(1:m)}\right) = \left(\prod_{i=1}^{m} f_i\left(\boldsymbol{x}^{(i)}\right)\right) \cdot \eta_{\vec{\boldsymbol{h}}}(\boldsymbol{x}^{(1:m)}) \cdot \frac{f_{\boldsymbol{y}|\boldsymbol{x}}\left(\boldsymbol{y}^{(1:m)}|\boldsymbol{x}^{(1:m)}\right) \mathbb{1}_{\mathcal{H}}(\boldsymbol{y}^{(1:m)})}{\eta_{\vec{\boldsymbol{h}}}(\boldsymbol{x}^{(1:m)})}, \tag{4.6}$$

If we consider $\mathcal{H}$ being linear, i.e., $\mathcal{H} := \left\{\boldsymbol{y} \in \mathbb{R}^{md} : \boldsymbol{A}\boldsymbol{y} = \boldsymbol{c}\right\}$ for some full rank matrix $A \in \mathbb{R}^{k \times md}$ and constant $\boldsymbol{c} \in \mathbb{R}^k$, then we can find a constant L such that $L\eta_{\vec{\boldsymbol{h}}}(\boldsymbol{x}^{(1:m)}) < 1$ and $\forall \boldsymbol{x}^{(1:m)}$,

$$L \cdot \eta_{\vec{\boldsymbol{h}}}\left(\boldsymbol{x}^{(1:m)}\right) = \exp\left[-\frac{1}{2T}(\boldsymbol{c} - \boldsymbol{A}\boldsymbol{x}^{(1:m)})^{\top}(\boldsymbol{A}\boldsymbol{A}^{\top})^{-1}(\boldsymbol{c} - \boldsymbol{A}\boldsymbol{x}^{(1:m)})\right] \leq 1, \tag{4.7}$$

where T is the length of the diffusion bridges.

To simulate Monte Carlo samples from (4.6), we just need to simulate $\boldsymbol{x}^{(i)}$ from each f_i and then simulate $\boldsymbol{y}$ with given $\boldsymbol{x}^{(1:m)}$ from the Gaussian density

$$\tilde{f}_{\boldsymbol{y}|\boldsymbol{x}}\left(\boldsymbol{y}^{(1:m)}|\boldsymbol{x}^{(1:m)}\right) = \frac{f_{\boldsymbol{y}|\boldsymbol{x}}\left(\boldsymbol{y}^{(1:m)}|\boldsymbol{x}^{(1:m)}\right) \mathbb{1}_{\mathcal{H}}(\boldsymbol{y}^{(1:m)})}{\eta_{\vec{\boldsymbol{h}}}(\boldsymbol{x}^{(1:m)})} \tag{4.8}$$

and finally deal with $\eta_{\vec{\boldsymbol{h}}}(\boldsymbol{x}^{(1:m)})$ via rejection sampling based on $L\eta_{\vec{\boldsymbol{h}}}(\boldsymbol{x}^{(1:m)}) \leq 1$. Now we can assemble the results and present Algorithm 23 which simulates exactly from equation (4.1).

Remark 4.2.1. *A key innovation of Algorithm 23, is that it turns the general constrained density $f_{\mathcal{H}}$ for non-Gaussian f_i to a constrained Gaussian density (Step 2 of Algorithm 23), which is easy to solve. The algorithm validates the answer from the constrained Gaussian problem via rejection sampling steps based on Langevin diffusion bridges. This explains the fact that our algorithm works very well (exact Monte Carlo algorithms) for non-Gaussian constraint problems.*

4.2.3.1 Sampling from Linearly Constrained Gaussian

Simulation from linearly constrained Gaussian distribution (4.8) has been studied in multiple papers (Vrins, 2018, Cong et al., 2017). First of all, let $\boldsymbol{Y} := [Y_1, \ldots, Y_m]^{\top}$ be a multivariate normal distribution, then $S = \sum_i Y_i$ is also normal. More importantly, $\boldsymbol{Y}|S = s$ is another multivariate normal distribution with its mean and covariance computable using the conditional Gaussian formula. Although $\boldsymbol{Y}|S = s$ will have

a non-invertible covariance matrix, some routines for sampling multivariate normal distribution will still work and return valid samples. In these cases, one can directly utilize those routines to do the constrained sampling.

Otherwise, one can achieve the sampling through the transformation method. Consider the following constrained Gaussian distribution

$$\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad \text{subject to} \quad \boldsymbol{A}X = \boldsymbol{c},\ \boldsymbol{A} \in \mathbb{R}^{k\times n},\ k < n. \tag{4.9}$$

Since the covariance matrix is always positive definite, $\boldsymbol{\Sigma}$ can be decomposed into $\boldsymbol{\Sigma} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^\top$, where $\boldsymbol{D}$ is a diagonal matrix with positive diagonal entries and $\boldsymbol{U}$ is orthogonal. Then

$$\boldsymbol{\Sigma} = (\boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}})(\boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}})^\top$$

where $\boldsymbol{D}^{\frac{1}{2}}$ is computed by taking square root of each diagonal entry of $\boldsymbol{D}$. Let $\boldsymbol{Z}$ follow a standard multivariate Gaussian distribution, then $\boldsymbol{X} \overset{d}{=} (\boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}})\boldsymbol{Z} + \boldsymbol{\mu}$. Thus the simulation problem (4.9) is equivalent to simulate

$$\boldsymbol{Z} \sim \mathcal{N}(0, \boldsymbol{I}_d) \quad \text{given that} \quad (\boldsymbol{A}\boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}})\boldsymbol{Z} = \boldsymbol{c} - \boldsymbol{A}\boldsymbol{\mu}$$

To simplify the notation, let $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}}$, $\boldsymbol{\alpha} = \boldsymbol{c} - \boldsymbol{A}\boldsymbol{\mu}$ and instead consider the following problem:

$$\boldsymbol{Z} \sim \mathcal{N}(0, \boldsymbol{I}_n) \quad \text{given that} \quad \boldsymbol{B}\boldsymbol{Z} = \boldsymbol{\alpha}$$

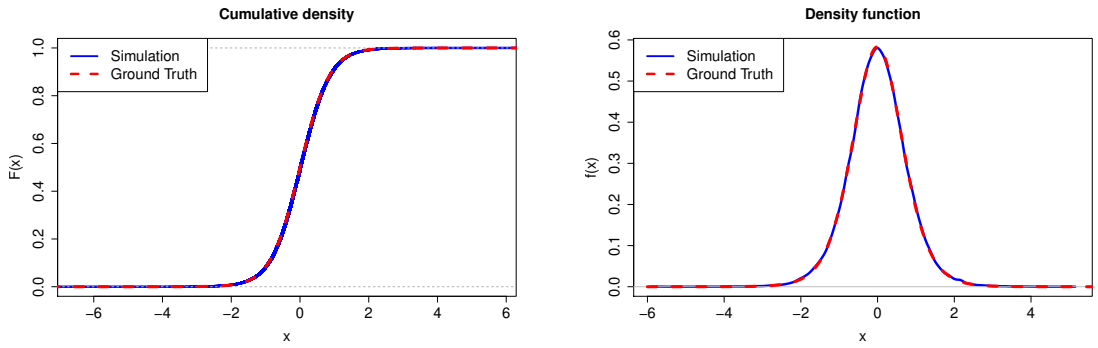
Let $\boldsymbol{B} = \boldsymbol{P}\boldsymbol{W}\boldsymbol{Q}^\top$ be a singular value decomposition of $\boldsymbol{B}$, where $\boldsymbol{P} \in \mathbb{R}^{k\times k}$, $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ are orthogonal matrices and $\boldsymbol{W} \in \mathbb{R}^{k\times n}$ is a rectangular diagonal matrix with non-negative entries on the diagonal, i.e.

$$\boldsymbol{W} = \begin{bmatrix} w_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & w_2 & \ddots & \vdots & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & w_k & 0 & \cdots & 0 \end{bmatrix}, w_i \geq 0,\ i \in 1, 2, \ldots, k$$

Then the constraint can be expressed as

$$\boldsymbol{W}(\boldsymbol{Q}^\top \boldsymbol{Z}) = \boldsymbol{P}^\top \boldsymbol{\alpha} \tag{4.10}$$

Let $\boldsymbol{Y} := \boldsymbol{Q}^\top \boldsymbol{Z} \sim \mathcal{N}(\boldsymbol{0}, \underbrace{\boldsymbol{Q}^\top \boldsymbol{I}_n \boldsymbol{Q}}_{=\boldsymbol{I}_n})$. Let y_i denote the ith element of $\boldsymbol{Y}$, and $\tilde{\alpha}_i$ denote



(a) **(b)**

Fig. 4.1 CDF and PDF of simulated data against ground truth

the ith element of $\boldsymbol{P}^\top \boldsymbol{\alpha}$. Then the constraint (4.10) is deterministic given by

$$y_i = \tilde{\alpha}_i / w_i, \quad i = 1, \ldots, k.$$

Since $\boldsymbol{Y}$ is a standard multivariate normal distribution, thus $[Y_{k+1}, \ldots, Y_n]$ conditioned on $[Y_1, \ldots, Y_k]$ is still a standard normal distribution. Note that in LCLRS (Algorithm 23), the covariance matrix $\boldsymbol{\Sigma}$ is always diagonal thus step 1 is not needed.

Algorithm 24: Sampling from a linearly constrained Gaussian distribution

Input: Target $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$;
Constraint matrix $\boldsymbol{A}$ and vector $\boldsymbol{c}$;
1 Compute singular value decomposition $\boldsymbol{\Sigma} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^\top$;
2 Compute $\boldsymbol{B} = \boldsymbol{A}\boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}}$;
3 Compute singular value decomposition $\boldsymbol{B} = \boldsymbol{P}\boldsymbol{W}\boldsymbol{Q}^\top$;
4 Compute $\tilde{\alpha} = \boldsymbol{P}^\top(\boldsymbol{c} - \boldsymbol{A}\boldsymbol{\mu})$;
5 Extract $w_1, \ldots, w_k$ from the diagonal of $\boldsymbol{W}$, k is the number of constraints or number of rows in $\boldsymbol{A}$;
6 Compute $y_i = \tilde{\alpha}_i / w_i$, for $i = 1, \ldots, k$;
7 Simulate $[y_{k+1}, \ldots, y_m] \sim \mathcal{N}_{m-k}(\boldsymbol{0}, \boldsymbol{I}_{m-k})$;
8 Set $\boldsymbol{Y} = [y_1, \ldots, y_m]$;
9 Return $\boldsymbol{X} = (\boldsymbol{U}\boldsymbol{D}^{\frac{1}{2}})\boldsymbol{Q}\boldsymbol{Y} + \boldsymbol{\mu}$;

4.2.3.2 Toy example for constrained sampling

We apply the Algorithm 23 on a relatively simple setting, in order to show that it is indeed an exact simulation algorithm. Consider the density function

$$f(x_1, x_2) \propto \left(1 + \frac{x_1^2}{3}\right)^{-2} \left(1 + \frac{x_2^2}{5}\right)^{-3} \mathbb{1}_{\{x_1 + x_2 = 0\}}(x_1, x_2) \tag{4.11}$$

which is a product of two Student-$\mathtt{T}_3(0, 1^2)$ and Student-$\mathtt{T}_5(0, 1^2)$ that is further restricted by a simple sum constraint $x_1 + x_2 = 0$.

This constraint density can be expressed as a function of solely x_1 or x_2, which has an analytical form. It easily follows

$$f(x_1) \propto \left(1 + \frac{x_1^2}{3}\right)^{-2} \left(1 + \frac{x_1^2}{5}\right)^{-3} \tag{4.12}$$

and its normalising constant can be numerically computed too.

In the simulation, Algorithm 2 is applied to sample from (4.11), gathering 10,000 samples. Using these samples, we compute the fitted density function and cumulative function using *ecdf* and *density* functions provided by R. The fitted functions are plotted against the function $f(x_1)$ computed using derived expression (4.12). The results are plotted in Fig. 4.1a and 4.1b. From the figures, we can see that the fitted densities exactly match the true curve providing a practical validation of the algorithm.

4.3 Poisson Estimator and Particle Filter

However, the performance of LCLRS may drop significantly due to low acceptance probability if the proposal is not close enough to the target function, which is the case in our study when sometimes the constraints land away from the typical set of the unconstrained target distribution. Especially when simulating time series, as the time propagates, the predictions are more likely to deviate from the constraint, causing the acceptance rate to drop as the length increases. Therefore, we consider a Particle Filter with resampling, which is often used in non-linear time series simulations (Fearnhead, 2005, De Bernardis et al., 2016), to make the simulation more practical.

To implement a particle filter, we need to compute the weight of our sampled particles which is defined by the two acceptance probability $AP_1 \times AP_2$. Under linear constraint, AP_1 is easy to compute once $\boldsymbol{x}$ and $\boldsymbol{y}$ are simulated. Although AP_2 is intractable, it is possible to derive an unbiased estimator for it. Here, we will use the method discussed in Beskos et al. (2006) and Fearnhead et al. (2008).

