A First Course in Monte Carlo Methods

D. Sanz-Alonso and O. Al-Ghattas

University of Chicago

List of Figures

List of Algorithms

Preface

Overview This is a concise mathematical introduction to Monte Carlo methods, a rich family of algorithms with far-reaching applications in science and engineering. Monte Carlo methods are an exciting subject for mathematical statisticians and computational and applied mathematicians: the design and analysis of modern algorithms are rooted in a broad mathematical toolbox that includes ergodic theory of Markov chains, Hamiltonian dynamical systems, transport maps, stochastic differential equations, information theory, optimization, Riemannian geometry, and gradient flows, among many others. These lecture notes celebrate the breadth of mathematical ideas that have led to tangible advancements in Monte Carlo methods and their applications. To accommodate a diverse audience, the level of mathematical rigor varies from chapter to chapter, giving only an intuitive treatment to the most technically demanding subjects. The aim is not to be comprehensive or encyclopedic, but rather to illustrate some key principles in the design and analysis of Monte Carlo methods through a carefully-crafted choice of topics that emphasizes timeless over timely ideas. Algorithms are presented in a way that is conducive to conceptual understanding and mathematical analysis —clarity and intuition are favored over state-of-the-art implementations that are harder to comprehend or rely on *ad-hoc* heuristics. To help readers navigate the expansive landscape of Monte Carlo methods, each algorithm is accompanied by a summary of its pros and cons, and by a discussion of the type of problems for which they are most useful. The presentation is self-contained, and therefore adequate for self-guided learning or as a teaching resource. Each chapter contains a section with bibliographic remarks that will be useful for computational scientists and graduate students interested in conducting research on Monte Carlo methods and their applications.

Acknowledgements These lecture notes developed as a result of several courses taught by Daniel Sanz-Alonso at the University of Chicago since 2019; Omar Al-Ghattas served as teaching assistant for two of these courses and contributed significantly to improve the presentation. The main target audience are advanced undergraduate and graduate students in computational mathematics, applied mathematics, and statistics. Our courses are also regularly taken by graduate students in financial mathematics, chemistry, physics, geophysics, economics, and public policy that are interested in using Monte Carlo methods in their research. A first draft was created by the students of the course Stochastic Simulation, STAT 31510, in Spring 2019. The students responsible for this first draft are: Weilin Chen, Fuheng Cui, Zhen Dai, Marlin Figgins, Kim Liu, Yuwei Luo, Shinpei Nakamura Sakai, David Noursi, Nick Rittler, Matthew Shin, Zihao Wang, Wen Yuan Yen, Joey Yoo, and Yanfei Zhou. We are grateful to these students, without whom this work would not exist, and to all the students that provided encouragement and constructive feedback over the years. We are particularly thankful to Zijian Wang for improving the material on Hamiltonian Monte Carlo and Gibbs samplers, to Shiv Agrawal and Yu-Chun Lai for first drafts on variational inference and hidden Markov models on discrete state-space, to

Zhaoming Li and Bobbi Shi for creating several figures, to Jiajun Bao, Haoyuan Mao and Adi Raman for developing new figures, exercises and Python notebooks, and to Ruiyi Yang for his feedback and for typesetting a first draft of the chapter on annealing strategies. Lanran Fang and Felix Poirier also contributed to develop new exercises.

Finally, we are grateful to DOE, FBBVA, NGIA and NSF for their support, which has facilitated the completion and shaped the presentation of these lecture notes.

Warning This is a preliminary abridged version of a forthcoming textbook on Monte Carlo methods, which will contain numerous exercises and additional references. This preliminary version is likely to contain typographical and mathematical errors, inconsistencies in notation, and incomplete bibliographical information. We hope that it is nonetheless useful. Please contact the authors with any feedback from typos, through mathematical errors and bibliographical omissions, to comments on the structural organization of the material. Instructors interested in exercises to use in their courses are particularly welcome to contact the authors.

D. Sanz-Alonso and O. Al-Ghattas
University of Chicago

Contents

Chapter 1
Introduction

This chapter provides a high-level introduction to Monte Carlo methods and to some overarching themes and ideas in this course. We start with a motivating example in Section 1.1, where we explain how a simple Monte Carlo method can be used to approximate the area π of the unit circle. Next, we introduce two guiding areas of application in Section 1.2: Bayesian statistics and statistical mechanics. Both of these application domains illustrate the need to develop Monte Carlo methods that scale well to high dimension and that are implementable when the normalizing constant of the target distribution is unknown. Section 1.3 overviews these and other challenges that play an important role in the design of Monte Carlo methods. Section 1.4 summarizes some key principles and ideas that underlie the development of many Monte Carlo methods and sampling algorithms. Section 1.5 discusses the structure of these lecture notes and how to use them as a teaching resource. The chapter closes in Section 1.6 with historical and bibliographical remarks.

1.1 ▪ Motivating Example

Monte Carlo methods represent a quantity of interest as a parameter of a distribution and use a random sample from that distribution to estimate the parameter. As an illustrative example, this section describes a Monte Carlo method to approximate π, the area of the unit circle C. Let $(X_1, X_2) \sim \text{Unif}([-1,1] \times [-1,1])$ be a random vector with uniform distribution on the square $S = [-1,1] \times [-1,1]$. Then,

$$p := \mathbb{P}\Big((X_1, X_2) \in C\Big) = \mathbb{E}\Big[\mathbf{1}_{\{(X_1,X_2)\in C\}}\Big] = \frac{\text{Area C}}{\text{Area S}} = \frac{\pi}{4}.$$

Hence, $\pi = 4p$, and we have expressed the quantity of interest π in terms of a probability p. Now, we can estimate p —and hence π— using a sample $\big(X_1^{(1)}, X_2^{(1)}\big), \ldots, \big(X_1^{(N)}, X_2^{(N)}\big)$ of N independent random vectors uniformly distributed on S. Precisely, define

$$B_N = \sum_{n=1}^{N} \mathbf{1}_{\left\{\left(X_1^{(n)}, X_2^{(n)}\right)\in C\right\}}.$$

Then, $B_N \sim \text{Binomial}(N, p)$ and we can estimate p by $\widehat{p}_N = B_N/N$ and π by $\widehat{\pi}_N = 4\widehat{p}_N$. For example, as shown in Figure 1.1, if $N = 10^4$ and we observe $B_{10000} = 7854$, we would estimate

$$\widehat{\pi}_{10000} = 4 \times \frac{7854}{10000} = 3.1416.$$

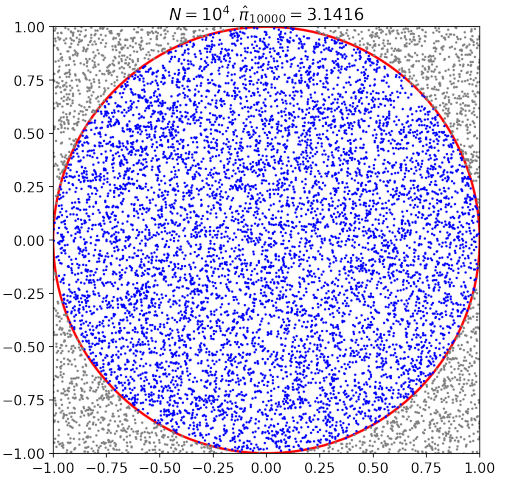


Figure 1.1. *Estimation of π with sample size $N = 10^4$. Here, $B_{10000} = 7854$ draws fell within the unit circle, leading to an estimate $\widehat{\pi}_{10000} = 3.1416$.*

If N is very large, the probability —*before the random sample is generated*— that our estimation is poor is rather small. For $\alpha \in (0,1)$, define the z-score $z_{\alpha/2}$ by the requirement that $\mathbb{P}(-z_{\alpha/2} < Z < z_{\alpha/2}) = 1-\alpha$, where $Z \sim \mathcal{N}(0,1)$. Then,

$$\left(\widehat{p}_N - z_{\alpha/2}\sqrt{\frac{\widehat{p}_N(1-\widehat{p}_N)}{N}}, \widehat{p}_N + z_{\alpha/2}\sqrt{\frac{\widehat{p}_N(1-\widehat{p}_N)}{N}}\right)$$

is a $1-\alpha$ confidence interval for p, and therefore

$$\left(\widehat{\pi}_N - z_{\alpha/2}\sqrt{\frac{\widehat{\pi}_N(4-\widehat{\pi}_N)}{N}}, \widehat{\pi}_N + z_{\alpha/2}\sqrt{\frac{\widehat{\pi}_N(4-\widehat{\pi}_N)}{N}}\right)$$

is a $1-\alpha$ confidence interval for π. In the example shown in Figure 1.1 where $N = 10^4$ and $\widehat{\pi}_{10000} = 3.1416$, using a z-score of 1.96 for a 95% confidence level, we obtain the confidence interval $(3.1094, 3.1738)$.

Let us summarize:

1. We have written the quantity of interest (in our case π) as an expectation, using that for any event A, $\mathbb{P}(A) = \mathbb{E}[\mathbf{1}_A]$. In this course, the quantity of interest we seek to approximate via Monte Carlo methods will often be an expected value.

2. We have constructed an estimator of the expectation in step 1 based on a random sample.

3. We have used statistical theory to assess the quality of the estimator. The law of large numbers guarantees that the sample approximation converges to the quantity of interest. Furthermore, the central limit theorem quantifies the speed of convergence and can be used to obtain confidence intervals.

1.2 ▪ Guiding Applications

This course is primarily concerned with Monte Carlo methods to compute integrals of the form

$$\mathcal{I}_f[h] := \int_E h(x)f(x)\,dx = \mathbb{E}_{X\sim f}\big[h(X)\big], \tag{1.1}$$

where f is a given *target distribution* supported on a set $E \subset \mathbb{R}^d$ and $h : E \to \mathbb{R}$ is a given *test function*. For instance, taking $h(x) = \mathbf{1}_A(x)$ the indicator function of a subset $A \subset E$, we can then compute the probability $\mathbb{P}_{X\sim f}(X \in A)$. In this section, we describe two application domains where computing integrals of the form (1.1) is important: Bayesian statistics and statistical mechanics. In both, the dimension d can be exceedingly large and the target distribution f is only known up to a normalizing constant, making the use of deterministic quadrature algorithms unfeasible and posing interesting challenges to the design of Monte Carlo methods.

Bayesian Statistics Given a likelihood model $f(y|\theta)$ and a prior density $f(\theta)$, the posterior density of the parameter θ given the data y is given by

$$f(\theta|y) = \frac{1}{c} f(y|\theta)f(\theta), \tag{1.2}$$

where $c = \int f(y|\theta)f(\theta)\,d\theta$ is a normalizing constant so that $f(\theta|y)$ integrates to 1. Bayesian inference relies on computing expectations of the form

$$\mathcal{I}_f[h] = \int h(\theta)f(\theta|y)\,d\theta,$$

where the posterior $f(\theta|y)$ is the target distribution and h is a test function. The most important choice of test function is arguably $h(\theta) = \theta$, which gives the posterior mean estimator of θ. Other choices of test function enable computation of credible intervals —regions of the parameter space with prescribed posterior probability— and higher posterior moments for uncertainty quantification.

Before the advent of Monte Carlo methods, Bayesian inference was only computationally feasible in low dimension and under the restrictive model assumption of conjugate priors: given a likelihood, choosing a conjugate prior ensures that the posterior belongs to the same family of distributions as the prior, whereby point estimators and credible intervals may admit closed form expressions. In modern applications, non-conjugate models are routinely used and the parameter θ can be high-dimensional; think, for instance, that θ may represent a vectorized high-resolution image which we may want to denoise. Then, the normalizing constant c in (1.2) —which represents the *marginal likelihood* of the data— is typically unknown and hard to compute, as it is itself defined as an integral on a high-dimensional parameter space. This course emphasizes the importance of designing Monte Carlo methods that scale well to high dimension and that are applicable when the normalizing constant of the target distribution is unknown.

The development of Monte Carlo methods goes hand in hand with the popularization of large-scale Bayesian statistics. The Bayesian approach to statistics allows to conveniently formulate hierarchical models and to do sequential inference by updating beliefs as new data arrives. Gibbs samplers and particle filters —two classes of Monte Carlo methods studied in this course— dovetail with hierarchical and sequential inference, respectively.

Statistical Mechanics Another important source of challenging sampling problems is statistical mechanics, and in particular the simulation of molecular dynamics. Boltzmann postulated

that the positions q and momenta p of the atoms in a molecular system of constant size, occupying a constant volume, and in contact with a heat bath (at constant temperature), are distributed according to

$$f(q,p) = \frac{1}{c}\exp\Big(-\beta\big(V(q)+K(p)\big)\Big),$$

where c is a normalizing constant known as the partition function, β represents the inverse temperature, V is a potential energy describing the interaction of the particles in the system, and K represents the kinetic energy of the system. The potential V is often a very rough function with many local minima. This and other typical features of the potential make computing averages with respect to $f(p,q)$ very difficult. However, many quantities of interest are defined as averages with respect to the Boltzmann distribution, and it is therefore of great scientific interest to compute integrals of the form

$$\mathcal{I}_f[h] = \int h(q,p)f(q,p)\,dq\,dp.$$

Similarly as in Bayesian statistics, computing the normalizing constant c of the Boltzmann (also known as Gibbs) distribution $f(q,p)$ is challenging, which motivates again the need to develop Monte Carlo methods for densities that are known up to a normalizing constant.

1.3 ▪ Six Challenges

To understand the design and motivation behind different Monte Carlo methods, it is essential to first appreciate the main challenges they attempt to tackle. This section provides a high-level summary of some of these challenges. Clearly, no single algorithm is best at simultaneously addressing all of them; otherwise, there would be no need to introduce the many types of Monte Carlo methods studied in this course! Throughout this course, each new algorithm we introduce will be accompanied by a discussion of its pros and cons in relation to addressing different challenges, thus helping explain its role within the vast landscape of Monte Carlo methods.