Let $\mathbb{W}_{(T,\boldsymbol{x},\boldsymbol{y})}$ denote the measure of a d dimensional Brownian bridge $\boldsymbol{X}_t, \{t \in [0, T]\}$ connecting $\boldsymbol{X}_0 = \boldsymbol{x}$ and $\boldsymbol{X}_T = \boldsymbol{y}$ and $f : \mathbb{R}^d \to \mathbb{R}$ be an arbitrary continuous function. Then we may define an unbiased estimator for

$$\mathbb{E}_{\mathbb{W}_{(T,\boldsymbol{x},\boldsymbol{y})}} \left[\exp\left\{-\int_0^T f(\boldsymbol{\omega}_t)dt\right\}\right] < \infty$$

given by

$$\exp\left\{(\lambda - c)T\right\} \lambda^{-\kappa} \prod_{j=1}^{\kappa} \left\{c - f(\omega_{\chi_j})\right\} \tag{4.13}$$

Algorithm 25: Particle Filter - LCLRS

input: Constraints $A\boldsymbol{y}_t = S_t$;
Unconstrained target (transition) density $f(\boldsymbol{y}_t|\boldsymbol{y}_{t-1})$;
Parameter T;
Total number of particles n;
Length of the time series τ;

Initialize particles $\boldsymbol{y}_0^{[i]}$ with weights $w_0^{[i]}$;
for $t = 1, 2, \ldots, \tau$ **do**
  for $i = 1 \ldots, n$ **do**
    Simulate the starting points $\boldsymbol{x}_t^{[i]} \sim f(\cdot|\boldsymbol{y}_{t-1}^{[i]})$;
    Simulate the particle $\boldsymbol{y}_t^{[i]} \sim \mathcal{N}(\boldsymbol{x}_t^{[i]}, T\mathbf{I}_d)$ constrained on $\boldsymbol{A}\boldsymbol{y}_t^{[i]} = S_t$;
    Accept the proposal with probability AP_1 given by equation (4.7);
    Simulate layer I for the BB, and choose λ, c;
    Generate a set time points $\{\chi_k\}_{k=1,\ldots,\kappa}$ for the Poisson Estimator to be evaluated on;
    Simulate a BB in layer I of time T connecting $\boldsymbol{x}_t^{[i]}$ and $\boldsymbol{y}_t^{[i]}$ at the time points $\{\chi_k\}_{k=1,\ldots,\kappa}$;
    Calculate AP_2 using the Poisson estimator given by (4.13);
    Update weight $w_t^{[i]} = w_{t-1}^{[i]} \cdot AP_2$;
  end
  Compute the Effective Sample Size $\text{ESS}_t = \left(\sum_i w_t^{[i]}\right)^2 / \left(\sum_i (w_t^{[i]})^2\right)$;
  if $ESS_t \leq n/10$ **then**
    Conduct resampling;
  end
end

where $c \in \mathbb{R}$, $\lambda > 0$, $\kappa \sim \text{Po}(\lambda t)$, $\chi_j \sim \mathcal{U}[0, T], j = 1, \ldots, \kappa$ and ω being a sample element of $\mathbb{W}_{(T,\boldsymbol{x},\boldsymbol{y})}$. This is called the Poisson estimator which is an unbiased estimator for the expectation above (Beskos et al., 2006). A recommended choice for the parameters is $c = \sup f$ and $\lambda = c - \inf f$. Thus the direct application of (4.13) is not very reliable when f is not bounded.

Thus, we apply the layered approach to construct a generalized Poisson estimator. The state space of $\mathbb{W}_{(T,\boldsymbol{x},\boldsymbol{y})}$ is partitioned into $\{L_1, L_2, \ldots\}$ and each L_i defines a bound for which the BBs lie in. Since the layer of the proposed Brownian bridge is known before we compute the Poisson estimator, we can determine $c = \sup \phi$ using layer I and continue the computation. Now, both rejection steps in Algorithm 23 can be replaced by assigning weight to the current sample, and Algorithm 23 can be extended to an importance sampling. With (4.7) and the Poisson estimator, a particle filter approach can be outlined as in Algorithm 25.

The PF approach no longer requires any rejection steps since AP_1 and AP_2 are computed numerically and used to update the weights. As presented in Algorithm 25,

the first rejection step is kept since it impedes minimal computational cost. We only use the acceptance probability in the second rejection stage AP_2 to update the weights. At each time step t, we need to sample/propagate a set of n particles from the previous step $t-1$. After sampling all particles of time t, the Effective Sample Size (ESS), ESS $= \left(\sum_i w_i\right)^2 / \left(\sum_i w_i^2\right)$, is computed to check if re-sampling is required. In practice, we used $n/10$ as the threshold for resampling as we usually find the batch of particles with ESS above that threshold to give reasonable estimates. In some of the literature, $n/2$ is used as the threshold (Doucet et al., 2009). Resampling can be done by various strategies. In this case, a categorical distribution is constructed using the weights and n samples are drawn to decide which particles are passed on, also referred to as the multinomial resampling in Douc and Cappé (2005).

4.4 Mean Squared Error Analysis

To understand the effect of imposing a linear constraint on an unconstrained model, we conduct a short theoretical study on Gaussian distributions where we can analytically compute the outcomes. In addition, we conclude, in the Gaussian case, why and when adding linear constraints can improve the accuracy of estimation. Non-Gaussian models are compared through simulation studies in the end. For simplicity, we will focus on a single-step model without considering time-series data.

Consider a simple linear regression setting. Let $Y^{(i)} \sim p(\cdot)$, $i = 1, \cdots, n$, to be n independent draws from the true population. Through the linear model, each draw of $Y^{(i)}$ has a corresponding estimator $\hat{Y}^{(i)}$ with $\hat{Y}^{(i)} \sim \mathcal{N}(\hat{\mu}_i, \hat{\sigma}_i^2)$ where $\hat{\mu}_i$ is the fitted mean of $Y^{(i)}$. Here the predictor $\hat{Y}^{(i)}$ does not need to be a sensitive predictor of $Y^{(i)}$, i.e. $\hat{Y}^{(i)}$ of $Y^{(i)}$ that may have a large bias or uncertainty. Note that the sum

$$\hat{S} := \sum_{i=1}^{n} \hat{Y}^{(i)},$$

is also Gaussian. Thus, the joint distribution of $(\hat{Y}_1, \ldots, \hat{Y}_n, \hat{S})$ is still a multivariate normal distribution with mean and variance are given by

$$\hat{\boldsymbol{\mu}} = \left[\hat{\mu}_1, \ \cdots, \ \hat{\mu}_n, \ \textstyle\sum_{i=1}^{n} \hat{\mu}_i\right]^\top, \quad \hat{\boldsymbol{\Sigma}} = [\boldsymbol{D}, \boldsymbol{V}; \boldsymbol{V}^\top, w^2]$$

where

$$\boldsymbol{D} := \text{diag}(\hat{\sigma}_1^2, \cdots, \hat{\sigma}_n^2), \quad \boldsymbol{V} := \left[\hat{\sigma}_1^2, \cdots, \hat{\sigma}_n^2\right]^\top, \quad w^2 := \sum_{i=1}^{n} \hat{\sigma}_i^2.$$

Let $\hat{\boldsymbol{Y}}$ denote the random vector of $(\hat{Y}_1, \ldots, \hat{Y}_n)$, then its distribution conditioned on

the constraint is $\hat{\boldsymbol{Y}} \left| \hat{S} = s \sim \mathcal{N}\left(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*\right) \right.$ where

$$\begin{aligned} \boldsymbol{\mu}^*(s) &= \left[\hat{\mu}_1 + \frac{\hat{\sigma}_1^2}{w^2}(s - \sum_{i=1}^{n} \hat{\mu}_i), \cdots, \hat{\mu}_n + \frac{\hat{\sigma}_n^2}{w^2}(s - \sum_{i=1}^{n} \mu_i) \right]^\top, \\ \boldsymbol{\Sigma}^*(s) &= \boldsymbol{D} - \left[\frac{\hat{\sigma}_i^2 \hat{\sigma}_j^2}{w^2} \right]_{i,j}^{1,n}. \end{aligned} \tag{4.14}$$

We examine the forecasting (imputation) performance in three aspects:

1. *Signed mean deviation* of each predictor $\hat{Y}^{(i)}$ computed by $\alpha_i := \hat{\mu}_i - \hat{Y}^{(i)}$
2. *Uncertainty* of the predictors Var$(\hat{Y}^{(i)})$
3. *Mean-squared error* (MSE) $\mathbb{E}\left[||\hat{\boldsymbol{Y}} - \boldsymbol{Y}||^2\right]$

where MSE, which measures both mean deviation and uncertainty, is a more comprehensive evaluation of the performance.

The potential improvement in MSE if the sum of the true values is incorporated into the model is given by:

$$\begin{aligned} &\mathbb{E}\left[||\hat{\boldsymbol{Y}} - \boldsymbol{Y}||^2\right] - \mathbb{E}\left[||\hat{\boldsymbol{Y}} - \boldsymbol{Y}||^2 \middle| S = \sum_{i=1}^{n} Y^{(i)} \right] \\ &\qquad = \underbrace{\sum_{i=1}^{n} \alpha_i^2 - \sum_{i=1}^{n} \left(\alpha_i - \frac{\hat{\sigma}_i^2}{w^2} \sum_{j=1}^{n} \alpha_j \right)^2}_{\Psi_1} + \underbrace{\frac{1}{w^2} \sum_{i=1}^{n} (\hat{\sigma}_i^4)}_{\Psi_2} \end{aligned} \tag{4.15}$$

where $\alpha_i = \hat{\mu}_i - Y^{(i)}$ is the mean deviation from the true value for the i-th predictor. We can deduce the following Propositions from equation (4.15).

Proposition 4.4.1 (Garaunteed Uncertainty Reduction)**.** *The sum of constrianed variance tr*(Σ^*) *is always less than the unconstrained variance with a reduction of*

$$tr(\boldsymbol{\Sigma}^*) - \sum_{i=1}^{m} \hat{\sigma}_i^2 = \frac{1}{w^2} \sum_{i=1}^{m} \hat{\sigma}_i^4$$

where $w^2 = \sum_i \hat{\sigma}_i^2$.

Proof. Follows directly from (4.15) since Ψ_2 denotes exactly the difference in uncertainty and $\Psi_2 \geq 0$. □

It is important to note that, we made no assumption about the imputed value $\hat{\boldsymbol{Y}} = (\hat{Y}^{(1)}, \cdots, \hat{Y}^{(m)})$, except that it is independent in its components and is Gaussian. If we could also bound its error, i.e. having a reasonable statistical model, then we can deduce a stronger result as follows:

Proposition 4.4.2 (Uncertainty Domination)**.** *Let $\alpha_i := Y^{(i)} - \hat{\mu}_i$ denote the signed mean deviation of $i-$th predictor from the true value and S_d denote the sum of α_i. Suppose that the predictors are reasonable that*

$$\exists M > 0 \text{ such that } |\alpha_i| \leq \lambda_i M, \text{ and also } w^2 \geq 2S_d M,$$

then the constrained model always has a non-negative improvement in mean-squared error compared with the unconstrained model.

Proof. We try to bound Ψ_1 in (4.15). Firstly recall that

$$S_d = \sum_{i=1}^{m} \alpha_i$$

Without loss of generality, let $S_d \geq 0$, then

$$\begin{aligned} \Psi_1 &= -2S_d \sum_{i=1}^{m} \lambda_i \alpha_i + \sum_{i=1}^{m} \lambda_i^2 S_d^2 \\ &\geq -2S_d M \sum_{i=1}^{m} \lambda_i^2 \end{aligned}$$

The second line follows by applying the bound on α_i. Note that $\Psi_2 = w^2 \sum_{i=1}^{m} \lambda_i^2$. Thus

$$\Psi_1 + \Psi_2 \geq w^2 \sum_{i=1}^{m} \lambda_i^2 - 2S_d M \sum_{i=1}^{m} \lambda_i^2 \geq 0$$

□

The main takeaway from Proposition 4.4.2 is actually very similar to Proposition 4.4.1, namely, it is most appropriate to apply a constrained model when the original model has a higher uncertainty compared to its expected error.

Remark 4.4.1. *We can observe from Ψ_1 the formula for the constrained mean deviation, namely $\alpha_i - \frac{\hat{\sigma}_i^2}{w^2} \sum_j \alpha_j$. Thus the correction on mean deviation is in the favorable direction only for the components whose α_i has the same sign as S_d the overall deviation from the constraint. Ideally, we are expected to see the mean deviation to improve in all dimensions only if all components have over(under)-estimated the target. However, it is still possible for components with small α_i but large relative variance λ_i to be over-corrected and end up with a worse mean deviation.*

Remark 4.4.2. *When the bias term Ψ_1 is not dominated by the variance term Ψ_2, then Ψ_1 can be negative and consequently rendering (4.15) negative. Ψ_1 can be viewed as*

a quadratic function of α_i for every i and when the leading coefficient is negative, the function is more likely to take a value below zero. One sufficient condition for this to happen is when $\lambda_i := \frac{\hat{\sigma}_i^2}{w^2} < 1 - \sqrt{(n-1)/n}$.

Remark 4.4.3. *One special case is when $\lambda_1 = \cdots = \lambda_n$, in which case Ψ_1 is guaranteed to be non-negative and the constrained model is always better than the unconstrained model in terms of MSE.*

4.4.1 Effect of relative variance on MSE

Since there is a guaranteed improvement in the variance component Ψ_2, it is interesting to analyze what happens if Ψ_1 is not dominated by Ψ_2. Define the following

$$\lambda_i = \hat{\sigma_i}^2/w^2, \qquad \Lambda := \sum_{i=1}^{n} \lambda_i^2$$

$$\begin{aligned}
\Psi_1(\boldsymbol{\alpha}) &= \sum_{i=1}^{n} \alpha_i^2 - \sum_{i=1}^{n} \left(\alpha_i - \frac{\sigma_i^2}{w^2} \sum_{j=1}^{n} \alpha_j \right)^2 \\
&= 2(\sum_{i=1}^{n} \alpha_i) \sum_{i=j}^{n} \alpha_j \lambda_j - \left(\sum_{i=1}^{n} \alpha_i \right)^2 \sum_{j=1}^{n} \lambda_j^2 \\
&= 2 \sum_{i=1}^{n} \alpha_i^2 \lambda_i + 2 \sum_{1 \le i < j}^{n} \alpha_i \alpha_j (\lambda_i + \lambda_j) - \sum_{i=1}^{n} \lambda_j^2 \left(\sum_{i=1}^{n} \alpha_i^2 + \sum_{1 \le i < j}^{n} 2\alpha_i \alpha_j \right) \\
&= \sum_{i=1}^{n} (2\lambda_i - \Lambda) \alpha_i^2 + 2 \sum_{1 \le i < j}^{n} (\lambda_i + \lambda_j - \Lambda) \alpha_i \alpha_j
\end{aligned}$$

Take α_1 for example, the roots lie at

$$\alpha_1 = \frac{1}{2(\lambda_1 - \Lambda)} \left(-2 \sum_{i=2}^{n} (\lambda_1 + \lambda_i - \Lambda) \alpha_i \pm \sqrt{\Delta} \right)$$

where

$$\Delta = \left(\sum_{i=2}^{n} (\lambda_1 - \lambda_i) \alpha_i \right)^2 \ge 0.$$

Thus, the equation has a repeated root if and only if $\lambda_1 = \cdots = \lambda_n$. In other words, depending on the mean deviation α_i, it is almost always possible for Ψ_1 to be negative and have a negative improvement for the overall MSE.

4.4.2 Simulation of Gaussian Case

A simulation is conducted to demonstrate the analysis made above, see Figure 4.1. A total of 100,000 samples are generated from both the constrained and unconstrained models in each setting. The settings differ by their model deviation $\boldsymbol{\alpha}$ and model uncertainty $\boldsymbol{\sigma^2}$. In the first case, the mean deviation is dominated by the variance and we see improvements in all three predictors (or called imputed values). For the second and third settings, the variances are kept the same but the mean deviations are increased. We can see the variance improvement are the same but MSE improvements are not all positive. However, the total improvement in model MSE is still positive, mainly because the correction for the third component which has the largest mean error has a large positive effect on the model MSE. In these two cases, the main contribution to improvement in overall MSE is no longer the reduction in variance (uncertainty).

Finally, for cases 4 and 5, we examine the situation when α_i are large but σ_i^2 are small. Note that for the fourth case, the first predictor has a very small relative variance and is below the $1 - \sqrt{(n - 1/n)}$ threshold, thus the overall MSE improvement is negative. This is different in the fifth case when the variance in all components is the same, which matches the condition in Remark 4.4.3 and we observe a small but positive improvement in MSE.

4.4.3 Simulation of Non-Gaussian case

When the modeling distribution is non-Gaussian, the constrained distribution becomes intractable and hence there is no analytic formula for measuring the MSE improvement. We have done the same simulation on gamma distribution and the Student's t-distribution cases, see also Figure 4.1, to compare with different target distributions. $\boldsymbol{\alpha}$ and $\boldsymbol{\sigma^2}$ still represent the mean deviation and variance respectively, and $\boldsymbol{\psi}$ denotes the shape parameters.

For the Gamma cases, we imitate the situation in Study 1 and set the true value of $\boldsymbol{Y}^{(1:3)}$ to $(4,\ 20,\ 25)$ and kept the shape parameter fixed to 4 in all predictors. In such a way, the uncertainty of the predictors will vary with their mean as in the real case. In both cases, we see positive improvements in the overall MSE, whereas the first case is mainly contributed by the improvement in uncertainty. We see an increase in uncertainty for the second case since the uncertainty varies with the model mean and the last two unconstrained predictors underestimate the true value by about half. Thus the constrained counterparts, which have larger means, also have larger uncertainties compared with the unconstrained predictors. Thus in the second case, we see the MSE improvement comes solely from the correction in the bias of the unconstrained predictor. Note that although the unconstrained predictors have smaller uncertainty, they also have a large bias, which means the uncertainty may have been underestimated.

The Gamma cases exhibit a different behavior compared to the Gaussian case where the uncertainty is not guaranteed to improve, so we also run the setup for generalized Student's t-distribution where the distribution is also symmetric, like the Gaussian distribution. The degrees of freedom are fixed to 5 and the density function can be expressed as

$$f(x;\nu,\mu,\sigma^2) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\Gamma(\frac{\nu}{2})}\left(1+\frac{(x-\mu)^2}{\nu\sigma^2}\right)^{-\frac{\nu+1}{2}}, \quad \nu = 5$$

The four cases examined for Student's t-distribution use the same parameter setting as cases 2-5 in the Gaussian simulation and the results almost match case by case. We see similar improvements in deviation estimation but overall larger improvements in uncertainty when applying a constraint to Student's t-distribution compared with the Gaussian cases. This is reasonable since Student's t-distribution has a heavier tail than the Gaussian distribution, and hence a larger uncertainty when the scaling parameter σ^2 is the same.

Gaussian	MSE Improv.			Devi. Improv.			Var. Improv.		
$\boldsymbol{\alpha} =$ (0.1 -1 2) $\boldsymbol{\sigma}^2 =$ (1 4 10)	[0.08	0.39	9.06]	[0.07	**-0.29**	0.73]	[0.33	1.34	3.34]
$\boldsymbol{\alpha} =$ (0.1 -2 4) $\boldsymbol{\sigma}^2 =$ (1 4 10)	[0.08	**-1.49**	15.9]	[0.06	**-0.56**	1.39]	[0.33	1.34	3.34]
$\boldsymbol{\alpha} =$ (1 -3 8) $\boldsymbol{\sigma}^2 =$ (1 4 10)	[0.70	**-11.1**	54.7]	[0.40	**-1.60**	4.00]	[0.33	1.34	3.34]
$\boldsymbol{\alpha} =$ (10 -3 -5) $\boldsymbol{\sigma}^2 =$ (1 2 3)	[6.78	**-3.80**	**-9.51**]	[0.34	**-0.67**	**-1.00**]	[0.33	0.67	1.00]
$\boldsymbol{\alpha} =$ (10 -3 -5) $\boldsymbol{\sigma}^2 =$ (2 2 2)	[13.6	**-3.79**	**-6.46**]	[0.67	**-0.67**	**-0.67**]	[0.67	0.67	0.67]
Gamma (shape fixed)									
$\boldsymbol{\alpha} =$ (0.1 -2 4) $\boldsymbol{\psi} =$ (4 4 4)	[**-0.12**	30.0	168.4]	[**-0.09**	**-0.11**	2.11]	[**-0.09**	30.5	155.7]
$\boldsymbol{\alpha} =$ (1 -10 -12) $\boldsymbol{\psi} =$ (4 4 4)	[**-12.1**	69.9	137.6]	[**-1.91**	6.80	11.7]	[**-4.67**	**-19.8**	**-6.22**]
Student's t (ν=5)									
$\boldsymbol{\alpha} =$ (0.1 -2 4) $\boldsymbol{\sigma}^2 =$ (1 4 10)	[0.28	0.26	21.2]	[0.02	**-0.60**	1.33]	[0.28	3.02	12.2]
$\boldsymbol{\alpha} =$ (1 -3 8) $\boldsymbol{\sigma}^2 =$ (1 4 10)	[0.70	**-11.8**	57.3]	[0.46	**-1.73**	3.82]	[0.01	1.48	10.6]
$\boldsymbol{\alpha} =$ (10 -3 -5) $\boldsymbol{\sigma}^2 =$ (1 2 3)	[7.85	**-3.09**	**-7.32**]	[0.38	**-0.69**	**-0.94**]	[0.45	1.53	2.98]
$\boldsymbol{\alpha} =$ (10 -3 -5) $\boldsymbol{\sigma}^2 =$ (2 2 2)	[14.3	**-2.90**	**-5.55**]	[0.66	**-0.66**	**-0.67**]	[1.53	1.53	1.54]

Table 4.1 Improvements in accuracy when adding sum constraint for different cases. Improvement in MSE, deviation and variance for all three components of the model are listed with positive values marked by an underscore and negative values marked in bold.

CHAPTER 5

Applications of Constrained Density Fusion

5.1 Time Series Disaggregation

From the previous chapter, we have seen an extension of the fusion sampler on linear constraints. Such constraints appear naturally when we deal with time series data, more specifically, time series that represents the accumulated quantity within a unit of time. For instance, water usage in a day is the sum of the usage for all the hours in the day, which means the sequence of 24 one-hour measures from in the day will sum to the daily reading of that day. In this section, we will primarily work on electricity usage data as a typical example of accumulated readings.

5.1.1 Background

In the current energy transition to fully autonomous and smart energy systems, an energy supplier usually has customers with different types of meters installed (Meng et al., 2018), e.g., customers with smart meters (record energy consumption from every hour to every minute), customers with time-of-use meters (e.g., Economy 7 in the UK that a day can be divided into two time periods and the meter records aggregated consumption over the two periods), and customers with traditional meters (record aggregated consumption). It poses a great challenge for the energy suppliers to fully understand energy customers' consumption patterns, especially for the latter two types of customers with conventional meters in the absence of high time resolution meter data. Even it is supposed to be easy to know the detailed and high-resolution energy consumption of customers with smart meters, the smart meter data may still be subject to delays and lower reliability (Peppanen et al., 2016), or being aggregated to preserve customers' privacy. For example in the UK, the smart meter data that the distributed network operators receive will be the aggregated reading without the real-time data (Poursharif et al., 2017). Therefore, the supplier often has low-resolution data for some (usually recent) periods, but possibly high-resolution data for the periods before. For those days with missing high-resolution data from a customer, nonetheless, the energy supplier may still want to know the more fine-grained energy consumption of such customers. For

instance, knowing consumption during peak time periods for customers with traditional meters or consumption during each hour for customers with time-of-use meters helps understand energy usage behaviors, which is essential to transform the energy systems in industrialized countries in order to reduce the total energy consumption (Burger et al., 2015).

A similar problem can be found in power distribution networks where a grid operator would like to understand when spikes of energy demand could occur. This study would require a continuous recording of energy usage in the network at a fine-grained level. Such detailed monitoring needs the installation of additional equipment and storage devices which can be expensive. However, such expenses can be avoided if one can reasonably predict the peak and trough measurements in each time period given the low-frequency data.

Overall, such kind of imputation problem often arises in studying time series data, where each data point represents quantity aggregated for each low-resolution time period, e.g., time series that record generation/consumption. The prediction of high-resolution data from low-resolution data is often named as ***disaggregation*** in some related literature (Wang et al., 2020, Rafsanjani et al., 2020). In some cases where the data set is about a single household, this problem is often simply formulated as ***data imputation*** problem. Throughout this paper, we refer to such problems as *disaggregation* or *imputation* in general, but when talking about the statistical models without constraint we still refer to them as *forecasting models* or *predictors*, and we refer to the high-resolution data estimated from the low-resolution data as *imputed data*.

5.1.2 Imputation Framework

Let $\boldsymbol{Y}_j = (Y_j^{(1)}, \cdots, Y_j^{(m)})$ denote the high-resolution data for time period j, where the high-resolution data is the result of naturally dividing each low-resolution data into m readings. Our target is to impute the missing high-resolution data $\boldsymbol{Y}_j$ for time period j from the existing data set $\mathcal{D}_j = \{\boldsymbol{Y}_k, k = 1, \cdots, j-1\}$ and a set of additional covariates $\boldsymbol{\Xi}_j$ containing information related to $\boldsymbol{Y}_j$, under a linear constraint, for example $\sum_{i=1}^m Y_j^{(i)} = S_j$. Here S_j corresponds to the low-resolution aggregated reading for time period j and it is available when we impute the high-resolution readings. Therefore, if we denote the imputed values as $\hat{\boldsymbol{Y}}_j = (\hat{Y}_j^{(1)}, \cdots, \hat{Y}_j^{(m)})$, it must also satisfy the constraint $\sum_{i=1}^m \hat{Y}_j^{(i)} = S_j$ too. In one of the application problems of this paper, we consider $m = 3$, i.e., peak time period (evening), off-peak time period (midnight), and day-time, since these are of most interest to electricity providers, and different tariffs are often made on these time periods. The data vector $\boldsymbol{Y}_j = (Y_j^{(1)}, \cdots, Y_j^{(m)})$ follows a density $\prod_i f(y_j^{(i)}|\boldsymbol{\theta}, \mathcal{D}_j, \boldsymbol{\Xi})$, and $Y_j^{(i)}, i = 1, \cdots, m$ are assumed to be independent conditioned on all historical data and covariates.

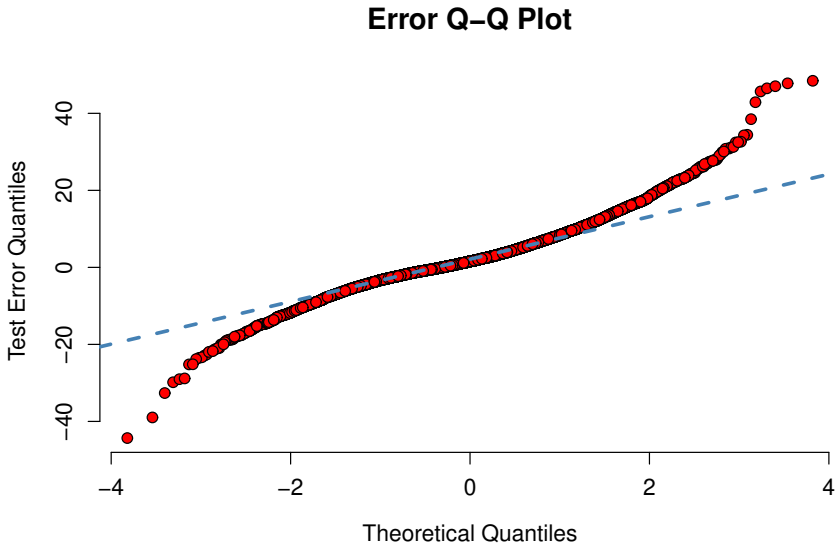


Fig. 5.1 Residual distribution plot based on AR model for the Irish Smart Meter Trial dataset.

Such imputation problems are very simple if f is Gaussian. For example, under linear regression models with Gaussian residuals, the response variable $\boldsymbol{Y}$ and its forecasting value $\hat{\boldsymbol{Y}}$ will follow Gaussian distributions, with or without the constraint $\sum_{i=1}^{m} \hat{Y}^{(i)} = S$. This kind of problem may exist in other research areas, such as Huang et al. (2017) where an anti-radiation missile follows an almost linear trajectory and an approximated linear constraint is evaluated and imposed on a Kalman Filter, which is a filter based on updating the Gaussian belief of the system, to produce a robust tracking performance.

However, if f is not Gaussian, how to solve such a problem is nontrivial since under the constraint $\sum_{i=1}^{m} \hat{Y}^{(i)} = S$, the mean and covariance matrix $\mathbb{E}[\hat{\boldsymbol{Y}}|S]$ and $\text{Var}(\hat{\boldsymbol{Y}}|S)$, and the density of $\hat{\boldsymbol{Y}}$ may not have an analytical solution. For electricity consumption data, the residual distribution based on time series models usually will not be Gaussian because of the extreme values and the heavier tail of the data (for example, due to abnormal weather conditions). Considering the dataset used in Section 5.1.4, the Irish Smart Meter Trial data, we fitted an auto-regressive time series model using the 2009 autumn season data (to avoid seasonal component) and the fitting error is presented in Figure 5.1 as a quantile-quantile plot where the error points are plotted against its normal estimation. It is clear from the graph that the Gaussian assumption for residuals is not appropriate (also evidenced by the Shapiro-Wilk test having a p-value $< 2.2\text{e}^{-16}$).