Challenge 1: High Dimension Monte Carlo integration methods are highly effective in high dimension, *provided that one can obtain a sample from the target distribution*. Unfortunately, sampling from a generic target distribution in high dimension is often a difficult task. On the bright side, there are important classes of target distributions for which scalable sampling algorithms are available. For instance, log-concave target distributions can be efficiently sampled in high dimension employing sampling algorithms reminiscent of convex optimization algorithms (Chapter 6). As another example, if one can find a *proposal distribution* that is close to the target and easy to sample from, then samples from the proposal can be turned into samples from the target (Chapters 2 and 3).

Challenge 2: Evaluating the Target and its Gradient The guiding applications to Bayesian statistics and statistical mechanics in Section 1.2 showcase that oftentimes the target distribution can only be evaluated up to an unknown normalizing constant. In some Bayesian inference problems, evaluating the likelihood function can also be challenging, which has motivated the development of approximate Bayesian computation algorithms, likelihood-free methods, and Monte Carlo algorithms for big data (Chapter 2). Relatedly, some Monte Carlo methods leverage the gradient of the logarithm of the target density to improve the scalability to high dimension (Chapters 6 and 8), but evaluating these gradients can be computationally expensive.

Challenge 3: Multi-Modality Many sampling algorithms studied in this course are *local* in the sense that each new draw typically lies on a small neighborhood around the previous one (Chapters 4, 6, and 8). For these methods, sampling multi-modal target distributions poses a significant challenge, especially when regions of high target density are separated by regions of low density. In this setting, local sampling methods may over-sample around one mode without adequately exploring others. Many strategies have been developed to address this issue, such as relying on flattened or *tempered* ancillary target distributions to speed-up exploration (Chapter 7), or introducing auxiliary variables to define an extended target in a higher dimensional space with a more benign landscape (Chapters 7 and 8).

Challenge 4: Computing Rare Events A rather different set of challenges arises when the test function $h(x) = \mathbf{1}_A(x)$ is the indicator function of a small probability event A. In that case, a random sample from the target distribution may typically include no draws that lie in the set A of interest, and the Monte Carlo estimate for $\mathbb{P}(A)$ would simply be 0. To obtain an estimate with low *relative error*, it is then paramount to obtain samples from a different proposal distribution, rather than the target (Chapters 3 and 9). How should one choose the proposal distribution in order to minimize the variance of Monte Carlo estimates of rare events? You will find out in Chapter 3.

Challenge 5: Choice of Variables and Conditioning of the Target For some algorithms, an important challenge is to sample joint distributions with strongly correlated variables (Chapter 5) or with variables that have different scales (Chapters 4, 6, and 8). To alleviate this issue, it is useful, whenever possible, to carefully parameterize the problem before utilizing Monte Carlo methods. Other techniques that help alleviate this issue include blocking together correlated variables, and adequately preconditioning sampling algorithms.

Challenge 6: Assessing Convergence Assessing the convergence of Monte Carlo methods can be challenging. Diagnostics can be utilized to rule out convergence, but these diagnostics can still not guarantee convergence (Chapter 4). This challenge is particularly conspicuous when sampling multi-modal distributions, since it is hard to identify lack of converge for algorithms that have adequately sampled around one mode but failed to explore all modes.

1.4 ▪ Ten Ideas

This section summarizes some key ideas that underlie the design of many Monte Carlo methods.

Idea 1: Deterministic Transformations To sample from a given target, deterministic transformation methods rely on a transport map to turn draws from an easy-to-sample proposal distribution into draws from the target (Chapter 2). Under mild assumptions on target and proposal, there are many deterministic transport maps between them. Deterministic transformation methods replace the question of how to sample the target with the question of how to parameterize and learn a transport map between proposal and target distributions.

Idea 2: Probabilistic Accept/Reject Mechanisms In their simplest form, probabilistic accept/reject methods rely on a probabilistic mechanism to turn draws from a proposal distribution into draws from a given target (Chapter 2). A related idea also underlies Markov chain Monte Carlo methods (Chapter 4), where a probabilistic accept/reject mechanism is used to turn samples from a proposal Markov kernel into samples from a Markov kernel that satisfies detailed

balance with respect to the target. The unifying idea is to leverage a probabilistic rule to turn easy-to-obtain samples into approximate samples from a given target distribution.

Idea 3: Weighting Samples Rather than deterministically transforming or probabilistically accepting/rejecting samples, some Monte Carlo methods assign weights to draws from a proposal distribution to approximate expected values with respect to the target (Chapters 3 and 9). While not computing a transport map and not rejecting any samples via a probabilistic accept/reject mechanism sounds appealing, a caveat of Monte Carlo methods that rely on weighted samples is that the variance of the weights is typically large when target and proposal are far apart. As a result, one draw from the proposal may receive an exceedingly large weight compared to the others, leading to a small effective sample size. Such weight degeneracy is particularly common in high dimension.

Idea 4: Markov Chains and Local Moves Some of the most popular Monte Carlo methods rely on random samples that are not independent, but that instead form a Markov chain: the distribution of each new random draw depends on the value of the previous one (Chapters 4, 5, 6, and 8). Positive correlation between samples increases the variance of Monte Carlo estimates (Appendix A), but the great flexibility afforded by the Metropolis Hastings framework for Markov chain Monte Carlo makes up for this caveat (Chapter 4). In particular, proposal Markov kernels that leverage the target and its gradient can be utilized to speed-up convergence in high dimension.

Idea 5: Discretization of (Stochastic) Differential Equations A key idea to design proposal kernels for Markov chain Monte Carlo algorithms is to leverage Langevin stochastic differential equations and Hamiltonian differential equations that preserve the target (Chapters 6 and 8). Time discretizations of these stochastic and ordinary differential equations provide natural proposal Markov kernels, which, if desired, can be corrected via a probabilistic accept/reject mechanism to ensure convergence to the target.

Idea 6: Optimization Optimization and sampling share numerous conceptual similarities, and their synergistic combination can be leveraged for algorithmic design. First, many methods to sample log-concave target distributions are similar to convex optimization algorithms: while gradient descent methods can be viewed as time-discretizations of gradient systems in parameter space, Langevin Monte Carlo methods can be viewed as discretizations of gradient flows in the space of probability densities (Chapter 6). Second, annealing algorithms for global optimization sample from a sequence of target distributions that become more peaked around their mode, thus leveraging sampling for optimization (Chapter 7). As a final example, variational inference methods find the closest tractable distribution to a given target, thus solving an optimization problem over densities to approximate integrals with respect to a given target (Chapter 10).

Idea 7: Annealing and Tempering To facilitate sampling multi-modal distributions, a powerful idea is to introduce auxiliary tempered distributions. In this direction, a popular strategy is parallel tempering, where one runs several tempered Markov chains in parallel occasionally swapping their states (Chapter 7). Tempering is also useful in Bayesian inference applications, where instead of directly targeting the posterior with the full data, one may introduce a sequence of intermediate tempered posteriors with smaller batches of data.

Idea 8: Auxiliary Variables Many Monte Carlo methods speed-up convergence by introducing auxiliary variables. In this course, you will study many algorithms that build on this idea,

including simulated tempering (Chapter 7), Hamiltonian Monte Carlo (Chapter 8), and the slice sampler (Chapter 5). Auxiliary variables also play an important role in sequential Monte Carlo methods (Chapter 9).

Idea 9: Coordinate-Wise Updates The Gibbs sampler and the coordinate ascent variational inference algorithm are coordinate-wise sampling and optimization algorithms, where joint distributions are iteratively sampled or approximated by updating only one coordinate at a time (Chapters 5 and 10). Because of the coordinate-wise structure of these algorithms, they are applicable in high dimension. Gibbs samplers are very effective in sampling from intractable joint distributions with easy-to-sample conditionals. Coordinate ascent variational inference is particularly appealing in exponential family models, where closed form formulas can be used to update the parameters.

Idea 10: Minimization-Maximization This course also covers the expectation maximization algorithm (Chapter 10), an important example of a minimization-maximization method that plays a similar role in maximum likelihood estimation as the Gibbs sampler does in posterior sampling; in particular the expectation maximization algorithm can be used to initialize the Gibbs sampler.

1.5 ▪ Course Structure

These lecture notes developed as a result of several courses taught at the University of Chicago since 2019. Here, we discuss the structure of the course Monte Carlo Simulation CAAM/STAT 31511 as taught in spring 2024. The prerequisites were multivariate calculus, linear algebra, a first course in probability and statistics (including Markov chains), and elementary knowledge of ordinary differential equations. We covered all the material in these notes, following the schedule outlined in Table 1.1 below. To review Markov chain theory in preparation for the class, students were encouraged to read Appendix A and to consult some of the standard textbooks on Markov chains and stochastic processes referenced therein.

The course had weekly homework assignments, each corresponding to one chapter. These 9 homework assignments accounted for 72% of the grade and we will include them in a future version of these notes. In addition, students had to complete a 10-page independent project on a topic of their choice, which accounted for the remaining 28% of the grade. The goal of this independent project was for students to learn more deeply about a topic of their choice, giving them freedom to focus on theory, methodology, or applications.

Numerous modifications of this timeline and grading scheme can be considered. At places that follow a semester rather than a quarter system, we would review Markov chain theory before teaching the Metropolis Hastings algorithm in Chapter 4. We would also include group projects as part of the grading, and have students present their projects to the class over the last two weeks of the semester.

1.6 ▪ Discussion and Bibliography

One of the first documented Monte Carlo experiments is Buffon's needle experiment in 1733; see [51] for a solution to this problem. The paper [134] suggested that this experiment could be used to approximate π. Historically, the main drawback of Monte Carlo methods was that they used to be expensive to carry out. Physical random experiments were difficult to perform, and so was the numerical processing of their results. This however changed fundamentally with the

Week	Date	Lecture Notes	Homework
1	3/19	Chapter 1	
	3/21	Chapter 2	
2	3/26	Chapter 2	
	3/28	Chapter 3	Homework 1
3	4/2	Chapter 3	
	4/4	Chapter 4	Homework 2
4	4/9	Chapter 4	
	4/11	Chapter 5	Homework 3
5	4/16	Chapter 5	
	4/18	Chapter 6	Homework 4
6	4/23	Chapter 6	
	4/25	Chapter 7	Homework 5
7	4/30	Chapter 8	
	5/2	Chapter 8	Homework 6
8	5/7	Chapter 9	
	5/9	Chapter 9	Homework 7
9	5/14	Chapter 10	
	5/16	Chapter 10	Homework 8
	5/23		Homework 9

Table 1.1. *Structure of the course Monte Carlo Simulation CAAM/STAT 31511 as taught in spring 2024.*

advent of the digital computer. Among the first to realize this potential were John von Neumann and Stanislaw Ulam, who were then working for the Manhattan project in Los Alamos; see [67]. They proposed in 1947 to use a computer simulation for solving the problem of neutron diffusion in fissionable material. Enrico Fermi previously considered using Monte Carlo techniques in the calculation of neutron diffusion using a mechanical device, the so-called "Fermiac," for generating the randomness. The name "Monte Carlo" goes back to Ulam, who claimed to be stimulated by playing poker and whose uncle once borrowed money from him to go gambling in Monte Carlo (unpublished remarks by Ulam in 1983). In 1949 Metropolis and Ulam published their results in the Journal of the American Statistical Association [163]. Nonetheless, in the following 30 years Monte Carlo methods were used and analyzed predominantly by physicists, and not by statisticians: it was only in the 1980s —following the paper [82] that proposed the Gibbs sampler— that the relevance of Monte Carlo methods in the context of Bayesian statistics was fully realized.

Chapter 2

Transformation and Accept/Reject Sampling Methods

This chapter is concerned with sampling methods: algorithms to generate random draws from a given distribution. To be concrete, suppose that the *target distribution* that we want to sample from has p.d.f. f on a subset $E \subset \mathbb{R}^d$. The goal is to obtain a *sample* $X^{(1)}, \ldots, X^{(N)} \in E$ satisfying that each independent draw $X^{(n)} \in E$, $1 \leq n \leq N$, is randomly distributed with p.d.f. f. Intuitively, a histogram of the sample should accurately approximate the target p.d.f. provided that the sample size N is sufficiently large. We focus on transformation and accept/reject sampling methods to showcase two important overarching themes in the development of Monte Carlo methods: the use of deterministic transformations and probabilistic accept/reject mechanisms to transform samples from a tractable distribution into samples from the target distribution.

Our starting point will be a "random" number generator to produce independent and uniformly distributed draws $U^{(1)}, \ldots, U^{(N)} \sim \text{Unif}(0, 1)$. In practice, random number generators are deterministic algorithms; each *pseudo-random* number $U^{(n)}$ is obtained by iteratively applying a deterministic transformation to a deterministically chosen initial seed. While the procedure is deterministic and the resulting sample is not truly random, it imitates the behavior of a uniform sample in that statistical tests for departure from independence and the uniform distribution are not rejected more often than would be expected by chance. Similarly, the sampling methods considered in this chapter are not truly random, but imitate the statistical behavior of a sample drawn from the target distribution. Throughout this course, we assume to have available a random number generator, that is, a computer code that outputs values $U^{(1)}, \ldots, U^{(N)}$ that are statistically indistinguishable from an i.i.d. sample from the $\text{Unif}(0, 1)$ distribution.

When the target distribution is not uniform and does not belong to a standard parametric model (e.g. normal, exponential, geometric, etc.), sampling can be challenging, especially in high dimension. In this chapter, we consider two general approaches for sampling: transformation methods (Section 2.1) and accept/reject methods (Section 2.2). Both approaches first produce a sample from a *reference* or *proposal distribution* that is tractable in the sense that it is easy to sample from. Transformation methods find a map to turn draws from the reference distribution into draws from the target distribution; accept/reject methods rely on a probabilistic mechanism to turn draws from the reference distribution into draws from the target distribution. The chapter closes in Section 2.3 with bibliographic remarks. For simplicity, we assume throughout that the target distribution is discrete or continuous, and characterized by a p.m.f. or p.d.f. denoted by f. Similarly, the reference distribution will be characterized by a p.m.f. or p.d.f. denoted by g.