5.1.3 Basic Time Series Model

Consider two parallel time series $\{S_t\}_{t=1,2,3,\ldots}$ and $\{Y_{t,i}\}_{t=1,2,3,\ldots}^{i=1,2,\ldots,k}$ where S_t is the low-frequency data and $Y_{t,i}$ is recorded at k times the frequency of S_t with index ordering

$(t-1,k) < (t,1) < (t,2) < \cdots < (t,k) < (t+1,1)$. For the rest of this section, we will denote the high-frequency data as $\{Y_t^{(1:k)}\}_{t=1,2,\ldots}$ to show that the time series $\{Y_t^{(1)}\}$ to $\{Y_t^{(k)}\}$ are treated as independent time series and modeled using k separate models.

Remark 5.1.1. *While treating modeling high-frequency data using separately time series might seem to be imposing a strong independence assumption. This, however, can be acceptable if the separation is made with respect to the seasonality of this particular time series. For instance, in energy consumption data, seasonality usually can be observed in the cycle length of one day. In which case, if every observation of the j-th time series $Y_t^{(j)}$ corresponds to the reading of the same j-th time slot in each cycle, the set of independent time series models will still perform well while having one seasonal component removed. In addition, this approach is also effective if the time series has multiple seasonality since removing one seasonality can greatly reduce the complexity of the model. Note here that energy consumption data usually also admits one year as the cycle length.*

5.1.3.1 Autoregressive Model with Gamma Link

In general, we will use the Autoregressive (AR) model to fit each high-frequency time series $Y_t^{(j)}$, $j = 1, \ldots, k$. In the same experiment, the k AR models will share the same model structure, e.g., model order K, choice of additional regressors, but can have distinct model parameters.

Referring back to Fig. 5.1, it is clear that a vanilla AR model $Y_t^{(j)}$ is not suitable for our data, since the model residue is clearly not Gaussian. In addition, energy consumption data are non-negative and positively skewed, so one may need to turn towards some non-Gaussian distributions to model the error term $\epsilon_t^{(j)}$. Here we use Gamma distribution to cover the non-negativity and positive skewness (Manning et al., 2005). Now, the AR model (of order K) with Gamma link can be expressed as:

$$Y_t^{(j)} \sim \Gamma\left(\psi^{(j)}, \frac{\psi^{(j)}}{\mu_t^{(j)}}\right), \qquad \mu_t^{(j)} = \sum_{r=1}^{K} \Phi_r^{(j)} Y_{t-r}^{(j)} + \boldsymbol{\Xi}_t \boldsymbol{\beta}^{(j)} \tag{5.1}$$

for some unknown shape parameter $\psi^{(j)}$, and here the mean $\mu_t^{(j)}$ is subject to the constraint $\mu_t^{(j)} > 0$. Our link function is the identity function, which is usually not the canonical choice since the mean of our predictor is still linear. However, this is how one would expect the time series to work, i.e., having a linear relation in the covariates $\boldsymbol{\Xi}_t$ and lagged terms $Y_{t-r}^{(j)}$.

In our experiments, the model parameters $\boldsymbol{\Phi}^{(j)}$ and $\boldsymbol{\beta}^{(j)}$ are fitted using the least-squares method while the shape parameters $\psi^{(j)}$ are chosen to be the maximum likelihood estimate. Let τ denote the total number of days used in the training model. We

first fit $\boldsymbol{\Phi}^{(j)} = \left(\Phi_1^{(j)}, \ldots, \Phi_K^{(j)}\right)$, and $\boldsymbol{\beta}^{(j)}$ by solving the ordinary least square equation, i.e., for each $(j) \in \{1, \ldots, k\}$, solve

$$\underset{\boldsymbol{\Phi}^{(j)}, \boldsymbol{\beta}^{(j)}}{\operatorname{argmin}} \sum_{t=1}^{\tau} \left[Y_t^{(j)} - \mu_t^{(j)}\right]^2, \quad \mu_t^{(j)} = \sum_{r=1}^{K} \Phi_r^{(j)} Y_{t-r}^{(j)} + \boldsymbol{\Xi}_t \boldsymbol{\beta}^{(j)}$$

where the evaluation of the cost function is done by directly replacing $Y_{i,t}^{(j)}$ by its observed value. Then fixing $\mu_{i,t}^{(j)}$, fit $\psi^{(j)}$ by maximising the log-likelihood function,

$$\begin{aligned}\underset{\psi^{(j)}}{\operatorname{argmax}}\ \ell(\psi^{(j)}; \boldsymbol{\Xi}, \boldsymbol{Y}^{(j)}) = \sum_{t=1}^{\tau} \Bigg[& \psi^{(j)} \log(\psi^{(j)}/\mu_t^{(j)}) - \log\Gamma(\psi^{(j)}) + \\ & (\psi^{(j)} - 1) \log\left(\psi^{(j)} Y_t^{(j)}/\mu_t^{(j)}\right) - \left(\frac{\psi^{(j)} Y_t^{(j)}}{\mu_t^{(j)}}\right)\Bigg]. \quad (5.2)\end{aligned}$$

Remark 5.1.2. *The optimal way to fit the parameters is to solve the maximum likelihood estimators for $\boldsymbol{\Phi}^{(j)}$, $\boldsymbol{\beta}^{(j)}$ and $\psi^{(j)}$ simultaneously by maximizing the full log-likelihood function, i.e., every $\mu_t^{(j)}$ in (5.2) need to be treated as a function of $\boldsymbol{\Phi}^{(j)}$ and $\boldsymbol{\beta}^{(j)}$. This optimization problem has a much higher dimension and could lead to stability issues. In addition, the model choice and fitting are not the primary focus of this study, thus we used a sub-optimal estimator that can be easily handled.*

5.1.3.2 Constrained Imputation

Given the parameter estimates $\left(\hat{\boldsymbol{\Phi}}^{(1:k)}, \hat{\boldsymbol{\beta}}^{(1:k)}, \hat{\boldsymbol{\psi}}^{(1:k)}\right)$, we proceed to state the constrained model. Let $Y_t^{(1:k)}$, $t = 1, 2, \ldots, \hat{t}$ be the high-resolution energy consumption we want to impute with respect to low-resolution time series data $\{S_t\}_{t=1,\ldots,\hat{t}}$ given. By construction of our AR model, every time point $\boldsymbol{Y}_t$ depends only on its previous times $\boldsymbol{Y}_{t-1}, \boldsymbol{Y}_{t-2}, \ldots$, and the constraint $\sum_{j=1}^{k} Y_t^{(j)} = S_t$. Then given its explanatory variables at time t $\boldsymbol{\Xi}_t$, the density of $\boldsymbol{Y}_t$ conditional on the past is

$$f(\boldsymbol{Y}_t | \boldsymbol{Y}_{t-K:t-1}) = \prod_{j=1}^{k} f\left(Y_t^{(j)} | Y_{t-K:t-1}^{(j)}\right) \mathbb{1}_{\left\{\sum_{j=1}^{k} Y_t^{(j)} = S_t\right\}} \quad (5.3)$$

where the likelihood function is from the Gamma model

$$f\left(Y_t^{(j)} \,\middle|\, Y_{t-K:t-1}^{(j)}\right) = \frac{\left(\hat{\psi}^{(j)}/\hat{\mu}_t^{(j)}\right)^{\hat{\psi}^{(j)}}}{\Gamma\left(\hat{\psi}^{(j)}\right)} \left(Y_t^{(j)}\right)^{\hat{\psi}^{(j)}-1} \exp\left(-\frac{\hat{\psi}^{(j)} Y_t^{(j)}}{\hat{\mu}_t^{(j)}}\right) \quad (5.4)$$

and

$$\hat{\mu}_t^{(j)} = \sum_{r=1}^{K} \hat{\Phi}_r^{(j)} Y_{t-r}^{(j)} + \boldsymbol{\Xi}_t \hat{\boldsymbol{\beta}}^{(j)}. \tag{5.5}$$

As we can see, for the simulation of each $\boldsymbol{Y}_t$, the target distribution (5.3) follows the shape of (4.1) where each factored density $f\left(Y_t^{(j)} | Y_{t-K:t-1}^{(j)}\right)$ can be simulated with a linear constraint added across the variables of interest. Thus Algorithm 23 (LCLRS) can be applied directly to this simulation problem by applying it sequentially in temporal order $\boldsymbol{Y}_1^{(1:k)} \to \boldsymbol{Y}_2^{(1:k)} \to \boldsymbol{Y}_3^{(1:k)} \to \cdots \to \boldsymbol{Y}_D^{(1:k)}$. Following Algorithm 25, to simulate time point $\boldsymbol{Y}_t$, one would first draw a sample $\boldsymbol{x}$ from (5.3) without the constraint as the starting points of the Brownian bridge. Then sample $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{x}, T\mathbf{I}_k)$ subject to sum constraint $\mathbb{1}_{\{||\boldsymbol{y}||_1 = S_t\}}$ which can be vectorized as $\boldsymbol{A}\boldsymbol{y} = S_t$, where A is a row vector of ones. Then the sampled particle will have its weight multiplied by AP_2 given by (4.4).

5.1.4 Study 1: Day-readings Disaggregation

In this section, we consider a problem that electric utility companies encounter. Modern time-of-use meters are capable of reporting electric usage for three periods per day (peak time, non-peak time, and midnight). However, such high-resolution meter readings may be missing due to unreliability or delay, etc., and on such days we only obtain one reading per day as from the traditional meters. We are interested in recovering their energy consumption in each time period from the aggregated consumption.

We use the dataset from the Irish Smart Meter Trial, which includes half-hourly energy consumption readings of individual residential smart meters from July 2009 to the end of 2010 and corresponding questionnaire data of residential customers including the social-economic data of occupants, the size of household, appliances information, etc. We processed the data to consider the problem of disaggregating the daily readings into tridaily readings.

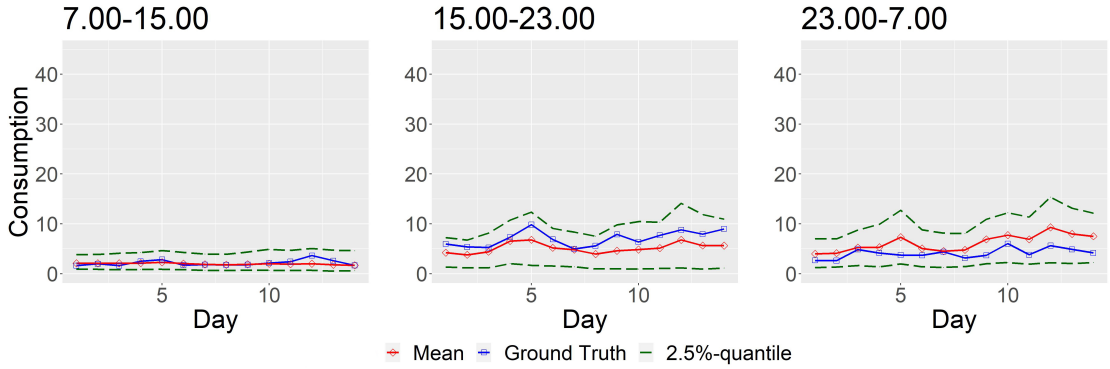
Model Parameter Estimation

In this particular problem, the aggregated readings form a time series recorded once per day and the goal is to impute a time series with three times the frequency, i.e., giving three readings per day. Therefore, we will need three separate AR models to impute the three time series. Recall that the equation for each AR model is given by

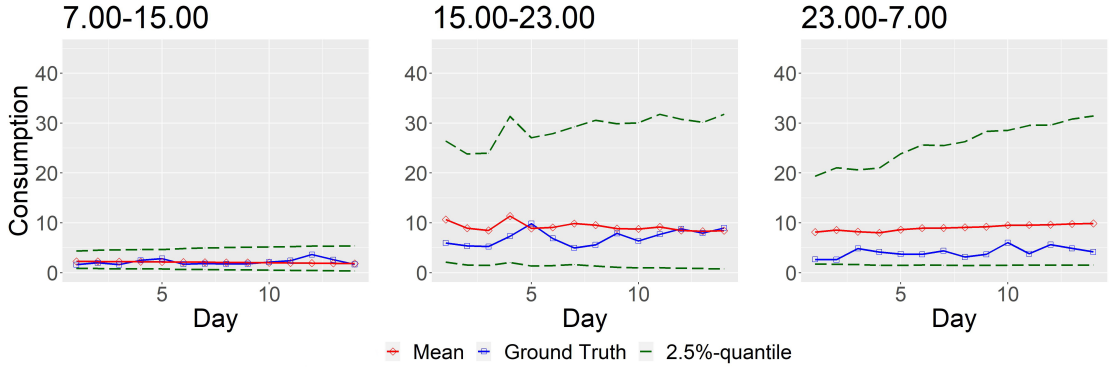
$$Y_{i,t}^{(j)} \sim \Gamma\left(\psi^{(j)}, \frac{\psi^{(j)}}{\mu_{i,t}^{(j)}}\right), \qquad \mu_{i,t}^{(j)} = \sum_{r=1}^{K} \Phi_r^{(j)} Y_{i,t-r}^{(j)} + \boldsymbol{\Xi}_i \boldsymbol{\beta}^{(j)} \tag{5.6}$$

where $j \in \{1, 2, 3\}$ is the index.

Terms in equation (5.6) have an additional subscript i to represent the estimation for the high-frequency readings for customer i. The parameters $\psi^{(j)}, \boldsymbol{\Phi}^{(j)}, \boldsymbol{\beta}^{(j)}$, however,



(a) Constrained simulation



(b) Unconstrained simulation

Fig. 5.2 Energy consumption imputation and error with and without constraints

are shared across all customers in this model. The model order K is chosen to be 7 as people tend to have their regular activities repeated in the unit of weeks.