2.1 ▪ Transformation Methods

Let $X \sim f$ be a random variable taking values on a set $E \subset \mathbb{R}^d$ and let $U \sim \text{Unif}(0,1)^d$ be a random variable uniformly distributed on the d-dimensional unit cube $(0,1)^d$. In this section, we seek to find a map $\mathcal{T} : (0,1)^d \to E$ so that the random variable $\mathcal{T}(U)$ has the same distribution as X, written $X \overset{d}{=} \mathcal{T}(U)$. Once we find such a *push-forward* or *transport* map $\mathcal{T}$, we can obtain a sample from the target distribution f by:

1. Sampling $U^{(1)}, \ldots, U^{(N)} \overset{\text{i.i.d.}}{\sim} \text{Unif}(0,1)^d$; and
2. Setting $X^{(n)} := \mathcal{T}(U^{(n)})$ for $1 \leq n \leq N$.

It is natural to ask if such a transport map $\mathcal{T}$ is guaranteed to exist. We will provide an affirmative answer and an explicit construction in this section under the assumption that the target admits a positive p.d.f. For exposition purposes, we consider first the scalar case $d = 1$ in Subsection 2.1.1, and then generalize to the d-dimensional case in Subsection 2.1.2.

2.1.1 ▪ Inverse Transformation Method

Let $X \sim f$ be a real-valued random variable, and denote by F its c.d.f. The inverse transformation method relies on the (generalized) inverse c.d.f. of X, given by

$$F^-(u) := \inf\{x : F(x) \geq u\},$$

to transform a uniform sample into a sample from f. The following result shows that the inverse c.d.f. is a valid transport map.

Theorem 2.1 (Transport: Uniform to Target). *Let X be a real-valued random variable with inverse c.d.f. F^- and let $U \sim \text{Unif}(0,1)$. It holds that $F^-(U) \overset{d}{=} X$.*

Proof. We only prove here the case where F is a bijection from $\mathbb{R}$ to $(0,1)$. In such a case, it holds that $F^- = F^{-1}$, as illustrated in Figure 2.1. Then, for any $x \in \mathbb{R}$,

$$\mathbb{P}\big(F^-(U) \leq x\big) = \mathbb{P}\big(F^{-1}(U) \leq x\big) = \mathbb{P}\big(U \leq F(x)\big) = F(x),$$

which shows that F is the c.d.f. of the random variable $F^-(U)$. The general case where the c.d.f. F may not be invertible is left as an exercise. □

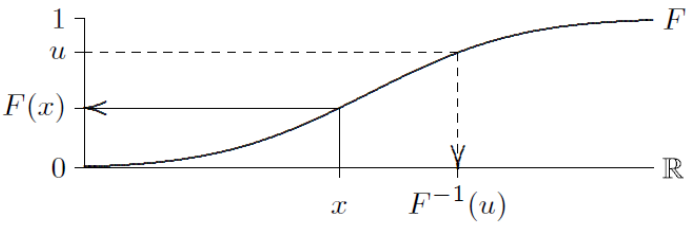


Figure 2.1. *A strictly increasing c.d.f. and associated inverse c.d.f.*

The inverse transformation method produces a sample from the target distribution as follows:

Algorithm 2.1 Inverse Transformation Method

1: **Input**: Target distribution f with inverse c.d.f. F^-, sample size N.
2: Set $X^{(n)} := F^-(U^{(n)}), \quad 1 \leq n \leq N, \quad$ where $U^{(1)}, \ldots, U^{(N)} \overset{\text{i.i.d.}}{\sim} \text{Unif}(0,1)$.
3: **Output**: Sample $\{X^{(n)}\}_{n=1}^{N} \overset{\text{i.i.d.}}{\sim} f$.

Pros and Cons: The inverse transformation method is very efficient whenever F^- has a closed form expression that can be evaluated cheaply. However, Algorithm 2.1 is only applicable to sample real-valued random variables.

Example 2.2 (Inverse Transformation Method: Exponential Distribution). Suppose that $X \sim \text{Exponential}(\lambda)$, so that $F(x) = 1 - e^{-\lambda x}$, $x \geq 0$. Let $U \sim \text{Unif}(0,1)$. Then,

$$-\frac{1}{\lambda}\log(1-U) \sim \text{Exponential}(\lambda).$$

Using that $1 - U \overset{d}{=} U$, we also have that

$$-\frac{1}{\lambda}\log(U) \sim \text{Exponential}(\lambda).$$

Figure 2.2 shows a histogram of $N = 10^5$ draws generated from an Exponential(1) distribution using the inverse transformation method. □

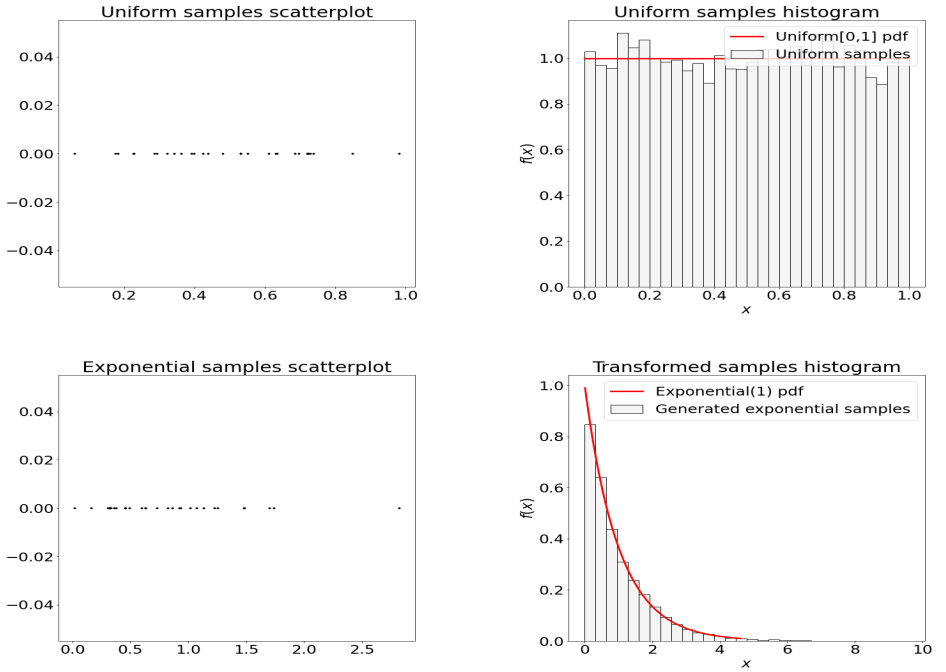


Figure 2.2. *Inverse transformation method for sampling from an Exponential(1) distribution. Uniform samples (first row) are transformed into samples from the target distribution (second row).*

So far, we have discussed how to transform a uniform sample into a sample from a given target distribution. In some applications, it is important to use a proposal distribution that is not uniform. The following theorem provides a transport map from proposal to target.

Theorem 2.3 (Transport: Proposal to Target). *Let Z be a real-valued random variable with continuous and strictly increasing c.d.f. F_Z and let X be a real-valued random variable with inverse c.d.f. F_X^-. Then, it holds that $F_X^- \circ F_Z(Z) \stackrel{d}{=} X$.*

Proof. By Theorem 2.1, it suffices to show that $F_Z(Z) \sim \text{Unif}(0,1)$. To that end, note that, for any $z \in (0,1)$,

$$\mathbb{P}\big(F_Z(Z) \leq z\big) = \mathbb{P}\big(Z \leq F_Z^{-1}(z)\big) = F_Z\big(F_Z^{-1}(z)\big) = z,$$

as desired. □

Let $X \sim f$ and let $Z \sim g$, where g is a strictly positive p.d.f. In view of Theorem 2.3, we can obtain a draw from f by (1) drawing $Z^{(n)} \sim g$; and (2) setting $X^{(n)} := F_X^- \circ F_Z(Z^{(n)})$.

2.1.2 ▪ Knothe-Rosenblatt Rearrangement

We now generalize the inverse transformation method to target distributions on $\mathbb{R}^d$, $d \in \mathbb{N}$. For ease of exposition, let us consider first the case $d = 2$. Let $X = (X_1, X_2) \sim f$ and let $U = (U_1, U_2) \sim \text{Unif}(0,1)^2$. The idea is to *disintegrate* the target f as the product of the marginal of X_1 times the conditional of X_2 given X_1:

$$f(x) = f_{X_1}(x_1) f_{X_2|X_1=x_1}(x_2|x_1), \quad x = (x_1, x_2) \in \mathbb{R}^2.$$

We then sequentially apply the inverse transformation method to the one-dimensional targets f_{X_1} and $f_{X_2|X_1=x_1}$ as will be detailed next.

Denote by F_{X_1} the marginal c.d.f. of X_1. By the inverse transformation method, we know that $F_{X_1}^-(U_1) \stackrel{d}{=} X_1$. Now, for any $x_1 \in \mathbb{R}$ denote by $F_{X_2|X_1=x_1}$ the c.d.f. of $X_2|X_1 = x_1$. By the inverse transformation method, we know that $F_{X_2|X_1=x_1}^-(U_2) \stackrel{d}{=} X_2|X_1 = x_1$. To summarize, define recursively maps $\mathcal{T}^1$ and $\mathcal{T}^2$ by

$$\begin{aligned} \mathcal{T}^1 &: (0,1) \to \mathbb{R}, & \mathcal{T}^2 &: (0,1)^2 \to \mathbb{R}, \\ & u_1 \mapsto F_{X_1}^-(u_1), & & (u_1, u_2) \mapsto F_{X_2|X_1=\mathcal{T}^1(u_1)}^-(u_2). \end{aligned}$$

Then, the triangular map $\mathcal{T} : (0,1)^2 \to \mathbb{R}^2$ given by

$$\mathcal{T}(u) = \begin{bmatrix} \mathcal{T}^1(u_1) \\ \mathcal{T}^2(u_1, u_2) \end{bmatrix}, \qquad u = (u_1, u_2) \in \mathbb{R}^2, \tag{2.1}$$

satisfies that $\mathcal{T}(U) \stackrel{d}{=} X$. The map $\mathcal{T}$ defined in (2.1) is called the *Knothe-Rosenblatt rearrangement* from $\text{Unif}(0,1)^2$ to f. The construction can be generalized to higher dimension:

Theorem 2.4 (Transport via Triangular Maps). *Let f be a strictly positive p.d.f. on $\mathbb{R}^d$. Let*

$X \sim f$ and let $U \sim \text{Unif}(0,1)^d$. For $1 \le i \le d$, define maps $\mathcal{T}^i$ as follows:[1]

$$\mathcal{T}^1(u_1) = F^-_{X_1}(u_1), \tag{2.2}$$

$$\mathcal{T}^i(u_{1:i}) := F^-_{X_i|X_{1:i-1}=\mathcal{T}^{1:i-1}(u_{1:i-1})}(u_i), \qquad 2 \le i \le d. \tag{2.3}$$

Then, the triangular map

$$\mathcal{T}(u) = \begin{bmatrix} \mathcal{T}^1(u_1) \\ \mathcal{T}^2(u_1, u_2) \\ \vdots \\ \mathcal{T}^d(u_1, \ldots, u_d) \end{bmatrix}, \qquad u = (u_1, \ldots, u_d) \in \mathbb{R}^d, \tag{2.4}$$

satisfies that $X \overset{d}{=} \mathcal{T}(U)$. Moreover, for all $1 \le i \le d$, the components $\mathcal{T}^i : (0,1)^d \to \mathbb{R}$ are monotonically increasing in the i-th variable.

The map $\mathcal{T}$ defined in Theorem 2.4 is called the Knothe-Rosenblatt (KR) rearrangement from $\text{Unif}(0,1)^d$ to f. The map $\mathcal{T}$ is referred to as a *triangular* map since its Jacobian is a (lower) triangular matrix. The KR rearrangement can be used for sampling, as summarized in the following algorithm:

Algorithm 2.2 Sampling via the KR Rearrangement

1: **Input**: KR rearrangement $\mathcal{T}$ from $\text{Unif}(0,1)^d$ to f, sample size N.
2: Set $X^{(n)} := \mathcal{T}(U^{(n)}), \quad 1 \le n \le N, \quad$ where $U_1, \ldots, U^{(N)} \overset{\text{i.i.d.}}{\sim} \text{Unif}(0,1)^d$.
3: **Output**: Sample $\{X^{(n)}\}_{n=1}^N \overset{\text{i.i.d.}}{\sim} f$.

Pros and Cons: As with other transformation methods, sampling via the KR rearrangement is very efficient if we can find and evaluate the KR map. However, computing the KR rearrangement can be expensive in high dimension. Conditional independence of the target-reference pair leads to sparsity in the KR map that can be exploited when computing or estimating the KR map. We refer to Section 2.3 for further discussion on this topic.

Using Theorem 2.3, it is possible to generalize the construction of the KR rearrangement to transport a strictly positive proposal p.d.f. g on $\mathbb{R}^d$ into a strictly positive target p.d.f. f. Such generalization is left as an exercise.

2.2 ▪ Accept/Reject Methods

In this section, we study rejection sampling as a prototypical example of an *accept/reject* method. The idea is to sample from a *proposal* (also known as reference or instrumental) distribution and discard samples that are unlikely under the *target* distribution of interest. We denote the target distribution by f and the proposal distribution by g. Typically, the target is expensive or impossible to sample from directly, but the proposal is easy to sample from. We present rejection sampling in Subsection 2.2.1 and a related accept/reject approach for Approximate Bayesian Computation (ABC) in Subsection 2.2.2.

[1] For vector $u = (u_1, \ldots, u_d) \in \mathbb{R}^d$ and $1 \le i \le d$, we denote by $u_{1:i} := (u_1, \ldots, u_i) \in \mathbb{R}^i$ the vector defined by the first i coordinates of u. Likewise, for triangular map $\mathcal{T} : \mathbb{R}^d \to \mathbb{R}^d$, we denote by $\mathcal{T}^{1:i} : \mathbb{R}^i \to \mathbb{R}^i$ the map defined by the first i components of $\mathcal{T}$.

2.2.1 ▪ Rejection Sampling

Before introducing the rejection sampling algorithm, we start with a simple identity and an example, both of which will correspond to a uniform proposal distribution g in the subsequent developments.

First, the identity

$$f(x) = \int_0^{f(x)} 1 \, du \tag{2.5}$$

shows that $f(x)$ is the marginal of a random vector that is uniformly distributed on the area under the density $f(x)$, that is, on the region $\{(x, u) : 0 \leq u \leq f(x)\}$. In the next example, we use the identity (2.5) to obtain an algorithm to sample from a beta distribution.

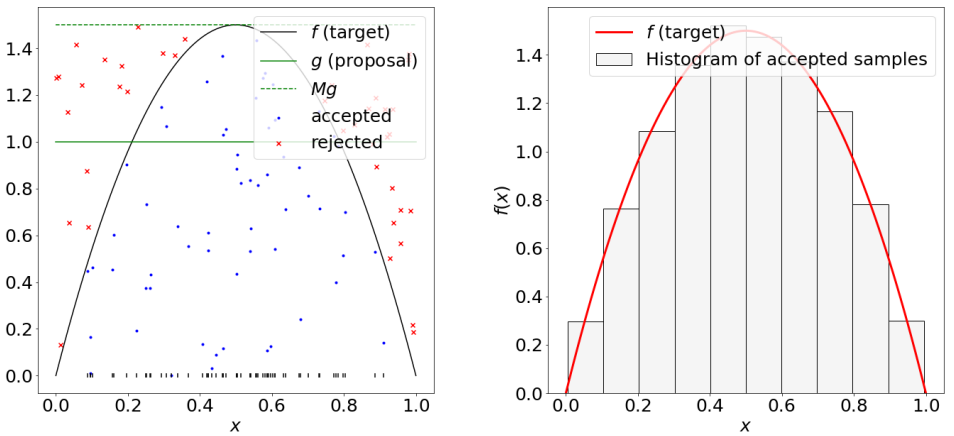


Figure 2.3. *Rejection sampling for a Beta*$(2, 2)$ *distribution, see Example 2.5. The target p.d.f. is given by* $f(x) = 6x(1 - x)$ *for* $x \in (0, 1)$. *Proposed samples are obtained by uniformly sampling in the rectangle with base* 1 *and height* M. *We accept those samples that lie below the curve of* f. *The first coordinate of each accepted sample is distributed according to the target.*

Example 2.5 (Rejection Sampling: Beta Distribution). Let $f = \text{Beta}(\alpha, \beta)$ so that $f(x) \propto x^{\alpha-1}(1 - x)^{\beta-1}$ for $x \in (0, 1)$, and suppose that $\alpha, \beta > 1$. The identity (2.5) suggests that to obtain a draw $X^{(n)} \sim f$, we could draw uniformly in the rectangle with base 1 and height M until we get a draw under the curve f (see Figure 2.3), and then keep the first coordinate. The following pseudocode formalizes the procedure:

1: For $n = 1, \ldots, N$ do:
2: **Step 1:** Generate independently $Z^{(n)} \sim \text{Unif}(0, 1)$ and $U^{(n)} \sim \text{Unif}(0, 1)$, so $(Z^{(n)}, MU^{(n)})$ is uniform on the rectangle $[0, 1] \times [0, M]$.
3: **Step 2:** Set $X^{(n)} = Z^{(n)}$ if $f(Z^{(n)}) \geq MU^{(n)}$ (that is, if $(Z^{(n)}, MU^{(n)})$ lies under the curve f). Otherwise, go back to step 1.

Any $X^{(n)}$ defined this way has p.d.f. f as desired. Note that the conditional probability that $(Z^{(n)}, MU^{(n)})$ is accepted if $Z^{(n)} = z$ is

$$\mathbb{P}\Big(MU^{(n)} < f(Z^{(n)}) \Big| Z^{(n)} = z\Big) = \mathbb{P}\left(U^{(n)} < \frac{f(z)}{M}\right) = \frac{f(z)}{M}.$$

Since $U^{(n)}$ and $Z^{(n)}$ are independent, we can rewrite the above procedure as follows:

1: For $n = 1, \ldots, N$ do:
2: **Step 1:** Draw $Z^{(n)} \sim \text{Unif}(0, 1)$.
3: **Step 2:** Set $X^{(n)} := Z^{(n)}$ with probability $\frac{f(Z^{(n)})}{M}$. Otherwise, go back to step 1.

□

Example 2.5 uses that the density $\text{Beta}(\alpha, \beta), \alpha, \beta > 1$ is bounded within a (finite) rectangle. The proposal distribution can then be taken to be uniform. We now generalize the procedure to cases where the range or the domain of f are unbounded. To that end, we use a proposal g satisfying the following conditions:

- g can be sampled from and evaluated.
- g has compatible support with f (i.e. $g(x) > 0$ when $f(x) > 0$).
- There is $M > 0$ such that, for all $x \in E$, $f(x) \leq Mg(x)$.

Notice that for the target $f = \text{Beta}(2, 2)$ in the example in Figure 2.3, the choices $g = \text{Unif}(0, 1)$ and $M = 3/2$ satisfy these conditions. We are ready to introduce the rejection sampling algorithm:

Algorithm 2.3 Rejection Sampling

Input: Target f, proposal g, constant $M > 0$ such that, for all $x \in E$, $f(x) \leq Mg(x)$, sample size N.
For $n = 1, \ldots, N$ do:
1: **Step 1:** Draw $Z^{(n)} \sim g$ independently of all other draws.
2: **Step 2:** Set $X^{(n)} := Z^{(n)}$ with probability $\frac{f(Z^{(n)})}{Mg(Z^{(n)})}$. Otherwise, go to step 1.
Output: Sample $\{X^{(n)}\}_{n=1}^{N} \overset{\text{i.i.d.}}{\sim} f$.

For a convenient implementation of steps 1 and 2, we may proceed as follows:

1: **Step 1:** Draw $Z^{(n)} \sim g$ and $U^{(n)} \sim \text{Unif}(0, 1)$ independently.
2: **Step 2:** Accept $X^{(n)} = Z^{(n)}$ if $U^{(n)} \leq \frac{f(Z^{(n)})}{Mg(Z^{(n)})}$. Otherwise, go to step 1.

Pros and Cons: Rejection sampling can be used when f and g are only known up to a normalizing constant (this will be the content of an exercise in a forthcoming version of these notes). The method is very versatile and can be applied in general state space E, beyond the Euclidean setting considered here. Potential disadvantages include that (i) samples that are not accepted are thrown away; (ii) the time to obtain each sample is random; and (iii) the method suffers from the curse of dimension, as will be discussed below.

The next result shows that the rejection sampling algorithm produces draws distributed according to the target. It also characterizes the acceptance rate under the assumption that f and g are normalized; otherwise, the second and third assertions need to be adjusted.

Theorem 2.6 (Rejection Sampling Acceptance Rate).

1. *The draws $X^{(n)}$ generated by the rejection sampling algorithm have distribution f and are independent.*
2. *The probability that a draw $Z^{(n)}$ generated in step 1 is accepted in step 2 is $1/M$.*
3. *The algorithm will finish in finite time (with probability one). The expected number of iterations to accept each sample $X^{(n)}$ is M.*

Proof. First, the independence of the $X^{(n)}$ follows from the fact that the $Z^{(n)}$ are drawn independently. Let now $A \subset E$. Note that

$$\begin{aligned}\mathbb{P}(Z^{(n)} \in A \;\&\; Z^{(n)} \text{ is accepted}) &= \int_A \frac{f(x)}{Mg(x)} g(x)\,dx \\ &= \frac{\int_A f(x)\,dx}{M},\end{aligned}$$

where $\frac{f(x)}{Mg(x)}$ is the probability that $Z^{(n)}$ is accepted given that $Z^{(n)} = x$ and g is the density of $Z^{(n)}$. Setting $A = E$ and noting that f integrates to 1 gives

$$\mathbb{P}(Z^{(n)} \text{ is accepted}) = \frac{1}{M},$$

proving the second claim in the statement. Now,

$$\begin{aligned}\mathbb{P}(X^{(n)} \in A) = \mathbb{P}(Z^{(n)} \in A | Z^{(n)} \text{ is accepted}) &= \frac{\mathbb{P}(Z^{(n)} \in A \;\&\; Z^{(n)} \text{ is accepted})}{\mathbb{P}(Z^{(n)} \text{ is accepted})} \\ &= \frac{\frac{\int_A f(x)\,dx}{M}}{\frac{1}{M}} = \int_A f(x)\,dx,\end{aligned}$$

which proves the first claim. The third claim follows by noting that M is the expected value of a geometric random variable with parameter $1/M$. □

Choice of Upper-Bound Theorem 2.6 shows that if f and g are p.d.f.s such that $f(x) \leq Mg(x)$ for every $x \in E$, then Algorithm 2.3 has acceptance rate $1/M$. Therefore, given f and g, M should be chosen as small as possible while still satisfying the requirement that $f(x) \leq Mg(x)$ for every $x \in E$. This way the acceptance rate is maximized. Note that it always holds that $M \geq 1$, since $1 = \int_E f(x)\,dx \leq M \int_E g(x)\,dx = M$.

Choice of Proposal We want our proposal to be (1) easy to sample from; and (2) close to the target, so that we can find a small constant $M \geq 1$ satisfying $f(x) \leq Mg(x)$ for every $x \in E$. These two goals are often in conflict with each other, and a compromise must be found. In the extreme case where $f = g$ one could choose $M = 1$ (lowest possible bound), which is clearly not optimal if f is extremely expensive (or impossible) to sample from directly! A common strategy is to look for proposals in some parametric family $\{g_\theta, \theta \in \Theta\}$. If for each $\theta \in \Theta$ we can find M_θ such that $f(x) \leq M_\theta g_\theta(x)$ for every $x \in E$, then it is recommended to use as proposal the density with parameter $\theta^\star$ that minimizes M_θ over $\theta \in \Theta$.

Remark 2.7. *Note that if $E \subset \mathbb{R}^d$ is unbounded and $f(x) \leq Mg(x)$ for every $x \in E$, then g necessarily has fatter tails than f. Several Monte Carlo methods (e.g. importance sampling in Chapter 3) share this principle that the proposal should have fatter tails than the target.*

Envelope Rejection Sampling If evaluating the target density f is expensive, delaying the acceptance can improve the efficiency. Specifically, let ℓ be a function that is cheap to evaluate and such that $\ell(x) \leq f(x)$ for all $x \in E$. Then, we can modify the rejection sampling algorithm by delaying acceptance as follows:

1: **Step 1:** Sample $Z^{(n)} \sim g$, $U^{(n)} \sim \text{Unif}(0,1)$.
2: **Step 2:** Accept $X^{(n)} = Z^{(n)}$ if $U^{(n)} \leq \frac{\ell(Z^{(n)})}{Mg(Z^{(n)})}$. Otherwise, go to step 3.
3: **Step 3:** Accept $X^{(n)} = Z^{(n)}$ if $U^{(n)} \leq \frac{f(Z^{(n)})}{Mg(Z^{(n)})}$. Otherwise, go to step 1.

Rejection Sampling and the Curse of Dimension Suppose that, for a given target f and proposal g, we have found a constant M with $f(x) \leq Mg(x)$ for every $x \in E = \mathbb{R}$. We know that the probability of accepting a sample from g in the rejection sampling algorithm is $\frac{1}{M}$. Now consider a d-dimensional i.i.d. setting, where f_d and g_d are p.d.f.s on $E = \mathbb{R}^d$ defined by

$$f_d(x) := f(x_1) \times \cdots \times f(x_d), \quad x = (x_1, \ldots, x_d) \in \mathbb{R}^d,$$
$$g_d(x) := g(x_1) \times \cdots \times g(x_d), \quad x = (x_1, \ldots, x_d) \in \mathbb{R}^d.$$

To implement rejection sampling with target f_d and proposal g_d we can only guarantee that, for every $x \in \mathbb{R}^d$, the following upper bound holds:

$$\frac{f_d(x)}{g_d(x)} = \frac{f(x_1) \times \cdots \times f(x_d)}{g(x_1) \times \cdots \times g(x_d)} \leq M^d.$$

Thus, Theorem 2.6 shows that the probability of accepting each sample from g_d decreases exponentially with d. In other words, the expected number of iterations needed to obtain one sample grows exponentially with d.

2.2.2 ▪ Approximate Bayesian Computation

In Bayesian statistics, inference on a parameter θ given data y is based on the posterior distribution $f(\theta|y)$. The posterior is obtained by combining a likelihood model $f(y|\theta)$ and a prior density $f(\theta)$ using Bayes' formula:

$$f(\theta|y) = \frac{1}{c} f(y|\theta) f(\theta),$$

where c is a normalizing constant so that $f(\theta|y)$ integrates to 1. Approximate Bayesian Computation (ABC) refers to a family of methods designed to obtain *approximate* posterior samples when the likelihood function is costly to evaluate but easy to sample from. Thus, ABC is useful in applications where it is possible to simulate data from given model parameters, but computing the likelihood is intractable. ABC was first introduced in population genetics, where one is interested in recovering the history of a population from genetic data. The history is represented by a genealogy tree, which is easy to sample from given parameters such as mutation and growth rates, but the likelihood of the corresponding model is intractable. Likelihood-free methods that avoid evaluating the likelihood are important in many applications beyond population genetics, including epidemiology and cosmology.

The key idea behind ABC is to replace the evaluation of likelihoods with a comparison between sampled and observed data. This comparison is based on a choice of distance and tolerance. In its original, most basic form, ABC is the following accept/reject method.