A total of eight columns from questionnaire data, including the number of adults/children in the household, number of bedrooms, and number of large electrical appliances, like washing machines, tumble dryers, and dishwashers, are used as additional covariates $\boldsymbol{\Xi}_i$ in this model. We drop the t subscript as the survey data is time-independent. The high-frequency data from 30 households across 20 days are used to fit the model parameters. As mentioned before, $\boldsymbol{\Phi}$ and $\boldsymbol{\beta}$ are fitted through least-squares estimation and ψ from maximum likelihood estimation.

Result

After estimating the set of parameters $\left(\hat{\boldsymbol{\Phi}}^{(1:3)}, \hat{\boldsymbol{\beta}}^{(1:3)}, \hat{\psi}^{(1:3)}\right)$, we have three AR models of order 7 to estimate separately energy consumption in three time periods of the day. The imputation is conducted on a separate household to up-scale the daily readings of 14 days to a time series of length 52. By implementing Algorithm 25, the imputation is done in temporal order progressing in time t, with 100,000 samples of $Y_t^{(1:3)}$ drawn in each step to compute an estimate for the sample mean, sample variance and, the 95% confidence interval. Results are presented in Fig. 5.2. The imputation results

under constrained settings are shown in Fig. 5.2a and the unconstrained predictions are provided in Fig. 5.2b. In these figures, the sample mean, ground truth, and 95% CI are plotted in red (diamond), blue (square) and green (dashed) lines respectively.

Looking at Fig. 5.2a and 5.2b we directly see the significance of injecting the sum constraint information. The sample mean trajectory from the constrained model shows similar fluctuation as the real trajectory, unlike the unconstrained model where the estimated mean is mostly flat. We also see improvements in the estimated 95% CI in the constrained case, as the 95%-CI is always increasing with time for the unconstrained case. This is reasonable since without extra information the uncertainty could only increase as extra uncertainty is injected every time step. On the other hand, in Fig. 5.2a, the constrained 95%-CI does not seem to increase with time. Instead, it increases with the true value which is reasonable as the variance of Gamma distribution is proportional to its squared mean.

5.1.5 Study 2: Max-Min Prediction

In this section, we consider the energy consumption modeling problem from the Western Power Distribution challenge (*Western Power Distribution Data Challenge*, 2013). Spikes in energy demand could strain the network and one might mitigate the effect by monitoring the usage and reacting to the surge. However, monitoring the power usage with high-resolution reading can be expensive since this requires the installation of additional facilities and an ever-expanding data storage system. Thus, instead of monitoring with high-frequency in the long term, one might instead want to gather enough data to train a model to impute the high-frequency data and only maintain a low-frequency monitoring system. The goal is to predict the peaks and troughs of high-frequency time series for each half-hour using its average power consumption. The peaks and troughs are measured with respect to the discretized reading at the higher frequency.

Model Parameter Estimation

For monitoring peaks and troughs of energy usage, we consider the time series at a much higher frequency than once per day. The low-frequency observation stream will have a reading every 30 minutes and the high-frequency stream will have a reading every 6 minutes. Both time series record the average power usage within the time interval and the goal is to estimate the peak and trough values every 30 minutes in the 6-minute time series.

Using a similar notation as in (5.1), let $S_t^{(i)}$ denote the 30-minute readings and $Y_t^{(j)}$ denote the 6-minute readings where t still denotes the day number. The difference from the study in section 5.1.4 is we need more than 5 models to solve the problem since a 30-minute period is not a valid cycle for energy consumption data. Instead, we still use

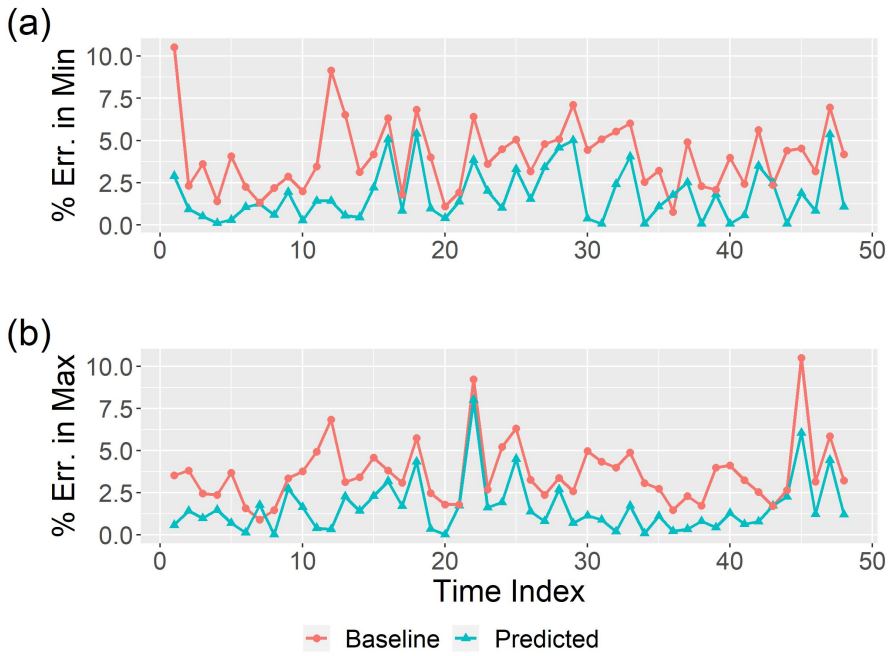


Fig. 5.3 Percentage difference prediction of peak and trough values, comparing the constrained model with the baseline.

one day as the cycle, and thus we consider each 6-minute period in a day separately which requires a total of $24 \times 60 \div 6 = 240$ separate AR models. The models can still be captured by (5.1):

$$Y_{i,t}^{(j)} \sim \Gamma\left(\psi^{(j)}, \frac{\psi^{(j)}}{\mu_{i,t}^{(j)}}\right), \qquad \mu_{i,t}^{(j)} = \sum_{r=1}^{K} \Phi_r^{(j)} Y_{i,t-r}^{(j)} + \boldsymbol{\Xi}_i \boldsymbol{\beta}^{(j)} \tag{5.1}$$

where j ranges from 1 to 240. By splitting the available data into 240 series and examining the auto-correlation plot for each time series, the order of the AR model is chosen to be 5 since more distant time points have a correlation close to zero. Each day has 48 half-hours and each half-hour induces a constraint

$$S_t^{(i)} = \sum_{j=1}^{5} Y_t^{(5(i-1)+j)}.$$

Here the number 5 is not the order of the AR model, but because each half-hour period consists of 5 6-minute periods and thus 5 random variables to be summed. Weather data near the power station is used as additional covariates $\boldsymbol{\Xi}$, including an hourly temperature and humidity reading.

Result

We used 3 months of high-frequency reading (from June to September 2021) to estimate the 240 sets of parameters $(\psi^{(i)}, \boldsymbol{\Phi}^{(i)}, \boldsymbol{\beta}^{(i)})$. Given the low-frequency readings for the

subsequent 7 days, we imputed the high-frequency readings and extracted the peaks and troughs for each 30-minute period. The peak and trough prediction performance is compared against the naive baseline which uses the 30-minute readings for both peak and trough estimates. The percentage error in peak and trough predictions is plotted in Figure 5.3, where red circles represent the baseline percentage errors and blue squares represent the percentage error for predicted values. We can see clearly that the predicted values are closer to the true values than the baseline. To be more specific, the RMSE of our predicted values is only 52% of the baseline RMSE.

5.2 Bias Mitigation in Job Advertisements

Another occasion, perhaps less common, is when one would like to balance the parameters of a model, for instance, in Lee-Carter's model, parameters α_i are required to sum to zero (Lee and Carter, 1992), or the quantity of items contributes towards an overall output which restricted to take certain value, e.g., zero. In this section, we will look at the latter situation, where we try to reword a piece of text to reduce bias by balancing the number of biased words.

To start with, we will quickly review some background literature related to gender bias. Firstly through the sociology point of view to look at how gender bias affects the workspace and secondly from the Natural Language Processing (NLP) community to see how gender bias is reflected in the trained models.

5.2.1 Background

5.2.1.1 Gender bias in job advertisement

Gender inequality in the labor market is longstanding and well-documented. Although there has been a long-term increase in women's labor force participation over the past few decades, research shows persistent gender segregation across many occupations and industries. Women continue to be underrepresented in senior and managerial positions (Sohrab et al., 2012), are less likely to be promoted and are perceived as less committed to professional careers (Wallace, 2008) and as less suitable to perform tasks in the fields that have been historically male-dominated (Hatmaker, 2013). The hiring process is a significant social encounter, in which employers search for the most 'suitable' candidate to fill the position (Rivera, 2020, Kang et al., 2016). Research demonstrates that 'suitability' is often defined categorically, is not neutral to bias, and is gendered (McCall, 2005). The wording of job advertisements, in particular, may play a role in generating such gender inequality. For instance, Bem and Bem (1973) and Kuhn et al. (2020) show that job advertisements with explicitly gendered words discourage potential applicants of the opposite gender from applying, even when they are qualified to

do so, which in turn reinforces the imbalance. More recent studies (Born and Taris, 2010, Askehave and Zethsen, 2014) have shown that words describing gendered traits and behaviors may also entail gendered responses from potential job applicants. Female students are substantially more attracted to advertisements that contain feminine traits than masculine traits (Born and Taris, 2010). Traits favored in leadership roles are predominately considered to be male-biased, correlating with the gender imbalance in top-management positions (Askehave and Zethsen, 2014). It has been shown that such bias co-exists with the salary gap where, on average, job posts that favor masculine traits offer higher salaries compared with job posts that favor feminine traits (Arceo-Gómez et al., 2020). Research also shows that using gender-neutral terms (e.g., police officer) or masculine/feminine pairs (e.g., policeman/policewoman) can help reduce gender barrier and attract both male and female applicants (Horvath and Sczesny, 2016, Sczesny et al., 2016, Bem and Bem, 1973).

5.2.1.2 Bias evaluation at the text level

Many studies can be found that collect and identify masculine and feminine words as a measure of gendered wording (Bem and Bem (1973); Bem (1974); Gaucher et al. (2011)). These word lists are consistent with previous research that examined gender differences in language use (Newman et al., 2008). Given the list of gender-coded words, text-level bias can be quantified by measuring the occurrences of each word in the list. Gaucher et al. (2011) calculated the percentage of masculine and feminine words in the text to produce two separate scores, for male and female biases respectively, to reveal the fact that job advertisements in male-dominated industries and female-dominated industries exhibit different score pairs. Tang et al. (2017) presents a slightly different approach where they assign weights to each gendered word by their level of gender implications that accumulate over the whole text, with the effects of masculine words and feminine words offsetting each other Tang et al. (2017).

Another technique of bias evaluation relies on the use of word embeddings. Using this technique, we can evaluate the level of bias owing to the fact that gender stereotype bias can be passed on from corpus to the embedding model through training (Bolukbasi et al., 2016). The Word Embedding Association Test (WEAT), proposed by Caliskan et al. (2017), is an analogue to the Implicit Association Test (IAT) used in Psychology studies. The purpose of WEAT is to test and quantify that two groups of target words, e.g., male-dominated professions vs. female-dominate professions, are indeed biased towards two groups of attribute words, e.g., $\{he\}$, $\{she\}$. A similar strategy is developed in Garg et al. (2018) called Relative Norm Distance (RND) which tests a single group of target words against two groups of attribute words, though the idea is much the same as WEAT. The bias of each word is evaluated by computing the difference in norm

distance between the word from a masculine word group and a feminine word group. This approach can be easily extended to the text level by averaging the bias score of each word in text (Kwak et al., 2021) or taking the average of word vectors prior to bias evaluation.

5.2.2 Modeling Gender Bias

Using gender-indefinite words alone does not remove gender signaling completely, since agentic attributes (e.g., *active* and *adventurous*), are usually considered to be masculine, and communal attributes (e.g., *considerate* and *sympathetic*), are often considered feminine. These attributes may be favored for certain job positions and it may not always be possible to find neutral alternatives to replace them. Thus it is more reasonable for the writer to keep these words while using words in the opposite gender to achieve inclusivity of both female and male applicants. Therefore, our methodology of mitigating bias in text involves the following steps:

1. Build an evaluation model of gender bias in words and texts;
2. Model probability distribution for the word occurrence of each group;
3. Provide guidance on how many words from each group should be used to mitigate bias.

5.2.2.1 Quantifying gender bias by words

To measure gender bias in job advertisements, we use a list of words that contain gendered psychological cues that may signal the employer's gender preferences for job candidates. Our word list builds on established inventories, i.e., Bem (1974) and Gaucher et al. (2011) inventories, which contain words that are well-established in the literature to signal implicit gender bias. Our word list also includes a further set of cues identified from job advertisements using expert coding that have not been included in the Bem and Gaucher inventories. For a full list of words used in our analysis and detailed information on the latter list, please see Konnikov et al. (2021). Moreover, we assume that every word in the masculine and feminine groups has a different level of signaling, so the words are sub-grouped further, in this case into two subgroups for computational simplicity, where each group of words is split into strongly or weakly masculine (or feminine) sets. In our setup, we used the GloVe Pennington et al. (2014) word embedding to achieve the split.