Algorithm 2.4 ABC Rejection Sampler

Input: Prior $f(\theta)$, likelihood $f(y|\theta)$, distance d, tolerance ε, data y_o, sample size N.
For $n = 1, \ldots, N$ do:
1: **Step 1:** Sample $\theta^{(n)} \sim f(\theta)$ from the prior.
2: **Step 2:** Sample data $y^{(n)} \sim f(y|\theta^{(n)})$ from the likelihood.
3: If $d(y_o, y^{(n)}) \leq \varepsilon$, accept $\theta^{(n)}$. Otherwise, go to step 1.
Output: $\{\theta^{(n)}\}_{n=1}^{N} \overset{\text{i.i.d.}}{\sim} f(\theta|d(y, y_{\text{sim}}) \leq \varepsilon)$, where $y_{\text{sim}} \sim f(y)$.

Pros and Cons: ABC is useful when the likelihood is easy to sample from but expensive to evaluate. For some Bayesian inference problems, ABC is virtually the only implementable method. Some caveats of ABC include that (i) it only produces approximate posterior samples; (ii) theoretical developments are scarce; and (iii) the tolerance and the distance are often chosen heuristically, and the results are sensitive to these choices.

Choice of Distance It is often impractical to define a distance $d(y_o, y^{(n)})$ between the full data-sets. Instead, it is preferable to use lower-dimensional sufficient summary statistics $S(y_o)$ and $S(y^{(n)})$ of the data-sets and define

$$d(y_o, y^{(n)}) = \tilde{d}\big(S(y_o), S(y^{(n)})\big),$$

where $\tilde{d}$ is a distance function defined on the summary statistic space.

Choice of Tolerance In choosing ε, there is a trade-off between computational burden and approximation of the posterior. Smaller ε decreases the acceptance probability, but leads to samples whose distribution is closer to the posterior. This trade-off will be illustrated in the following example.

Example 2.8 (ABC: Beta-Binomial Bayesian Model). Consider a random experiment which involves n_o flips of a coin with unknown probability of heads p. Letting y denote the number of heads observed out of the n_o flips, we have a Binomial(n_o, p) likelihood function

$$f(y|p) = \binom{n_o}{y} p^y (1-p)^{n_o - y}.$$

Taking a Beta(α, β) prior for p, with p.d.f. $f(p) \propto p^{\alpha-1}(1-p)^{\beta-1}$, $p \in (0,1)$, leads to a posterior distribution

$$\begin{aligned} f(p|y) &\propto f(y|p) f(p) \\ &\propto p^{\alpha+y-1}(1-p)^{n_o+\beta-y-1}, \qquad 0 < p < 1. \end{aligned}$$

That is, the posterior of p given y is a Beta$(\alpha + y, n_o + \beta - y)$ distribution. For the purpose of this example, we choose $\alpha = \beta = 1$, which corresponds to a Unif(0,1) prior.

Next, for the sake of the example, assume that we do not know the true posterior distribution of p. We will approximate it using ABC. To that end, we simulate the model using a sample size of $n_o = 500$ with $p = 0.4$ and record the number y_o of observed successes. We seek to approximate the posterior of p given y_o. We repeatedly sample values $p^{(n)}$ from the prior distribution Beta$(1, 1)$, and simulate the number of successes $y^{(n)} \sim f(\cdot|p^{(n)})$. We next define a distance function,

$$d(y_o, y^{(n)}) := \frac{1}{n_o}|y^{(n)} - y_o|.$$

Using three tolerance values $\varepsilon = 0.01, 0.02, 0.05$, we accept the sample $p^{(n)}$ if $d(y^{(n)}, y_o) < \varepsilon$. For each choice of tolerance, we continue this procedure until we obtain $N = 1000$ accepted samples. Finally, for each tolerance, we plot a histogram of the accepted samples against the true posterior distribution, which is Beta$(1 + y_o, n_o + 1 - y_o)$. Figure 2.4 shows the results, along with the number of samples generated in each case to reach $N = 1000$ accepted samples. □

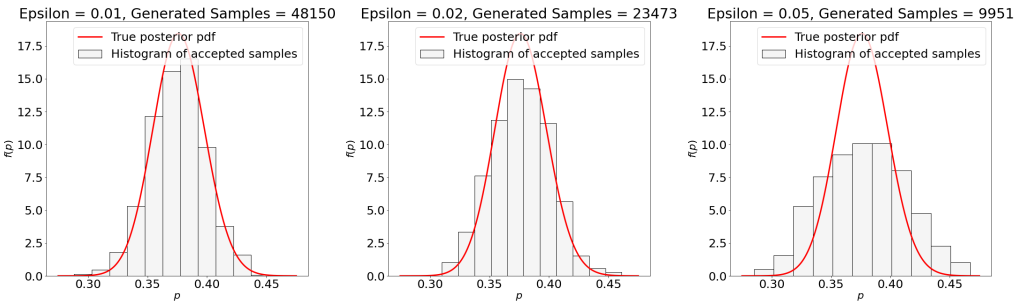


Figure 2.4. *ABC for a Beta-binominal model as described in Example 2.8. The results illustrate the trade-off between cost and accuracy in the choice of tolerance.*

2.3 ▪ Discussion and Bibliography

In this chapter, we took as our starting point a random number generator to produce uniform draws. The study of random number generators is a rich subject in and on itself; we refer to [83, 127, 136] for in-depth surveys of existing techniques and philosophical discussions on the use of deterministic algorithms for random simulation.

Transformation methods and accept/reject methods for non-uniform sampling are reviewed in many textbooks, see e.g. [56, 192]. The first algorithmic use of rejection sampling is often credited to Von Neumann [231] although the idea is already underpinning Buffon's needle experiment. The paper [231] also considers inversion from a uniform proposal distribution. Another early example of a transformation method is the Box-Muller algorithm [33] to sample from a normal distribution. The KR rearrangement was introduced in [126, 203]. We refer to [7] for further background on disintegration of measures. More recently, [157] investigates the use of transport maps for sampling. Triangular transport maps also form the building blocks of many complex architectures for *normalizing flows* in machine learning [180].

We refer to [226] for an accessible introduction to ABC. The main ideas behind ABC can be traced back at least to [204], but it was not until the late 90s that ABC methods received their name and gained popularity through a series of influential works, including [223, 187, 18]. The ABC rejection sampler in Algorithm 2.3 was introduced in [187]. Implementations of ABC using Markov chain Monte Carlo algorithms and particle filters were introduced in [155, 213]. ABC originates in the population genetics literature, where models are inherently complex and likelihoods are usually prohibitively costly to evaluate or impossible to compute. ABC techniques are also widely used in genetics [234, 212, 75], epidemiology [222], human demographic history [104], phylogeography [19], and many other applications. From a theoretical viewpoint, an influential paper is [237], which shows that ABC gives exact inference for a wrong model. ABC methods for Bayesian model choice have been extensively investigated, see for instance [194] and references therein. An active area of research relies on conditional density estimation for likelihood-free inference, see [179] and also [181]. These techniques have the advantage of

not relying on heuristically chosen tolerance levels.

Chapter 3

Monte Carlo Integration and Importance Sampling

This chapter introduces Monte Carlo methods to compute integrals of the form

$$\mathcal{I}_f[h] := \int_E h(x)f(x)\,dx = \mathbb{E}_{X\sim f}\big[h(X)\big],$$

where f is a p.d.f. supported on a set $E \subset \mathbb{R}^d$ and $h : E \to \mathbb{R}$ is a real-valued *test function*. For instance, taking $h(x) = \mathbf{1}_A(x)$ the indicator function of a subset $A \subset E$, we can then compute the probability $\mathbb{P}_{X\sim f}(X \in A)$. The methods in this chapter also apply to discrete target distributions and to vector-valued test functions, and are thus useful to compute moments of discrete or continuous target distributions.

The main idea behind Monte Carlo integration is to compute $\mathcal{I}_f[h]$ using a random sample. We will consider two strategies:

1. *Classical Monte Carlo integration* relies on a sample $X^{(1)}, \dots, X^{(N)} \overset{\text{i.i.d.}}{\sim} f$ from the target distribution to define an estimator

$$\mathcal{I}_f^{\text{MC}}[h] := \frac{1}{N}\sum_{n=1}^{N} h(X^{(n)}) \approx \mathcal{I}_f[h].$$

2. *Importance sampling* relies on a sample $X^{(1)}, \dots, X^{(N)} \overset{\text{i.i.d.}}{\sim} g$ from a proposal distribution $g \neq f$ to define an estimator

$$\mathcal{I}_f^{\text{IS}}[h] := \frac{1}{N}\sum_{n=1}^{N} w(X^{(n)})h(X^{(n)}) \approx \mathcal{I}_f[h]. \tag{3.1}$$

   Each draw $X^{(n)} \sim g$ is given an *importance weight* $w(X^{(n)}) := f(X^{(n)})/g(X^{(n)})$ determined by the ratio between target and proposal.

We will study classical Monte Carlo in Section 3.1 and importance sampling in Section 3.2. There are two main motivations to use importance sampling: (i) it can reduce the variance of classical Monte Carlo; and (ii) it enables Monte Carlo integration in cases where sampling from f is intractable. In addition to the importance sampling estimator in (3.1), we will introduce an *autonormalized* estimator that is applicable when the target and proposal distributions can only be evaluated up to a normalizing constant. Section 3.3 discusses other variance reduction techniques and Section 3.4 closes with bibliographical remarks.

3.1 ▪ Classical Monte Carlo

Classical Monte Carlo integration presupposes that we can obtain N independent draws from the target distribution. Then, the integral of interest $\mathcal{I}_f[h]$ is approximated by the average of the values of h at those draws. We refer to the method, summarized in Algorithm 3.1 below, as *classical* Monte Carlo to distinguish it from other Monte Carlo methods. In the literature it is often simply called Monte Carlo, standard Monte Carlo, or vanilla Monte Carlo.

Algorithm 3.1 Classical Monte Carlo Integration

Input: Target distribution f, test function h, sample size N.

1: Sample $X^{(1)}, \dots, X^{(N)} \overset{\text{i.i.d.}}{\sim} f$.

Output: Monte Carlo estimator $\mathcal{I}_f^{\text{MC}}[h] := \frac{1}{N}\sum_{n=1}^{N} h(X^{(n)}) \approx \mathcal{I}_f[h]$.

The following result shows that the classical Monte Carlo estimator is unbiased and that its mean squared error (which agrees with its variance) is of order $1/N$.

Theorem 3.1 (Classical Monte Carlo Error). *The following holds:*

1. $\mathbb{E}\big[\mathcal{I}_f^{\text{MC}}[h]\big] = \mathcal{I}_f[h]$;
2. $\mathbb{V}\big[\mathcal{I}_f^{\text{MC}}[h]\big] = \frac{1}{N}\mathbb{V}_{X\sim f}\big[h(X)\big]$.

Proof. For the first claim, linearity of expectation and the fact that $X^{(n)} \sim f$ give

$$\mathbb{E}\big[\mathcal{I}_f^{\text{MC}}[h]\big] = \mathbb{E}\left[\frac{1}{N}\sum_{n=1}^{N} h\big(X^{(n)}\big)\right] = \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}\big[h\big(X^{(n)}\big)\big] = \mathbb{E}_{X\sim f}\big[h(X)\big] = \mathcal{I}_f[h].$$

For the second claim note that since $X^{(n)} \overset{\text{i.i.d.}}{\sim} f$,

$$\mathbb{V}\big[\mathcal{I}_f^{\text{MC}}[h]\big] = \frac{1}{N^2}\sum_{n=1}^{N}\mathbb{V}\big[h\big(X^{(n)}\big)\big] = \frac{1}{N^2}N\mathbb{V}_{X\sim f}\big[h(X)\big] = \frac{1}{N}\mathbb{V}_{X\sim f}\big[h(X)\big]. \qquad \square$$

Theorem 3.1 can be used to construct confidence intervals for $\mathcal{I}_f^{\text{MC}}[h]$ as an estimator of $\mathcal{I}_f[h]$. To do so, note that by the central limit theorem, as $N \to \infty$,

$$\sqrt{N}\,\frac{\mathcal{I}_f^{\text{MC}}[h] - \mathcal{I}_f[h]}{\sqrt{\mathbb{V}_{X\sim f}[h(X)]}} \Rightarrow \mathcal{N}(0,1). \tag{3.2}$$

For $\alpha \in (0,1)$ define the z-score $z_{\alpha/2}$ by the requirement that $\mathbb{P}(-z_{\alpha/2} < Z < z_{\alpha/2}) = 1-\alpha$, where $Z \sim \mathcal{N}(0,1)$. Then, it follows from (3.2) that

$$\left(\mathcal{I}_f^{\text{MC}}[h] - z_{\alpha/2}\sqrt{\frac{\mathbb{V}_{X\sim f}[h(X)]}{N}}, \mathcal{I}_f^{\text{MC}}[h] + z_{\alpha/2}\sqrt{\frac{\mathbb{V}_{X\sim f}[h(X)]}{N}}\right)$$

is an approximate $1-\alpha$ confidence interval for $\mathcal{I}_f[h]$. The variance term can also be approximated by Monte Carlo:

$$\begin{aligned}\mathbb{V}_{X\sim f}\big[h(X)\big] &= \mathbb{E}\Big[\big(h(X) - \mathbb{E}_{X\sim f}\big[h(X)\big]\big)^2\Big] \\ &\approx \frac{1}{N}\sum_{n=1}^{N}\Big(h(X^{(n)}) - \mathcal{I}_f^{\text{MC}}[h]\Big)^2.\end{aligned} \tag{3.3}$$

The sample size N is typically large in Monte Carlo simulations, and hence large N asymptotics, needed to invoke the central limit theorem (3.2) and to ensure the convergence of the variance approximation (3.3), are well justified.

Pros and Cons: The standard error of the classical Monte Carlo estimator is of order $N^{-1/2}$, which implies a very slow rate of convergence compared to most deterministic quadrature methods: increasing by a factor of 100 the number of samples only reduces the expected error by a factor of 10. Another drawback of Monte Carlo methods is that the error can only be understood probabilistically, while for deterministic quadrature rules deterministic bounds can be derived. On the bright side, Theorem 3.1 does not make any assumption on the state space E (or its dimension) and the only implicit assumption on h is that $\mathbb{E}_{X\sim f}\left[h(X)^2\right] < \infty$. In contrast, classical error bounds for deterministic quadrature rules in $\mathbb{R}^d$ need the number N of function evaluations to scale as N^d as $d \to \infty$. Moreover, differentiability conditions on h often needed for deterministic methods play no role in Theorem 3.1. The insensitivity of classical Monte Carlo to dimension and roughness of f and h makes it appealing in cases where deterministic methods struggle. It is important to note, however, that the classical Monte Carlo method presupposes that sampling from f (without sampling error) is possible. When this is not true (which is often the case in applications) the error of Monte Carlo methods that rely on approximate target samples may be sensitive to the dimension of the state space.

3.2 ▪ Importance Sampling

Now we study how to approximate $\mathcal{I}_f[h]$ using samples from a proposal distribution $g \neq f$. There are two distinct motivations to sample from a proposal distribution g, rather than from the target density f.

Motivation 1: Variance Reduction If the proposal distribution is cleverly chosen, one can obtain an estimator with a relatively lower variance. From a historical perspective, variance reduction was the original motivation for introducing importance sampling; in particular, the design of proposal distributions for the simulation of rare events has been extensively researched. A natural question in this area is how best to choose the proposal distribution in order to compute the probability of a rare event under the target. In such a context, the test function h would be the indicator of a small-probability set. □

Motivation 2: Intractability of Target In some applications, sampling directly from f may not be possible, but one may have access to samples from a different distribution g. This scenario arises for instance in sequential signal estimation, where prior samples obtained by a stochastic dynamics model are used to compute expectations with respect to a posterior distribution which incorporates observed data. A natural question in this area is: given a proposal and a target, what is the worst-case error of importance sampling over a given class of test functions? □

In its simplest form, importance sampling is based on the observation that if f, g and h have compatible supports, then

$$\mathcal{I}_f[h] = \int_E h(x)f(x)\,dx = \int_E h(x)\frac{f(x)}{g(x)}g(x)\,dx = \mathcal{I}_g\left[h\frac{f}{g}\right].$$

This identity suggests that $\mathcal{I}_f[h]$ may be approximated using classical Monte Carlo integration by sampling the density g and considering hf/g as the test function:

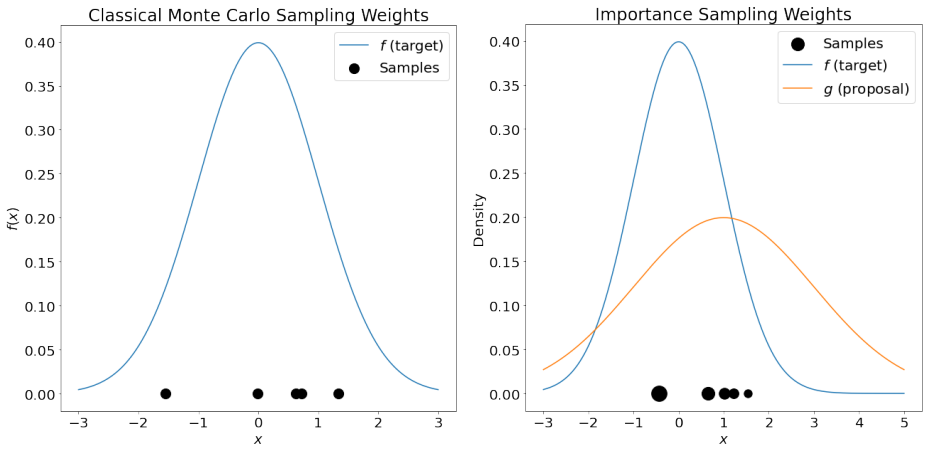


Figure 3.1. *In classical Monte Carlo (left), samples from f are given equal weights. In importance sampling (right), samples from g are given weights proportional to f/g, represented by the radii of the dots.*

Algorithm 3.2 Importance Sampling

Input: Target distribution f, proposal distribution g, test function h, sample size N.

1: Sample $X^{(1)}, \ldots, X^{(N)} \overset{\text{i.i.d.}}{\sim} g$.

2: Set

$$w(X^{(n)}) := \frac{f(X^{(n)})}{g(X^{(n)})}.$$

Output: Estimator $\mathcal{I}_f^{\text{IS}}[h] := \frac{1}{N}\sum_{n=1}^{N} w(X^{(n)})h(X^{(n)}) \approx \mathcal{I}_f[h]$.

Note that the importance sampling estimator $\mathcal{I}_f^{\text{IS}}[h]$ depends on the choice of proposal g, but we do not include that dependence in our notation. In addition to studying the importance sampling estimator $\mathcal{I}_f^{\text{IS}}[h]$, we will study an autonormalized implementation motivated by the two following drawbacks of Algorithm 3.2.

First, note that in order to compute the weights

$$w(X^{(n)}) := \frac{f(X^{(n)})}{g(X^{(n)})}$$

we need to be able to evaluate the ratio f/g, which may in practice involve evaluating both the target and the proposal with their appropriate normalizing constants.

Second, note that

$$\mathcal{I}_g[w] = \int w(x)g(x)\,dx = \int \frac{f(x)}{g(x)}g(x)\,dx = 1,$$

but in general

$$\frac{1}{N}\sum_{n=1}^{N} w(X^{(n)}) \neq 1.$$

In other words, importance sampling does not estimate exactly the constant test function $h \equiv 1$. These two observations motivate the use of *autonormalized* importance sampling (AIS).

Algorithm 3.3 Autonormalized Importance Sampling

Input: Target f, proposal g, test function h, sample size N.

1: Sample $X^{(1)}, \dots, X^{(N)} \overset{\text{i.i.d.}}{\sim} g$.

2: Set

$$w^*(X^{(n)}) := \frac{w(X^{(n)})}{\sum_{l=1}^{N} w(X^{(l)})}.$$

Output: Autonormalized estimator $\mathcal{I}_f^{\text{AIS}}[h] := \sum_{n=1}^{N} w^*(X^{(n)}) h(X^{(n)}) \approx \mathcal{I}_f[h]$.

The quantities $w^*(X^{(n)})$ are known as *normalized* weights because they add up to 1. Notice that w appears in the numerator and the denominator of w^*, and so w^* can be evaluated as long as $w(x) = \frac{f(x)}{g(x)}$ is known up to a multiplicative constant. Thus, autonormalized importance sampling can be implemented as long as f and g are known up to a normalizing constant, which greatly extends the applicability of the method. As we will see, the advantages afforded by the autonormalized estimator come at the cost of introducing a bias.

3.2.1 ▪ Error Bounds for Importance Sampling

In this subsection we analyze the importance sampling estimator $\mathcal{I}_f^{\text{IS}}[h]$. The presentation is focused on the variance reduction motivation for importance sampling; thus we aim at understanding the error for a given test function, and we will study the choice of proposal distribution to reduce the variance of the Monte Carlo estimator.

The following result is parallel to Theorem 3.1. As in classical Monte Carlo, the importance sampling estimator is unbiased and its mean squared error (which agrees with its variance) is of order $1/N$. Now, however, the mean squared error depends on the proposal distribution g.

Theorem 3.2 (Importance Sampling Error). *The following holds:*

1. $\mathbb{E}\big[\mathcal{I}_f^{\text{IS}}[h]\big] = \mathcal{I}_f[h]$;

2. $\mathbb{V}\big[\mathcal{I}_f^{\text{IS}}[h]\big] = \frac{1}{N}\mathbb{V}_{X\sim g}\big[h(X)w(X)\big]$.

Proof. It is a corollary of Theorem 3.1 since $\mathcal{I}_f[h] = \mathcal{I}_g[hw]$ and $\mathcal{I}_f^{\text{IS}}[h] = \mathcal{I}_g^{\text{MC}}[hw]$. □

Theorem 3.2 suggests that in order to approximate $\mathcal{I}_f[h]$ for given target f and test function h one should choose the proposal g that minimizes

$$\mathbb{V}_{X\sim g}[h(X)w(X)].$$

The next result is interesting for historical reasons and to develop intuition. As will be discussed below, its practical significance is rather limited.

Theorem 3.3 (Importance Sampling Optimal Proposal). *The proposal g that minimizes the variance of $\mathcal{I}_f^{\text{IS}}[h]$ is*

$$g^*(x) = \frac{|h(x)|f(x)}{\int_E |h(t)|f(t)dt}.$$

Proof. We have that

$$N\mathbb{V}\big[\mathcal{I}_f^{\text{IS}}[h]\big] = \mathbb{V}_g\left[h(X)\frac{f(X)}{g(X)}\right] = \mathbb{E}_g\left[\Big(h(X)\frac{f(X)}{g(X)}\Big)^2\right] - \mathbb{E}_g\left[h(X)\frac{f(X)}{g(X)}\right]^2.$$

Since $\mathbb{E}_g\left[h(X)\frac{f(X)}{g(X)}\right]^2 = \mathcal{I}_f[h]^2$ does not depend on g, we only need to minimize the first term in the right-hand side. Note that:
(i) Plugging $g = g^*$ into said term gives

$$\mathbb{E}_{g^*}\left[\Big(h(X)\frac{f(X)}{g^*(X)}\Big)^2\right] = \int \frac{h(x)^2 f(x)^2}{g^*(x)}\,dx = \int \frac{h(x)^2 f(x)^2}{|h(x)|f(x)}\,dx \int |h(t)|f(t)\,dt = \Big(\int |h(x)|f(x)\,dx\Big)^2.$$

(ii) Applying Jensen's inequality we have that, for any g,

$$\mathbb{E}_g\left[\Big(\frac{h(X)f(X)}{g(X)}\Big)^2\right] \geq \mathbb{E}_g\left[\frac{|h(X)|f(X)}{g(X)}\right]^2 = \Big(\int |h(x)|f(x)\,dx\Big)^2.$$

Combining (i) and (ii) gives the result. □

Theorem 3.3 is of little practical relevance: in order to use g^* as a proposal we need to be able to evaluate g^*, which involves computing an integral similar to the one we originally wanted to compute! In fact, if h is positive, then the denominator in the definition of g^* equals $\mathcal{I}_f[h]$; in that case, the proof of Theorem 3.3 implies that $\mathcal{I}_f^{\text{IS}}[h]$ has zero variance, and a direct calculation using the definition of $\mathcal{I}_f^{\text{IS}}[h]$ shows that $\mathcal{I}_f^{\text{IS}}[h] = \mathcal{I}_f[h]$. Despite the limited practical relevance of Theorem 3.3, a useful takeaway is that importance sampling can be super-efficient: it is possible to obtain a better (lower) variance estimator using samples from a proposal distribution than from the target. To further demonstrate this point, note that when $g = g^*$,

$$N\mathbb{V}\big[\mathcal{I}_f^{\text{IS}}[h]\big] = \mathbb{E}_f\big[|h(X)|\big]^2 - \mathcal{I}_f[h]^2 \leq \mathbb{E}_f\big[h(X)^2\big] - \mathbb{E}_f\big[h(X)\big]^2 = \mathbb{V}_f\big[h(X)\big] = N\mathbb{V}\big[\mathcal{I}_f^{\text{MC}}[h]\big].$$

Another useful takeaway is that samples from the proposal should concentrate on regions where $|h|$ is large, as illustrated in the following example.

Example 3.4 (Computing a Gaussian Tail Probability). Let $p = \mathbb{P}(X > 4)$ where $X \sim \mathcal{N}(0,1)$ with corresponding p.d.f. f. Using classical Monte Carlo we can approximate p by

$$\widehat{p} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{1}_{\{X^{(n)}>4\}}, \qquad X^{(1)}, \ldots, X^{(N)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1).$$

However, N would have to be very large to get an estimate that differs from 0. An alternative is to generate a sample of i.i.d. $\text{Exponential}(1)$ random variables translated to the right by 4, with corresponding p.d.f. g. The weights given to the samples would be determined by the weight function

$$w(x) = \frac{f(x)}{g(x)} = \frac{1}{\sqrt{2\pi}}\exp\Big(-\frac{1}{2}x^2 + (x-4)\Big).$$

Figure 3.2 shows an experiment where sampling from g rather than f is advantageous. □

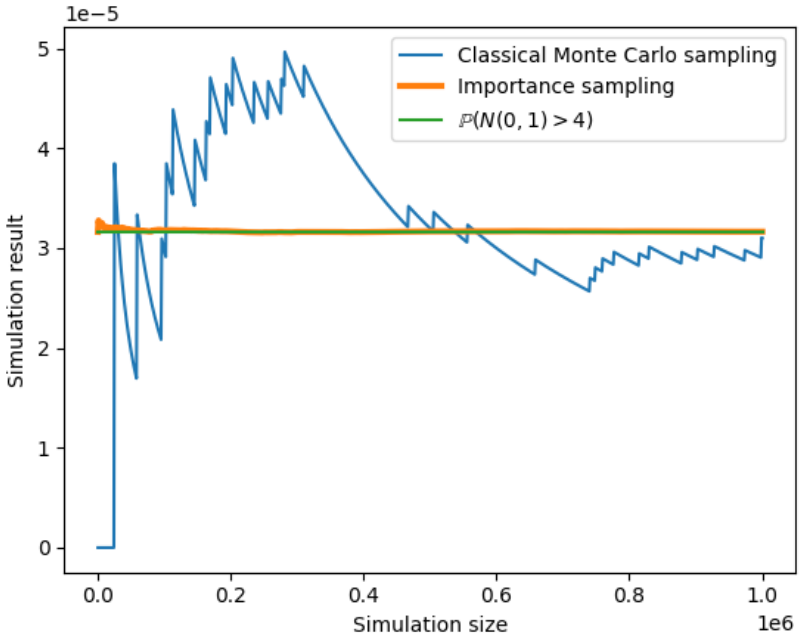


Figure 3.2. *Approximation of $p = \mathbb{P}(X > 4)$ with $X \sim \mathcal{N}(0,1)$ using classical Monte Carlo and importance sampling with a shifted exponential. The green line shows the true value, computed with Python. The approximation with importance sampling is accurate and stable with $N = 10^4$ samples, while classical Monte Carlo requires around $N = 10^6$ samples to reach a similar level of accuracy.*

3.2.2 ▪ Error Bounds for Autonormalized Importance Sampling

Recall that autonormalized importance sampling can be used as long as the ratio f/g can be evaluated up to a multiplicative constant. Here we analyze AIS in the following setting:

- Target $f(x) = \tilde{f}(x)/Z$, where f is a normalized distribution and $\tilde{f}$ is unnormalized.
- Normalized proposal distribution $g(x)$.