We assume that the overall bias expressed from a piece of text is equal to the sum of the bias expressed from each word, and more importantly, the effect of masculine words can be canceled out by the usage of feminine words in suitable proportions. Let Y_i denote the bias score of the i-th job text and $\boldsymbol{X}_i = (X_{i,\text{sm}}, X_{i,\text{wm}}, X_{i,\text{sf}}, X_{i,\text{wf}})$ denote

the number of occurrences of each word in the i-th job text aggregated according to the word groups, i.e., $X_{i,\text{sf}}$ denote the total number of *strongly feminine* words appearing in the i-th job text. Let $\beta_0, \boldsymbol{\beta}$ denote the model parameter, then

$$Y_i = \beta_0 + \boldsymbol{\beta}^\top \boldsymbol{X}_i.$$

5.2.2.2 Gender bias score at the text level

To collect the data for response Y_i in a comprehensive manner, we combine two different metrics to measure the bias at the text level. The first approach is based on the method proposed by Gaucher et al. (2011), which measures the bias purely through word counts and produces a score in $\{-1, 0, 1\}$ for feminine, neutral and masculine respectively. Since a discrete bias score is not adequate for capturing the degree of bias in texts, we adopted a word counting approach but modified the metric to give a continuous output in $[-1, 1]$. The score is computed as follows. The sign of the score is determined as in Gaucher et al. (2011) where a negative value represents feminine bias and a positive value represents masculine bias. The magnitude of the score is computed using the following equation:

$$|S_1| = \max\left\{\frac{X_{\text{mas}} - X_{\text{fem}}}{X_{\text{mas}}}, \frac{X_{\text{fem}} - X_{\text{mas}}}{X_{\text{fem}}}\right\}, \tag{5.7}$$

in which case when $X_{\text{mas}} = X_{\text{fem}}$ the measure will output 0.

However, this measure does not consider potential differences in the levels of bias exhibited by different words. Thus, we consider a second bias metric similar to the Relative Norm Distance (RND) (Garg et al., 2018) or the Word Embedding Association Test (WEAT) (Caliskan et al., 2017) with the help of a word embedding. Let V be a word embedding of choice, then V can be seen as a mapping of $V : \mathcal{W} \to \mathbb{R}^d$, where $\mathcal{W}$ is the vocabulary of the embedding, i.e., the set of all words it can recognize. Every word in its vocabulary $\mathcal{W}$ is assigned a vector of dimension d, which is 300 in GloVe. With some abuse of notation, we let $V_w = V(w)$ whenever $w \in \mathcal{W}$.

Since we need a text-level score, we average the word vectors from the same text to produce a text vector and compute its cosine distance to each of the masculine and feminine words in our word list. The difference in average cosine distance is our second score:

$$S_2 = \frac{1}{|\mathcal{M}|}\sum_{w\in\mathcal{M}} \frac{V_T \cdot V_w}{\|V_T\|\,\|V_w\|} - \frac{1}{|\mathcal{F}|}\sum_{w\in\mathcal{F}} \frac{V_T \cdot V_w}{\|V_T\|\,\|V_w\|}, \quad V_T = \frac{1}{|T|}\sum_{w\in T} V_w, \tag{5.8}$$

where T denotes the text with its cardinality $|T|$ defined as the number of words in T, V_w denote the word vector of word w, and $\mathcal{M}$, $\mathcal{F}$ denotes the set of masculine and feminine words, respectively. The scores S_1 and S_2 are combined through a linear combination

Algorithm 26: Text-bias evaluation

Input: List of masculine-coded words $\mathcal{M}$; List of feminine-coded words $\mathcal{F}$;
Word embedding V;
Text T to be evaluated;
Combination coefficient λ;

Count the number of masculine and feminine words in T and get X_m, X_f;
Compute score S_1=sign(X_m-X_f)$\max\left\{\frac{X_{\mathrm{m}}-X_{\mathrm{f}}}{X_{\mathrm{m}}}, \frac{X_{\mathrm{f}}-X_{\mathrm{m}}}{X_{\mathrm{f}}}\right\}$;
Compute text vector $V_T = \frac{1}{|T|}\sum_{w\in T} V_w$;
Initialize $S_m = 0$;
Initialize $S_f = 0$;
foreach *Masculine word* w *in Masculine list* **do**

$$S_m \mathrel{+}= \frac{V_T \cdot V_w}{\|V_T\|\,\|V_w\|}$$

end
foreach *Feminine word* w *in Feminine list* **do**

$$S_f \mathrel{+}= \frac{V_T \cdot V_w}{\|V_T\|\,\|V_w\|}$$

end
Compute $S_2 = \frac{1}{\mathcal{M}}S_m + \frac{1}{\mathcal{F}}S_f$;

Output: Combined score $S_\lambda = S_1 + \lambda S_2$

with coefficient λ to produce the final bias score for every text. In practice, λ is tuned through grid search, choosing the one with the largest R^2 value in the fitting stage (see the next section).

5.2.2.3 Bias Reduction as a Sampling Problem

The combined scores can be used to estimate the model parameters $\left(\hat{\beta}_0, \hat{\boldsymbol{\beta}}\right)$ through linear regression. With the model parameters $\left(\hat{\beta}_0, \hat{\boldsymbol{\beta}}\right)$ estimated, the goal is to minimize the overall bias by adjusting the frequency of different word types x_i. In theory, eliminating the use of gender-biased words may eliminate the bias completely. However, this is usually not possible since it can be hard to find neutral replacements for every word. Thus, we would like to seek a minimal adjustment to the word counts while reducing the bias. We would need to statistically model the word counts so that the debiased word count is highly correlated with the original word counts while satisfying some constraint (of zero bias) at the same time.

Although word counts are always integers, due to the complexity of solving probabilistic integer programming problems, we instead consider the continuous version with

a deterministic objective:

$$\hat{\beta}_0 + \hat{\boldsymbol{\beta}}^\top \boldsymbol{X}_i = 0. \tag{5.9}$$

where $\boldsymbol{X}_i$ is allowed to be a real vector which we can later round to an integer vector after debiasing.

With respect to the constraint above, the distribution of $\boldsymbol{X}_i$ should also be modeled in order for the adjusted word counts to be as close to the original as possible. However, a set of discrete random variables has almost no chance of meeting the zero bias objective in (5.9). In this case, we consider the Gamma distribution as a continuous substitute for Poisson distribution since it is also non-negative. We assume that each job text is an instance of its own text distribution and thus every word count is from the same distribution but with distinct parameters, even for word counts of the same group. Therefore, rather than finding a common posterior distribution for the word count for each group, we would like to parameterize each distribution separately. To avoid over-complication, we leave 1 degree of freedom for each word count distribution to adjust its mean while using a common rate parameter for each group. Let $\boldsymbol{X}_i = (X_{i,\text{sm}}, X_{i,\text{wm}}, X_{i,\text{sf}}, X_{i,\text{wf}})$ and for each word group $g \in \mathcal{G} := \{\text{sm}, \text{wm}, \text{sf}, \text{wf}\}$, $X_{i,g} \sim \Gamma(\alpha_{i,g}, \psi_g)$ with the density function given by

$$f_{i,g}(x) = \frac{\psi_g^{\alpha_{i,g}}}{\Gamma(\alpha_{i,g})} x^{\alpha_{i,g}-1} \exp(-\psi_g x), \quad \alpha_{i,g} := \tilde{X}_{i,g}\psi_g, \tag{5.10}$$

where ψ_g is the fitted rate parameter using the collected word counts for each word group g separately and the mean of the distribution is chosen as the unadjusted word count $\tilde{X}_{i,g}$ for group g in text i. Now we have the following constrained distribution for job post i:

$$f_i(\boldsymbol{X}_i) = \prod_{g\in\mathcal{G}} f_{i,g}(X_{i,g}; \alpha_{i,g}, \psi_g) \quad \text{w.r.t.} \quad \hat{\boldsymbol{\beta}}^\top \boldsymbol{X}_i = -\hat{\beta}_0. \tag{5.11}$$

Finally, we can sample the unknown debiased word counts by simulating from the above distribution to give a natural choice of wording that also reduces the bias.

5.2.2.4 Constrained density fusion

Let $d = |\mathcal{G}|$ denote the number of different word types. Recall that our target is to sample from the constrained product density function

$$f(\boldsymbol{X}) \propto \prod_{g\in\mathcal{G}} f(X_g; \alpha_g) \quad \text{w.r.t.} \quad \hat{\boldsymbol{\beta}}^\top \boldsymbol{X} = -\hat{\beta}_0, \tag{5.12}$$

where $\boldsymbol{X} = (X_{\text{sm}}, X_{\text{wm}}, X_{\text{sf}}, X_{\text{wf}})$. Hence, Algorithm 23 or 25 can be applied to sample from the target distribution. Consider the following proposal distribution $h(\boldsymbol{X}, \boldsymbol{Y})$:

$$h(\boldsymbol{X}, \boldsymbol{Y}) \propto \prod_{j=1}^{d} f(X_j; \alpha_j) \times \eta_{\hat{\boldsymbol{\beta}}}(\boldsymbol{X}) \times \frac{\mathcal{N}_d\left(\boldsymbol{Y}; \boldsymbol{X}, T\boldsymbol{I}_d\right) \mathbb{1}_{\left\{\hat{\boldsymbol{\beta}}^\top \boldsymbol{Y} = -\hat{\beta}_0\right\}}}{\eta_{\hat{\boldsymbol{\beta}}}(\boldsymbol{X})} \times Q, \quad (5.13)$$

where

$$Q = \mathbb{E}_{\mathbb{W}_{(T,\boldsymbol{X},\boldsymbol{Y})}}\left[\Phi(\boldsymbol{\omega})\right], \quad \Phi(\boldsymbol{\omega}) = \exp\left[-\sum_{j=1}^{d} \int_0^T \phi_i(\boldsymbol{\omega}_s^{(i)}) \mathrm{d}s\right]. \quad (5.14)$$

Using $'$ to denote the derivative symbol, the definition of ϕ_i is given by

$$\phi_i(x) = \frac{1}{2}\left[A_i'(x)^2 + A_i''(x)\right] - l_i, \quad A_i(x) := \log f_i(x), \quad (5.15)$$

with $l_i > -\infty$ being a lower bound of ϕ_i. Finally, the normalizing constant η is

$$\eta_{\hat{\boldsymbol{\beta}}}(\boldsymbol{X}) = \exp\left[-\frac{1}{2TB}\left(\hat{\beta}_0 + \hat{\boldsymbol{\beta}}^\top \boldsymbol{X}\right)^2\right], \qquad B = \left|\left|\hat{\boldsymbol{\beta}}\right|\right|^2.$$

Usually, it would be preferable to implement the particle filter approach as in Algorithm 25 for better efficiency. Since we are not simulating time series in this case, one does not need to maintain and update the weight of the particles, where the algorithm essentially becomes an importance sampling. By estimating the rejection probability, the rejection sampling can be turned into an importance sampling approach as presented in Algorithm 27. The shape parameters ψ_g in the algorithm are assumed to be known. In practice, we can estimate a shape parameter for each word group by fitting a Gamma distribution to the existing data. After simulating enough weighted samples, one can use the estimated mean as the debiased result. The rounded figure suggests how many words of each group should be included in the paraphrased text.

5.2.3 Application Result

In this section, we test the evaluation and debiasing strategy and algorithms on a real job post dataset that consists of 100,000 data points collected from one of the largest job advertising platforms in the UK. The raw dataset contains job post information including *job title*, *job sector*, *job description*, *job location*, *full time or part time job,* and *salary*. Although job titles can be biased towards a certain gender, such gendered words have always appeared as part of a pair in the job titles in our dataset, e.g., postman/postwoman. Since the other fields are not the primary interest of this paper, we focused only on the *job description* data containing the main advertisement text.

The job texts are parsed from HTML to plain text and further processed to remove

Algorithm 27: Bias reduction on word counts

Input: Word Counts $\tilde{X}_{\text{sm}}, \tilde{X}_{\text{wm}}, \tilde{X}_{\text{sf}}, \tilde{X}_{\text{wf}}$;
Bias weights $\hat{\boldsymbol{\beta}} = (\beta_{\text{sm}}, \beta_{\text{wm}}, \beta_{\text{sf}}, \beta_{\text{wf}})$;
Intercept $\hat{\beta}_0$;
Gamma rate parameter ψ_g for each word group, estimated from the dataset;
Number of samples N; Tuning parameter T;

foreach *word group* g *in* $\mathcal{G}$ **do**
  Compute gamma shape parameter $\alpha_g = \tilde{X}_g \psi_g$;
end
for *i = 1,..., N* **do**
  foreach *word group* g *in* $\mathcal{G}$ **do**
    Sample $X_{i,g} \sim \Gamma(\alpha_g, \psi_g)$;
  end
  Simulate $\boldsymbol{Y}_i \sim \mathcal{N}(\boldsymbol{X}_i, TI_d)\mathbb{1}_{\{\hat{\boldsymbol{\beta}}^\top \boldsymbol{Y}_i = -\hat{\beta}_0\}}$;
  Compute normalizing constant $\eta_{\hat{\boldsymbol{\beta}}}(\boldsymbol{X}_i)$;
  Compute the Poisson estimate $\hat{\Phi}_i$ of Q_i;
  Importance weight $w_i = \eta_{\hat{\boldsymbol{\beta}}}(\boldsymbol{X}_i) * \hat{\Phi}_i$;
end
$\bar{\boldsymbol{Y}} = \sum_{i=1}^{N} w_i \boldsymbol{Y}_i$;
Output: Empirical mean $\bar{\boldsymbol{Y}}$ *rounded* to the nearest integer;

symbols. Then, the word counts are conducted by counting the total number of words in an advertisement and counting the occurrences of every word in our word list (see Konnikov et al. (2021) for a full list of words). Some entries in the word list are root words, e.g., *aggress**, in which case any variant that matches this root, e.g., *aggressive* and *aggression*, shares the same counter. Sometimes *regex* can match words that are misspelled, which should not be counted. In this case, we filter out these words by checking if they are contained in a dictionary. We used WordNet in our implementation.

In the end, the word counts are aggregated according to their word groups, {*strongly*, *weakly*} × {*masculine*, *feminine*}. The split is achieved using the GloVe word embedding (Pennington et al., 2014) by ranking the cosine similarity between each word and the gender direction $he - she$.

5.2.3.1 Bias score

The text-level bias score is evaluated by combining two distinct measures based on word counts (Gaucher et al., 2011) and word embeddings (Garg et al., 2018), respectively, as described in Algorithm 26. Let S_λ denote the combined score using coefficient λ, in this case, $\lambda = 2$ which gives the best R^2 score in the regression stage compared with other candidates in the grid search. We formulate and solve the linear regression problem

$$S_{i,\lambda} = \beta_0 + \beta_{\text{sm}}\tilde{X}_{i,\text{sm}} + \beta_{\text{wm}}\tilde{X}_{i,\text{wm}} + \beta_{\text{sf}}\tilde{X}_{i,\text{sf}} + \beta_{\text{wf}}\tilde{X}_{i,\text{wf}} + \epsilon_i,$$

	Estimate	Std. Error	t value
Intercept	-0.1439 ***	0.0035	-40.78
Strong masculine	0.1580 ***	0.0008	199.42
Weak masculine	0.0073 ***	0.0004	16.39
Strong feminine	-0.1824 ***	0.0016	-115.45
Weak feminine	-0.1440 ***	0.0008	-175.35
R^2			0.465

*** $p < 0.001$

Table 5.1 Estimated weight for each word group.