Our analysis is motivated by applications where sampling f directly may not be possible. We are interested in estimating $\mathcal{I}_f[h]$ for many different test functions h or for a test function h that is unknown before designing the algorithm. Thus, we will be concerned with bounding the *worst-case error* over a *family* of test functions. We will work with the family of bounded functions, but similar results can be obtained for other classes of test functions.

Autonormalized importance sampling is based on writing $\mathcal{I}_f[h]$ as a ratio of expectations with respect to g and approximating both expectations with classical Monte Carlo. Precisely,

$$\mathcal{I}_f[h] = \frac{\mathcal{I}_g[h(X)\tilde{w}(X)]}{\mathcal{I}_g[\tilde{w}(X)]} \approx \frac{\mathcal{I}_g^{\text{MC}}[h\tilde{w}]}{\mathcal{I}_g^{\text{MC}}[\tilde{w}]} = \mathcal{I}_f^{\text{AIS}}[h],$$

where $\tilde{w}(x) = \tilde{f}(x)/g(x)$. Note that the denominator $\mathcal{I}_g^{\text{MC}}[\tilde{w}]$ is an approximation of the unknown normalizing constant Z of the density f. The next result is parallel to Theorems 3.1 and

3.2. Note that, in contrast to classical Monte Carlo and importance sampling, autonormalized importance sampling gives a *biased* estimator, since the ratio of two unbiased estimators is in general biased.

Theorem 3.5 (Autonormalized Importance Sampling Error). *Let $h : E \to \mathbb{R}$ with $|h|_\infty := \sup_{x\in E} |h(x)| \leq 1$. The following holds:*

1. $\left|\mathbb{E}\left[\mathcal{I}_f^{\text{AIS}}[h] - \mathcal{I}_f[h]\right]\right| \leq \frac{2}{N}\frac{\mathcal{I}_g[\tilde{w}^2]}{\mathcal{I}_g[\tilde{w}]^2}$;
2. $\mathbb{E}\left[\left(\mathcal{I}_f^{\text{AIS}}[h] - \mathcal{I}_f[h]\right)^2\right] \leq \frac{4}{N}\frac{\mathcal{I}_g[\tilde{w}^2]}{\mathcal{I}_g[\tilde{w}]^2}$.

Proof. Note that

$$\begin{aligned}\mathcal{I}_f^{\text{AIS}}[h] - \mathcal{I}_f[h] &= \mathcal{I}_f^{\text{AIS}}[h] - \frac{\mathcal{I}_g[\tilde{w}h]}{\mathcal{I}_g[\tilde{w}]} \\ &= \left(\mathcal{I}_g[\tilde{w}] - \mathcal{I}_g^{\text{MC}}[\tilde{w}]\right)\frac{\mathcal{I}_f^{\text{AIS}}[h]}{\mathcal{I}_g[\tilde{w}]} - \frac{\mathcal{I}_g[\tilde{w}h] - \mathcal{I}_g^{\text{MC}}[\tilde{w}h]}{\mathcal{I}_g[\tilde{w}]}.\end{aligned}$$

This identity will be used to show the bounds for the bias and the mean squared error. We start with the latter:

$$\begin{aligned}\mathbb{E}\left[(\mathcal{I}_f^{\text{AIS}}[h] - \mathcal{I}_f[h])^2\right] &\leq \frac{2}{\mathcal{I}_g[\tilde{w}]^2}\left(\mathbb{E}\left[(\mathcal{I}_f^{\text{AIS}}[h])^2(\mathcal{I}_g[\tilde{w}] - \mathcal{I}_g^{\text{MC}}[\tilde{w}])^2\right] + \mathbb{E}\left[(\mathcal{I}_g[\tilde{w}h] - \mathcal{I}_g^{\text{MC}}[\tilde{w}h])^2\right]\right) \\ &\leq \frac{2}{\mathcal{I}_g[\tilde{w}]^2 N}\left(\mathbb{V}_g[\tilde{w}] + \mathbb{V}_g[\tilde{w}h]\right) \\ &\leq \frac{4}{N}\frac{\mathcal{I}_g[\tilde{w}^2]}{\mathcal{I}_g[\tilde{w}]^2}.\end{aligned}$$

We have used (i) for any $a, b \in \mathbb{R}$, $(a-b)^2 \leq 2(a^2+b^2)$; (ii) Theorem 3.1 for MSE of classical Monte Carlo; (iii) that h is bounded by 1; and (iv) the bound $\mathbb{V}[\tilde{w}] \leq \mathbb{E}_g[\tilde{w}^2] = \mathcal{I}_g[\tilde{w}^2]$.

Now we prove the result for the bias:

$$\begin{aligned}\left|\mathbb{E}\left[\mathcal{I}_f^{\text{AIS}}[h] - \mathcal{I}_f[h]\right]\right| &= \frac{1}{\mathcal{I}_g[\tilde{w}]}\left|\mathbb{E}\left[(\mathcal{I}_f^{\text{AIS}}[h] - \mathcal{I}_f[h])(\mathcal{I}_g[\tilde{w}] - \mathcal{I}_g^{\text{MC}}[\tilde{w}])\right]\right| \\ &\leq \frac{1}{\mathcal{I}_g[\tilde{w}]}\left(\mathbb{E}\left[(\mathcal{I}_f^{\text{AIS}}[h] - \mathcal{I}_f[h])^2\right]\right)^{1/2}\left(\mathbb{E}\left[(\mathcal{I}_g[\tilde{w}] - \mathcal{I}_g^{\text{MC}}[\tilde{w}])^2\right]\right)^{1/2} \\ &\leq \frac{1}{\mathcal{I}_g[\tilde{w}]}\left(\frac{4}{N}\frac{\mathcal{I}_g[\tilde{w}^2]}{\mathcal{I}_g[\tilde{w}]^2}\right)^{1/2}\left(\frac{\mathcal{I}_g[\tilde{w}^2]}{N}\right)^{1/2} \\ &\leq \frac{2}{N}\frac{\mathcal{I}_g[\tilde{w}^2]}{\mathcal{I}_g[\tilde{w}]^2}.\end{aligned}$$

For the first equality we used that $\mathbb{E}\left[\mathcal{I}_g[\tilde{w}] - \mathcal{I}_g^{\text{MC}}[\tilde{w}]\right] = 0$, for the first inequality Cauchy-Schwartz, and for the second inequality the bound for the mean squared error. □

Remark 3.6. *The autonormalized importance sampling estimator is biased. Its bias can be bounded uniformly over the class of bounded test functions $h : E \to \mathbb{R}$ and it is of order $\frac{1}{N}$. The MSE of the autonormalized importance sampling estimator is of order $\frac{1}{N}$ as well. Hence, since*

$$\textit{MSE} = \textit{bias}^2 + \textit{variance},$$

it follows that for large N the MSE is dominated by the variance term.

Autonormalized Importance Sampling and Closeness Between Target and Proposal The upper bounds on the bias and MSE depend on

$$\zeta := \frac{\mathcal{I}_g[\tilde{w}^2]}{\mathcal{I}_g[\tilde{w}]^2}.$$

The value of ζ quantifies the variability on the weights. It holds that $\zeta \geq 1$ with equality only if $\tilde{w}$ is constant, i.e. if $f = g$. Larger values of ζ imply worse behavior of importance sampling over the class of bounded test functions. Thus, if no knowledge of the relationship between the test function h and the target is available, it is important to choose a proposal that is close to the target, so that ζ is small.

The quantity ζ is intimately related to the χ^2-divergence between the target and the proposal. The χ^2-divergence is a way to quantify the closeness between two densities, defined by

$$d_{\chi^2}(f\|g) := \int_E \left(\frac{f(x)}{g(x)} - 1\right)^2 g(x)\, dx.$$

Therefore, we have that $d_{\chi^2}(f\|g) = \zeta - 1$. Hence, Theorem 3.5 implies that for AIS to perform accurately over the class of bounded test functions, the χ^2-divergence between the target and the proposal needs to be small/moderate.

Effective Sample Size The effective sample size

$$\text{ESS} := \frac{1}{\sum_{n=1}^N w^*(X^{(n)})^2}, \qquad w^*(X^{(n)}) := \frac{\tilde{w}(X^{(n)})}{\sum_{n=1}^N \tilde{w}(X^{(n)})}$$

is widely used by practitioners as a diagnostic for the performance of (autonormalized) importance sampling. Note that

- ESS $= 1$ if there is $n \in \{1, \ldots, N\}$ such that $w^*(X^{(n)}) = 1$ (in which case all other normalized weights are zero).
- ESS $= N$ if $w^*(X^{(n)}) = \frac{1}{N}$ for all $n \in \{1, \ldots, N\}$.
- ESS $\in [1, N]$ for any configuration of weights.

The effective sample size ESS is intimately related to ζ and $d_{\chi^2}(f\|g)$. Indeed,

$$\frac{\text{ESS}}{N} = \frac{1}{N\sum w^*(X^{(n)})^2} = \frac{(\sum \tilde{w}(X^{(n)}))^2}{N\sum \tilde{w}(X^{(n)})^2} = \frac{\left(\frac{\sum \tilde{w}(X^{(n)})}{N}\right)^2}{\frac{\sum \tilde{w}(X^{(n)})^2}{N}} = \frac{\mathcal{I}_g^{\text{MC}}[\tilde{w}]^2}{\mathcal{I}_g^{\text{MC}}[\tilde{w}^2]},$$

and so

$$\frac{N}{\text{ESS}} = \frac{\mathcal{I}_g^{\text{MC}}[\tilde{w}^2]}{\mathcal{I}_g^{\text{MC}}[\tilde{w}]^2} \approx \zeta.$$

It is important to note, however, that ESS does not include any test function h in its definition. Therefore, the performance of (autonormalized) importance sampling with any given test function cannot be fully understood via the effective sample size.

Autonormalized Importance Sampling and Curse of Dimension Consider (as for rejection sampling) the i.i.d. setting

$$\tilde{f}_d(x) = \tilde{f}(x_1) \times \cdots \times \tilde{f}(x_d), \quad \tilde{f} : \mathbb{R} \longrightarrow \mathbb{R}, \quad x = (x_1, \ldots, x_d) \in \mathbb{R}^d,$$
$$g_d(x) = g(x_1) \times \cdots \times g(x_d), \qquad g : \mathbb{R} \longrightarrow \mathbb{R},$$

where the tildes stand for unnormalized densities. Then,

$$\frac{\mathcal{I}_{g_d}[\tilde{w}_d^2]}{\mathcal{I}_{g_d}[\tilde{w}_d]^2} = \left(\frac{\mathcal{I}_g[\tilde{w}^2]}{\mathcal{I}_g[\tilde{w}]^2}\right)^d.$$

This identity along with Theorem 3.5 suggest that in order to have the same accuracy in importance sampling with proposal g_d and target f_d across a sequence of problems with increasing d, one needs to increase the number of samples exponentially with d.

Pros and Cons: As opposed to classical Monte Carlo, (autonormalized) importance sampling can be implemented when f cannot be sampled from and, in contrast to rejection sampling, no samples are thrown away. Importance sampling can have smaller variance than classical Monte Carlo integration, but finding an adequate proposal that gives variance reduction can be challenging. Autonormalized importance sampling can have smaller variance than both importance sampling and classical Monte Carlo, but is biased. On the other hand, importance sampling requires evaluating the normalized target and proposal distributions, but is unbiased. If the target and the proposal are far apart (which is typically the case in high-dimensional problems) the weights have a large variance. Then, the effective sample size will be small, and there will be test functions for which importance sampling performs poorly.

3.3 ▪ Variance Reduction Techniques

We have seen in Theorem 3.3 that importance sampling with an appropriate choice of proposal distribution —adapted to both the target and the test function— can have smaller variance than classical Monte Carlo. In this section, we describe two other variance reduction techniques: antithetic sampling and control variates. The general problem that we consider is again the estimation of $\mathcal{I}_f[h] := \mathbb{E}_{X\sim f}[h(X)]$ by using N queries of the p.d.f. f. Recall that the classical Monte Carlo estimator $\mathcal{I}_f^{\text{MC}}[h] = \frac{1}{N}\sum_{n=1}^N h(X^{(n)})$ is unbiased and has variance σ^2/N with $\sigma^2 := \mathbb{V}_{X\sim f}[h(X)]$.

3.3.1 ▪ Antithetic Sampling

This method requires the target distribution to be symmetric with respect to some center c, so that $f(x) = f(2c - x)$ holds for all x. We let N be an even integer and draw $N/2$ samples from f; symmetry is used to define the remaining $N/2$ samples.

Algorithm 3.4 Antithetic Sampling

Input: Target f which is symmetric with respect to c, even sample size N.

1: Sample $X^{(1)}, \ldots, X^{(\frac{N}{2})} \overset{\text{i.i.d.}}{\sim} f$.

2: Compute symmetrized samples $\tilde{X}^{(n)} := 2c - X^{(n)}$, $n = 1, \ldots, N/2$.

Output: Antithetic estimator $\mathcal{I}_f^{\text{AS}}[h] := \frac{1}{N}\sum_{n\leq \frac{N}{2}} \left(h(X^{(n)}) + h(\tilde{X}^{(n)})\right) \approx \mathcal{I}_f[h]$.