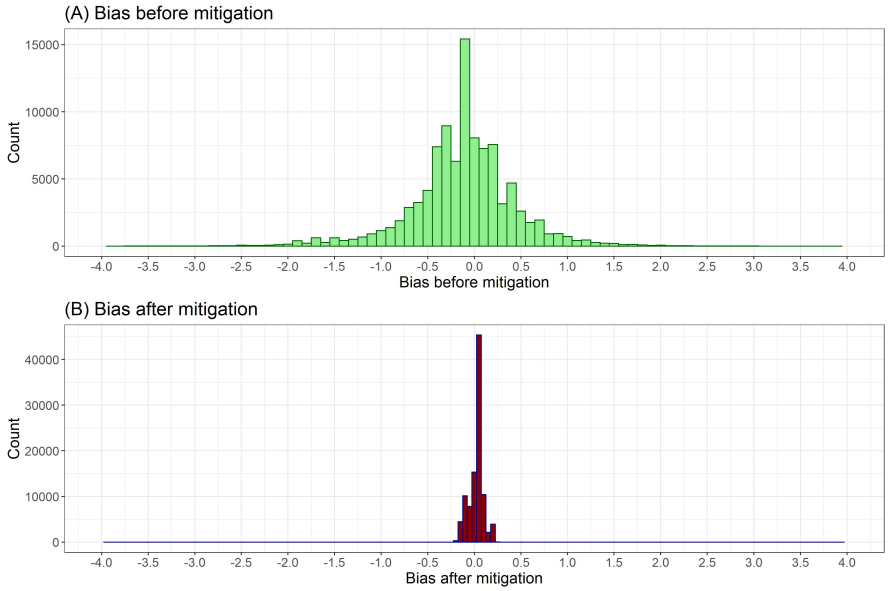


Fig. 5.4 Histogram of bias score distribution (A) before and (B) after debiasing algorithm is applied. Both scores are measured using the fitted metric in Section 4.1.

where ϵ_i is i.i.d. Gaussian noise and $\tilde{X}_{i,g}$ is the word count for word group g in the i-th text. The fitted parameters are shown in Table 5.1. We can see from the R^2 that the regression model fits the estimated bias score reasonably well given the relatively simple and crude split of word groups. Let S_β denote the bias score estimated using the model parameters. Our fitted bias evaluation S_β is consistent with the combined bias score S_λ with a high Pearson's correlation, $\text{cor}(S_\lambda, S_\beta) = \mathbf{0.68}$.

The direction of bias in the bias score is recovered with *positive* towards *masculine* and **negative** towards **feminine**. In addition, the regression parameter validates the strong/weak split as the strong groups have coefficients with a larger magnitude than the weak groups. Overall, we can see that masculine words are assigned smaller weights, which can be caused by the wider usage of masculine words in the job text, similarly for the intercept which is negative.

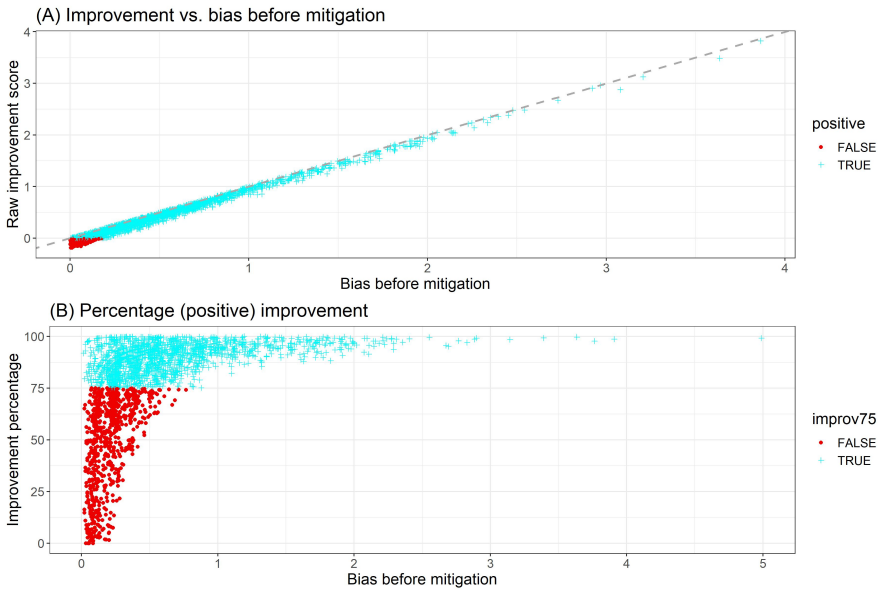


Fig. 5.5 (A) Raw improvement and (B) percentage improvement plotted against the unsigned bias score before debiasing. In the percentage plot, only positive improvements are plotted since the points with negative improvement were already close to no bias and thus not relevant to the context.

Statistics	Among those with			
	all data	improv. > 0	bias > 0.23	bias > 0.75
mean |before|	0.4149	0.4536	0.6269	1.2362
mean |after|	0.0628	0.0588	0.0647	0.0677
mean improv.	0.3521	0.3948	0.5623	1.1685
mean % improv.	32.77%	75.92%	86.08%	93.89%

Table 5.2 Mean unsigned bias before and after debiasing with mean improvement and percentage improvement for different groups of data.

5.2.3.2 Debiasing Result

With the bias weights $\hat{\boldsymbol{\beta}}$ and intercept $\hat{\beta}_0$ estimated, we progress to sample the debiased word counts to reduce overall bias while keeping the relevant word counts close to the original version. For each word group, we fit a Gamma distribution to the 100,000 data points to get the corresponding rate parameter, $(\psi_{\text{sm}}, \psi_{\text{wm}}, \psi_{\text{sf}}, \psi_{\text{wf}}) = (0.362, 0.258, 0.353, 0.350)$. Then we assume that the word count of group g in the i-th text $X_{i,g}$, $g \in \mathcal{G}$ is a random variable that follows a Gamma distribution, $X_{i,g} \sim \Gamma(\tilde{X}_{i,g}\psi_g, \psi_g)$. Let $f(X_{i,g})$ given by (5.10) denote its density function. To debias each job text, we consider sampling from the target distribution (5.12)

The simulation is done by following Algorithm 27, and Figure 5.4 shows a comparison of bias score distribution before and after applying our bias mitigation approach.

Before debiasing, the majority of job advertisements have bias scores between -2.0 and 2.0. After the bias mitigation, the bias score distribution is reduced to between -0.25 and 0.25 as shown in Figure 26 (b), with a high concentration around 0.

The individual improvements are plotted in Figure 5.5a and 5.5b. The bias improvement is computed by taking the difference between the unsigned (absolute value) bias score before debiasing and the unsigned bias score after debiasing. To avoid overcrowding the scatter plot, both Figure 5.5a and 5.5b contain 3000 randomly sampled data points from the output. In Figure 5.5a, the bias improvement is strongly linear with the unsigned bias before debiasing and the linear relation has a slope close to 1. More importantly, the majority of points (**over 90%**) have positive improvements while the points with negative improvements have a very small unsigned bias score (< 0.23) in the first place. In practice, the debiasing process of these points can be omitted since their original level of gender bias is close to 0.

Therefore, we only use the points with positive improvements in Figure 5.5b, where the percentage improvement is plotted against the unsigned bias score before debiasing. Overall, 67.7% of the points have percentage improvements greater than 75%, and the percentage increases to 99.9% for those with unsigned bias score greater than 0.75. From Table 5.2 we can see that the mean improvement gets better when we filter out texts with a lower magnitude of bias. For texts with a bias score of > 0.75, the mean improvement percentage is 93.89% while the mean bias score after debiasing is 0.0677, which is very close to the mean debiased score across all data points 0.0628.

CHAPTER 6

Discussion

Perfect sampling is an important area in computational statistics that demands further exploration and research. The promise of having no convergence issue is an attractive advantage for favoring exact methods over non-exact (MCMC) algorithms. On the list of perfect sampling algorithms we introduced in this thesis, we started with the most general framework which is "rejection sampling" where one samples from the proposal distribution and corrects the outcomes by a rejection step using a probability proportional to the density quotient. We showed that the rejection sampling framework can also be extended to simulate diffusion bridges which are fundamentally infinite dimensional random variables. Based on the simulation of diffusion bridges, we introduced the Monte Carlo Fusion algorithm that samples from product densities. Although MCF is again an algorithm for sampling from finite dimensions, its rejection does not require the estimation of the bounding constant M in the usual rejection sampling (Algorithm 2). This boosts the versatility of rejection sampling and allows it to be implemented easily on product densities.

Although the mechanism of MCF is based on simulating diffusion processes, only the endpoint is returned as the accepted sample, which means the time length T can be chosen freely. From Chapter 3, we discussed the problem that when T is inappropriate, the efficiency of the algorithm drops significantly. For that purpose, we introduced and investigated the setup where the tuning parameter T can be chosen separately for different fusion components. In short, choosing T_i separately does boost performance provided that the values are appropriately tuned. However, the tuning strategy may be further improved by better estimating the second acceptance probability.

In chapter 4, we looked at another extension of the MCF algorithm which allows it to simulate the product density with respect to a linear constraint. We have analyzed theoretically, in the Gaussian case, that adding a sum constraint to an unconstrained model could in many cases improve the MSE. One must be aware that adding the constraint does not make the mean estimation closer to the true value in every dimension. The main benefit of applying the sum constraint is to reduce uncertainty in the model and such reduction in uncertainty outweighs the marginal increase in bias in cases when the mean estimation is relatively accurate but the uncertainty is high. Through simu-

lation, we have also shown the effect of applying such sum constraint to some other non-Gaussian models and the theoretical result carried over quite well.

The extension is demonstrated to be effective when dealing with imputation problems in time series models, where the problem of uncertainty propagation becomes rather significant. In the first application setting, we mainly looked at the problem of oversampling, i.e., magnifying the frequency of a time series by imputing the higher frequency readings with respect to a sum constraint. This is applicable to all time series that record accumulated values like consumption or flow, e.t.c. The same approach can also be applied to imputing missing values in time series provided that the aggregated value exists. Another application we examined is to reduce the aggregated bias in a piece of text, which happens to have a sum constraint. Although the bias evaluation model is rather simple, we showed that the potential applications of this algorithm are not limited to time series models.

As for future work, we can see several potential improvements to the works summarized in this thesis:

- The layered partition of the path space in the path-space rejection sampling algorithm (Algorithm 19) is defined across the full time length. However, when the starting and ending points $\boldsymbol{X}_0$ and $\boldsymbol{X}_t$ are distant from each other, the width of the layer interval will be large. As a result, the upper bound for ϕ could be large even for the first layer. If it is possible to apply different layer intervals for different time sections of the path, e.g., by incrementally deciding the position of skeleton points, one should be able to improve the acceptance rate by having tighter layer bounds.

- The tuning algorithm for the time parameter in Chapter 3 could be improved by better estimating the expectation of acceptance rate AP_2. One may also consider a hierarchical model for tuning the set of time parameters t_i.

- In all the diffusion simulations we presented in this thesis, the proposal diffusion is always a Brownian motion. Though the choice is not limited, one may consider other proposal diffusion processes like the Ornstein-Uhlenbeck process (Dai et al., 2019) or other guided proposal processes Schauer et al. (2017).

- Chapter 4 and 5 worked with the linearly constrained MCF algorithm which we were able to derived smoothly from the base MCF setup due to the use of the Gaussian proposal. However, in non-linear constraints, there is no obvious way to sample the two ends of the diffusion process. Extending MCF to address non-linear constraints would be an interesting and challenging work to take on, where one shall find an efficient way to generate proposals that land exactly on the constraint.

- In the energy consumption disaggregation problems, we used several independent time series to model the different fusion components, e.g., three independent time series are used for disaggregating the daily consumption into tridaily readings. This imposes a rather strong assumption that could undermine the model performance. It would be worth considering using multi-dimensional distributions to model the fusion components so that the correlations between them can be retained.

References

Amann, H. and Escher, J. (2009), *Analysis III*, Springer.

Arceo-Gómez, E. O., Campos-Vázquez, R. M., Salas, R. Y. B. and López-Araiza, S. (2020), Gender stereotypes in job advertisements: What do they imply for the gender salary gap?, *in* 'Mexico. Retrieved from http://conference. iza. org/conference files'.

Askehave, I. and Zethsen, K. K. (2014), 'Gendered constructions of leadership in Danish job advertisements', *Gender, Work and Organization* **21**(6), 531–545.

Asmussen, S., Glynn, P. and Pitman, J. (1995), 'Discretization error in simulation of one-dimensional reflecting brownian motion', *The Annals of Applied Probability* **5**(4), 875–896.

Asselin, P., Evans, R. F. L., Barker, J., Chantrell, R. W., Yanes, R., Chubykalo-Fesenko, O., Hinzke, D. and Nowak, U. (2010), 'Constrained monte carlo method and calculation of the temperature dependence of magnetic anisotropy', *Physical Review B* **82**(5), 054415.

Bem, S. L. (1974), 'The measurement of psychological androgyny.', *Journal of Consulting and Clinical Psychology* **42**(2), 155.

Bem, S. L. and Bem, D. J. (1973), 'Does sex-biased job advertising "aid and abet" sex discrimination?', *Journal of Applied Social Psychology* **3**(1), 6–18.

Bertoin, J. and Pitman, J. (1994), 'Path transformations connecting brownian bridge, excursion and meander', *Bulletin des sciences mathématiques* **118**(2), 147–166.

Beskos, A., Papaspiliopoulos, O. and Roberts, G. O. (2008), 'A factorisation of diffusion measure and finite sample path constructions', *Methodology and Computing in Applied Probability* **10**(1), 85–104.

Beskos, A., Papaspiliopoulos, O., Roberts, G. O. and Fearnhead, P. (2006), 'Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes (with discussion)', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68**(3), 333–382.

Beskos, A., Roberts, G. O. et al. (2005), 'Exact simulation of diffusions', *The Annals of Applied Probability* **15**(4), 2422–2444.

Binder, K. (1997), 'Applications of monte carlo methods to statistical physics', *Reports on Progress in Physics* **60**(5), 487.

Bird, G. (1981), 'Monte-carlo simulation in an engineering context', *Progress in Astronautics and Aeronautics* **74**, 239–255.

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V. and Kalai, A. T. (2016), 'Man is to computer programmer as woman is to homemaker? Debiasing word embeddings', *Advances in Neural Information Processing Systems* **29**, 4349–4357.

Born, M. P. and Taris, T. W. (2010), 'The impact of the wording of employment advertisements on students' inclination to apply for a job', *The Journal of Social Psychology* **150**(5), 485–502.

Brubaker, M., Salzmann, M. and Urtasun, R. (2012), A family of mcmc methods on implicitly defined manifolds, *in* 'Artificial Intelligence and Statistics', pp. 161–172.

Burger, P., Bezençon, V., Bornemann, B., Brosch, T., Carabias-Hütter, V., Farsi, M., Hille, S. L., Moser, C., Ramseier, C., Samuel, R., Sander, D., Schmidt, S., Sohre, A. and Volland, B. (2015), 'Advances in understanding energy consumption behavior and the governance of its change – outline of an integrated framework', *Frontiers in Energy Research, https://doi.org/10.3389/fenrg.2015.00029* .

Byrne, S. and Girolami, M. (2013), 'Geodesic monte carlo on embedded manifolds', *Scandinavian Journal of Statistics* **40**(4), 825–845.

Caliskan, A., Bryson, J. J. and Narayanan, A. (2017), 'Semantics derived automatically from language corpora contain human-like biases', *Science* **356**(6334), 183–186.

Chow, W. C. (2009), 'Brownian bridge', *Wiley interdisciplinary reviews: computational statistics* **1**(3), 325–332.

Chua, A. J. (2020), 'Sampling from manifold-restricted distributions using tangent bundle projections', *Statistics and Computing* **30**(3), 587–602.

Cong, Y., Chen, B., Zhou, M. et al. (2017), 'Fast simulation of hyperplane-truncated multivariate normal distributions', *Bayesian Analysis* **12**(4), 1017–1037.

Dai, H. (2019), 'A review on the exact monte carlo simulation', *Bayesian Inference on Complicated Data* .

Dai, H., Pollock, M. and Roberts, G. (2019), 'Monte carlo fusion', *Journal of Applied Probability* **56**(1), 174–191.

Davis, P. J. and Rabinowitz, P. (2007), *Methods of numerical integration*, Courier Corporation.

De Bernardis, C., Vicente-Guijalba, F., Martinez-Marin, T. and Lopez-Sanchez, J. M. (2016), 'Particle filter approach for real-time estimation of crop phenological states using time series of ndvi images', *Remote Sensing* **8**(7), 610.

Djuric, P. M., Huang, Y. and Ghirmai, T. (2002), 'Perfect sampling: a review and applications to signal processing', *IEEE Transactions on Signal processing* **50**(2), 345–356.

Douc, R. and Cappé, O. (2005), Comparison of resampling schemes for particle filtering, *in* 'Ispa 2005. proceedings of the 4th international symposium on image and signal processing and analysis, 2005.', IEEE, pp. 64–69.

Doucet, A., Johansen, A. M. et al. (2009), 'A tutorial on particle filtering and smoothing: Fifteen years later', *Handbook of nonlinear filtering* **12**(656-704), 3.

Durrett, R. (2019), *Probability: theory and examples*, Vol. 49, Cambridge university press.

Fearnhead, P. (2005), 'Using random quasi-monte-carlo within particle filters, with application to financial time series', *Journal of Computational and Graphical Statistics* **14**(4), 751–769.

Fearnhead, P., Papaspiliopoulos, O. and Roberts, G. O. (2008), 'Particle filters for partially observed diffusions', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **70**(4), 755–777.

Fishman, G. (2013), *Monte Carlo: concepts, algorithms, and applications*, Springer Science and Business Media.

Fleiss, J. (1993), 'Review papers: The statistical basis of meta-analysis', *Statistical methods in medical research* **2**(2), 121–145.

Flury, B. D. (1990), 'Acceptance–rejection sampling made easy', *Siam Review* **32**(3), 474–476.

Friedman, A. (1975), *Stochastic Differential Equations and Applications: Volume 1*, Elsevier Science.

Garg, N., Schiebinger, L., Jurafsky, D. and Zou, J. (2018), 'Word embeddings quantify 100 years of gender and ethnic stereotypes', *Proceedings of the National Academy of Sciences* **115**(16), E3635–E3644.

Gaucher, D., Friesen, J. and Kay, A. C. (2011), 'Evidence that gendered wording in job advertisements exists and sustains gender inequality.', *Journal of Personality and Social Psychology* **101**(1), 109.

Girsanov, I. V. (1960), 'On transforming a certain class of stochastic processes by absolutely continuous substitution of measures', *Theory of Probability and Its Applications* **5**(3), 285–301.

Glasserman, P. (2004), *Monte Carlo methods in financial engineering*, Vol. 53, Springer.

Haggstrom, O. and Nelander, K. (1999), 'On exact simulation of markov random fields using coupling from the past', *Scandinavian Journal of Statistics* **26**(3), 395–411.

Hansen, N. R. (2003), 'Geometric ergodicity of discrete-time approximations to multivariate diffusions', *Bernoulli* **9**(4), 725–743.

Hatmaker, D. M. (2013), 'Engineering identity: Gender and professional identity negotiation among women engineers', *Gender, Work and Organization* **20**(4), 382–396.

Horvath, L. K. and Sczesny, S. (2016), 'Reducing women's lack of fit with leadership positions? Effects of the wording of job advertisements', *European Journal of Work and Organizational Psychology* **25**(2), 316–328.

Huang, Y., Wang, X., Guo, Y. and An, W. (2017), State estimation with incomplete linear constraint, *in* '2017 20th International Conference on Information Fusion (Fusion)', IEEE, pp. 1–6.

Huber, M. (1998), Exact sampling and approximate counting techniques, *in* 'Proceedings of the thirtieth annual ACM symposium on Theory of computing', pp. 31–40.

Itô, K. (1951), 'On a formula concerning stochastic differentials', *Nagoya Mathematical Journal* **3**, 55–65.

Kang, S. K., DeCelles, K. A., Tilcsik, A. and Jun, S. (2016), 'Whitened résumés: Race and self-presentation in the labor market', *Administrative Science Quarterly* **61**(3), 469–502.

Karatzas, I. and Shreve, S. (2012), *Brownian motion and stochastic calculus*, Vol. 113, Springer Science and Business Media.

Kibria, B. G. and Joarder, A. H. (2006), 'A short review of multivariate t-distribution', *Journal of Statistical research* **40**(1), 59–72.

Kingman, J. F. C. (1992), *Poisson processes*, Vol. 3, Clarendon Press.

Konnikov, A., Denier, N., Hu, Y., Hughes, K. D., Ding, L., Al-Ani, J. A., Rets, I. and Tarafdar, M. (2021), 'Word inventory for work and employment diversity, (in)equality and inclusivity'. pre-print on SocArXiv.

Kuhn, P., Shen, K. and Zhang, S. (2020), 'Gender-targeted job ads in the recruitment process: Facts from a Chinese job board', *Journal of Development Economics* **147**, 102531.

Kwak, H., An, J., Jing, E. and Ahn, Y.-Y. (2021), 'Frameaxis: Characterizing microframe bias and intensity with word embedding', *PeerJ Computer Science* **7**, e644.

Lee, R. D. and Carter, L. R. (1992), 'Modeling and forecasting us mortality', *Journal of the American Statistical Association* **87**(419), 659–671.

Maggs, A. (2005), 'Monte carlo simulation of a model of water', *Physical Review E* **72**(4), 040201.

Manning, W. G., Basu, A. and Mullahy, J. (2005), 'Generalized modeling approaches to risk adjustment of skewed outcomes data', *Journal of Health Economics* **24**(3), 465–488.

Mazhdrakov, M., Benov, D. and Valkanov, N. (2018), *The Monte Carlo method: engineering applications*, ACMO Academic Press.

McCall, L. (2005), 'The complexity of intersectionality', *Signs: Journal of Women in Culture and Society* **30**(3), 1771–1800.

Meng, F., Zeng, X.-J., Zhang, Y., Dent, C. J. and Gong, D. (2018), 'An integrated optimization+ learning approach to optimal dynamic pricing for the retailer with multi-type customers in smart grids', *Information Sciences* **448**, 215–232.

Meyn, S. P. and Tweedie, R. L. (2012), *Markov chains and stochastic stability*, Springer Science and Business Media.

Mitha, F. and Huber, M. L. (2012), 'Monotonic multigamma coupling for perfect sampling', *Journal of Statistical Computation and Simulation* **82**(4), 603–622.

Murdoch, D. J. and Green, P. J. (1998), 'Exact sampling from a continuous state space', *Scandinavian Journal of Statistics* **25**(3), 483–502.

Murray, L. M. (2013), 'Bayesian state-space modelling on high-performance hardware using libbi', *arXiv preprint arXiv:1306.3277* .

Newman, M. L., Groom, C. J., Handelman, L. D. and Pennebaker, J. W. (2008), 'Gender differences in language use: An analysis of 14,000 text samples', *Discourse Processes* **45**(3), 211–236.

Pennington, J., Socher, R. and Manning, C. D. (2014), Glove: Global vectors for word representation, *in* 'Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)', pp. 1532–1543.

Peppanen, J., Zhang, X., Grijalva, S. and Matthew, R. (2016), 'Handling bad or missing smart meter data through advanced data imputation', *Conference: IEEE PES Innovative Smart Grid Technologies (ISGT), September, DOI: 10.1109/ISGT.2016.7781213* .

Pitman, J. W. (1975), 'One-dimensional brownian motion and the three-dimensional bessel process', *Advances in Applied Probability* **7**(3), 511–526.

Pötzelberger, K. and Wang, L. (2001), 'Boundary crossing probability for brownian motion', *Journal of applied probability* **38**(1), 152–164.

Poursharif, G., Brint, A., Black, M. and Mark, M. (2017), 'Analysing the ability of smart meter data toprovide accurate information to the ukdnos', *24th International Conference and Exhibition on Electricity Distribution (CIRED), September* .

Propp, J. G. and Wilson, D. B. (1996), 'Exact sampling with coupled markov chains and applications to statistical mechanics', *Random Structures and Algorithms* **9**(1-2), 223–252.

Rafsanjani, H. N., Moayedi, S., Ahn, C. R. and Alahmad, M. (2020), 'A load-disaggregation framework to sense personalized energy-use information in commercial buildings', *Energy and Buildings* **207**, 109633.

Rivera, L. A. (2020), 'Employer decision making', *Annual Review of Sociology* **46**, 215–232.

Robert, C. P., Casella, G. and Casella, G. (1999), *Monte Carlo statistical methods*, Vol. 2, Springer.

Roberts, G. O. and Rosenthal, J. S. (2004), 'General state space markov chains and mcmc algorithms', *Probability surveys* **1**, 20–71.

Roberts, G. O. and Tweedie, R. L. (1996), 'Exponential convergence of langevin distributions and their discrete approximations', *Bernoulli* pp. 341–363.

Schauer, M., Van Der Meulen, F. and Van Zanten, H. (2017), 'Guided proposals for simulating multi-dimensional diffusion bridges', *Bernoulli* **23**(4A), 2917–2950.

Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I. and McCulloch, R. E. (2016), 'Bayes and big data: The consensus monte carlo algorithm', *International Journal of Management Science and Engineering Management* **11**(2), 78–88.

Sczesny, S., Formanowicz, M. and Moser, F. (2016), 'Can gender-fair language reduce gender stereotyping and discrimination?', *Frontiers in Psychology* **7**, 25.

Shepp, L. A. (1979), 'The joint density of the maximum and its location for a wiener process with drift', *Journal of Applied probability* **16**(2), 423–427.

Smith, T. C., Spiegelhalter, D. J. and Thomas, A. (1995), 'Bayesian approaches to random-effects meta-analysis: a comparative study', *Statistics in medicine* **14**(24), 2685–2699.

Sohrab, G., Karambayya, R. and Burke, R. J. (2012), 'Women in management in Canada', *Women in Management Worldwide: Progress and Prospects* pp. 165–181.

Tang, S., Zhang, X., Cryan, J., Metzger, M. J., Zheng, H. and Zhao, B. Y. (2017), 'Gender bias in the job market: A longitudinal analysis', *Proceedings of the ACM on Human-Computer Interaction* **1**(CSCW), 1–19.

Valenzuela, J. and Mazumdar, M. (2001), 'Monte carlo computation of power generation production costs under operating constraints', *IEEE Transactions on Power Systems* **16**(4), 671–677.

Vrins, F. (2018), 'Sampling the multivariate standard normal distribution under a weighted sum constraint', *Risks* **6**(3), 64.

Wallace, J. E. (2008), 'Parenthood and commitment to the legal profession: Are mothers less committed than fathers?', *Journal of Family and Economic Issues* **29**(3), 478–495.

Wang, S., Li, R., Evans, A. and Li, F. (2020), 'Regional nonintrusive load monitoring for low voltage substations and distributed energy resources', *Applied Energy* **260**, 114225.

Western Power Distribution Data Challenge (2013), `https://codalab.lisn.upsaclay.fr/competitions/213#learn_the_details`.

Wilson, D. B. (2000), 'How to couple from the past using a read-once source of randomness', *Random Structures and Algorithms* **16**(1), 85–113.

Zappa, E., Holmes-Cerfon, M. and Goodman, J. (2018), 'Monte carlo on manifolds: Sampling densities and integrating functions', *Communications on Pure and Applied Mathematics* **71**(12), 2609–2647.