The estimator $\mathcal{I}_f^{\text{AS}}[h]$ is unbiased and has variance

$$\begin{aligned}\mathbb{V}\left[\mathcal{I}_f^{\text{AS}}[h]\right] &= \frac{N/2}{N^2}\,\mathbb{V}\big[h(X)+h(\tilde{X})\big] \\ &= \frac{1}{2N}\Big(\mathbb{V}[h(X)]+\mathbb{V}[h(\tilde{X})]+2\,\mathrm{Cov}\big(h(X),h(\tilde{X})\big)\Big) \\ &= \frac{\sigma^2}{N}(1+\rho),\end{aligned}$$

where $\tilde{X} = 2c - X$, $\sigma^2 = \mathbb{V}[h(X)] = \mathbb{V}[h(\tilde{X})]$, and $\rho = \mathrm{Corr}\big(h(X),h(\tilde{X})\big)$. Note that when h is linear, we have $\rho = -1$ and hence zero variance. Indeed, for any affine function $h(X) = AX + b$, the estimator $\mathcal{I}_f^{\text{AS}}[h]$ is constant and agrees with the expectation of interest. In the general case where $-1 \le \rho \le 1$, we have $\mathbb{V}\left[\mathcal{I}_f^{\text{AS}}[h]\right] \le \frac{2\sigma^2}{N} = \mathbb{V}\left[\mathcal{I}_f^{\text{MC}}[h]\right]$, i.e. in the worst case antithetic estimation has twice the variance of classical Monte Carlo.

In order to analyze the variance for different choice of test functions, we decompose h into $h_o + h_e$, where

$$h_o(x) := \frac{h(x)-h(\tilde{x})}{2}, \qquad h_e(x) := \frac{h(x)+h(\tilde{x})}{2}$$

are odd and even components of h with respect to the density f [2]. Note that $\mathbb{E}[h_e] = \mathbb{E}[h]$ and $\mathbb{E}[h_o] = 0$. Furthermore, h_e and h_o are uncorrelated

$$\begin{aligned}\mathrm{Cov}\big(h_e(X),h_o(X)\big) &= \mathbb{E}\left[\left(\frac{h(X)+h(\tilde{X})}{2}-\mathbb{E}[h]\right)\left(\frac{h(X)-h(\tilde{X})}{2}\right)\right] \\ &= \frac{1}{4}\,\mathbb{E}\left[\Big(h(X)+h(\tilde{X})-2\,\mathbb{E}[h]\Big)\Big(h(X)-h(\tilde{X})\Big)\right] \\ &= \frac{1}{4}\,\mathbb{E}\left[h(X)^2-h(\tilde{X})^2-2\,\mathbb{E}[h]\big(h(X)-h(\tilde{X})\big)\right] \\ &= 0.\end{aligned}$$

As a result, $\sigma^2 = \sigma_e^2 + \sigma_o^2$, where $\sigma_e^2 := \mathbb{V}[h_e(X)]$ and $\sigma_o^2 := \mathbb{V}[h_o(X)]$. Rewriting the variance of our estimators, we have

$$\mathbb{V}\left[\mathcal{I}_f^{\text{MC}}[h]\right] = \frac{1}{N}\sigma^2 = \frac{\sigma_e^2+\sigma_o^2}{N}$$

and

$$\mathbb{V}\left[\mathcal{I}_f^{\text{AS}}[h]\right] = \frac{1}{2N}\,\mathbb{V}\big[h(X)+h(\tilde{X})\big] = \frac{1}{2N}\,\mathbb{V}\big[2h_e(X)\big] = \frac{2\sigma_e^2}{N}.$$

Observe that antithetic sampling eliminates σ_o^2 but amplifies σ_e^2. In particular, any affine function is "odd" with respect to a symmetric density, which explains again our previous findings. Generally, antithetic sampling performs well when the "odd" part of the function has smaller variance than the "even" part.

Example 3.7 (Integral Approximation with Antithetic Sampling). Suppose we want to approximate an integral

$$\int_0^1 h(x)\,dx = \mathbb{E}_{X\sim\text{Unif}(0,1)}[h(X)].$$

We can apply Algorithm 3.4 since the uniform distribution is symmetric with respect to the center $c = \frac{1}{2}$. For instance, we can first draw 50 samples $X^{(1)}, X^{(2)}, \ldots, X^{(50)}$ uniformly from $[0,1]$ and estimate the integral by $\frac{1}{100}\sum_{i=1}^{50}\big(h(X^{(i)})+h(1-X^{(i)})\big)$. □

[2] Recall that the definition of symmetry depends on the center of the target distribution.

3.3.2 ▪ Control Variates

If we have access to an approximation $\widehat{h}$ of the test function h along with its true mean $\mu := \mathbb{E}_{X\sim f}\big[\widehat{h}(X)\big]$ we can apply control variates.

Algorithm 3.5 Control Variates

Input: Target f, test function h and approximation $\widehat{h}$, $\mu := \mathbb{E}\big[\widehat{h}(X)\big]$, sample size N.

1: Sample $X^{(1)}, \dots, X^{(N)} \overset{\text{i.i.d.}}{\sim} f$.

Output: Estimator $\mathcal{I}_f^{\text{CV}}[h] := \mu + \frac{1}{N}\sum_{n\leq N}\big(h(X^{(n)}) - \widehat{h}(X^{(n)})\big) \approx \mathcal{I}_f[h]$.

Note that the trivial approximation $\widehat{h} = 0$ leads to the classical Monte Carlo estimator $\mathcal{I}_f^{\text{CV}}[h] \equiv \mathcal{I}_f^{\text{MC}}[h]$. On the other hand, the oracle approximation $\widehat{h} = h$ gives $\mathcal{I}_f^{\text{CV}}[h] = \mathcal{I}_f[h]$, which is unsurprising since control variates assumes access to the true mean of our approximation $\widehat{h} = h$. In general, $\mathcal{I}_f^{\text{CV}}[h]$ is an unbiased estimator with variance $\mathbb{V}\big[\mathcal{I}_f^{\text{CV}}[h]\big] = \frac{1}{N}\mathbb{V}\big[h(X) - \widehat{h}(X)\big]$. Therefore, the performance of control variate estimation is determined by the quality of the approximation $\widehat{h} \approx h$.

Example 3.8 (Integral Approximation with Control Variates). Suppose, as in Example 3.7, that we wish to approximate

$$\int_0^1 h(x)\,dx = \mathbb{E}_{X\sim\text{Unif}(0,1)}[h(X)],$$

where h is a smooth test function which does not admit a closed-form primitive. We can use a polynomial approximation $\widehat{h}$ (e.g. Taylor expansion) of h, compute $\mu = \int_0^1 \widehat{h}(x)\,dx$ exactly, and use Algorithm 3.5 by drawing uniform samples from the unit interval. □

3.4 ▪ Discussion and Bibliography

The book chapter [105] is a classic reference on the Monte Carlo method and variance reduction techniques. Modern textbooks include [14, 178, 90, 130, 145, 192, 95, 15, 173, 193]. Some of these books emphasize specific applications, such as finance [90, 233, 116], statistics [14, 192, 178], scientific computing [145], statistical physics [173], stochastic simulation [95], or computer vision, machine learning, and artificial intelligence [15]. For a textbook on quantum Monte Carlo methods with applications in chemistry, we refer to [99]. The textbook [193] introduces Monte Carlo methods through examples coded in R, while [130] includes numerous worked examples using MATLAB. Finally, [40] provides an accessible overview of Monte Carlo methods aimed at applied mathematicians.

In Section 3.1 we invoked the central limit theorem to derive an asymptotic confidence interval for classical Monte Carlo integration. Non-asymptotic analyses are also possible, and we refer to [95, Chapter 3] for further details. Empirical process theory provides another perspective on Monte Carlo methods, see e.g. [230, Chapter 8] for a gentle introduction.

The lecture notes [9] compare Monte Carlo and importance sampling through examples. Importance sampling was developed as a variance reduction technique in the early 1950's [121, 120]. Theorem 3.3, which shows how to optimally choose the proposal density for a given test function h and target density f, was already shown in [121]. For recent methodological developments, see [140, 177, 146, 221, 122]. The book [49] contains a modern presentation of importance sampling in a general framework. A review of importance sampling, from the

perspective of filtering and sequential importance resampling, can be found in [3]; the proofs in this chapter closely follow the presentation in that paper. For a review of adaptive importance sampling techniques, see [39].

The key role of the second moment of the weight function w has long been realized [144, 185], and it is known to be asymptotically linked to the effective sample size [128, 129, 144]. The question of how to choose a proposal distribution that leads to small value of said second moment has been widely studied, and we refer to [142] and references therein. Similar to [46], Theorem 3.5 quantifies the estimation error in terms of the χ^2-divergence between target and proposal; recent complementary analysis of importance sampling in [45] utilizes the Kullback-Leibler divergence. Necessary sample size results for importance sampling in terms of several divergences between target and proposal were established in [206, 208]. A review of useful distances between probability measures can be found in [86]. Effective sample size based on discrepancy measures are studied in [156]. The paper [71] overviews existing notions of effective sample size for importance sampling algorithms and proposes new ones.

Variance reduction techniques are covered in most standard textbooks on Monte Carlo methods, see e.g. [14, Chapter 5] and [192, Chapter 4]. The monograph [95] considers more advanced variance reduction techniques. In particular, we refer to [95, Part III] for variance reduction techniques for simulation of stochastic differential equations.

Chapter 4
Metropolis Hastings

This chapter introduces the powerful idea of using Markov chains for sampling. This idea underlies many Markov chain Monte Carlo (MCMC) algorithms studied in this and subsequent chapters. Here, we focus on the Metropolis Hastings framework: we combine a proposal Markov kernel with an accept/reject mechanism to obtain a Markov kernel that satisfies detailed balance with respect to the target distribution. We refer to Appendix A for background on Markov chains that is needed to understand the material in this chapter.

The Metropolis Hastings algorithm outputs draws $\{X^{(n)}\}_{n=1}^N$ that are correlated and only approximately distributed like the target. Similar to the Monte Carlo integration methods in Chapter 3, the draws $\{X^{(n)}\}_{n=1}^N$ can be used to approximate integrals of the form

$$\mathcal{I}_f[h] = \int_E h(x) f(x)\, dx,$$

where the target p.d.f. f is assumed to be supported on $E \subset \mathbb{R}^d$ and $h : E \to \mathbb{R}$ is a real-valued test function.[3] To that end, one can define an estimator

$$\mathcal{I}_f^{\text{MH}}[h] := \frac{1}{N} \sum_{n=1}^N h(X^{(n)}) \approx \mathcal{I}_f[h].$$

Standard Markov chain theory, reviewed in Appendix A, guarantees that if the chain $\{X^{(n)}\}_{n=1}^N$ is ergodic, then the estimator $\mathcal{I}_f^{\text{MH}}[h]$ is asymptotically unbiased (see Theorem A.32), and if the chain is geometrically ergodic, then it satisfies a central limit theorem (see Theorem A.34). As in classical Monte Carlo, the samples are given uniform weights $1/N$, but now they are not drawn from f, and they are not independent. Thus, the Metropolis Hastings algorithm is useful when the target distribution is intractable and cannot be sampled directly. Importance sampling sidesteps the intractability of the target by sampling from a proposal distribution and giving each draw an importance weight. In contrast, Metropolis Hastings sidesteps the intractability of the target by sampling from a Markov kernel and using an accept/reject mechanism. An advantage of this latter approach is that the draws receive uniform weights, avoiding the weight degeneracy of importance sampling in high dimension. Additionally, the choice of a proposal Markov kernel affords more flexibility than the choice of a proposal distribution.

This chapter is organized as follows. Section 4.1 introduces the Metropolis Hastings algorithm. Two choices of proposal Markov kernel are discussed in Section 4.2: the independence

[3] The methodology is also applicable for discrete target distributions f and vector-valued test functions.

sampler and the random walk Metropolis Hastings algorithm. Section 4.3 discusses implementation issues and convergence diagnostics. These diagnostics address an important caveat of the Metropolis Hastings algorithm: it is often difficult in practice to determine whether the Markov chain has converged, and, relatedly, whether our sample size N is large enough to meet a given error tolerance. Section 4.4 closes with bibliographical remarks.

4.1 ▪ Metropolis Hastings Algorithm

The Metropolis Hastings algorithm provides a flexible framework to define a Markov kernel p_{MH} that can be sampled from and leaves the target invariant. The idea is to leverage a probabilistic accept/reject mechanism to turn a user-chosen proposal Markov kernel q into a Markov kernel p_{MH} that satisfies detailed balance with respect to the target f. From the definition of Markov kernel, for each $x \in E$, $q(x, z)$ will represent the probability (or probability density in the continuous case) with which a move from x to z is proposed. Recall further that, for fixed x, $q(x, \cdot\,)$ is a p.m.f. if E is discrete and a p.d.f. if E is continuous.

The method is summarized in Algorithm 4.1. Given the n-th sample $X^{(n)}$, we first draw a proposed move $Z^* \sim q(X^{(n)}, \cdot\,)$. We accept the move, which means setting $X^{(n+1)} = Z^*$, with probability $a(X^{(n)}, Z^*)$. If the proposed draw Z^* is rejected, we set $X^{(n+1)} = X^{(n)}$. The probability of accepting a proposed move from $x \in E$ to $z \in E$ is defined to be

$$a(x, z) := \min\left\{1, \frac{f(z)}{f(x)} \frac{q(z, x)}{q(x, z)}\right\}.$$

We will show in Theorem 4.3 that this choice of acceptance probability turns the kernel q into a kernel p_{MH} that satisfies detailed balance with respect to f.

Algorithm 4.1 Metropolis Hastings Algorithm

Input: Target f, initial distribution π_0, proposal Markov kernel $q(x, z)$, sample size N.
Initial draw: Sample $X^{(0)} \sim \pi_0$.
Subsequent draws: For $n = 0, \ldots, N-1$ do:

1: **Proposal step:** Sample $Z^* \sim q(X^{(n)}, \cdot\,)$.
2: **Accept/reject step:** Update

$$X^{(n+1)} = \begin{cases} Z^* & \text{with probability } a(X^{(n)}, Z^*), \\ X^{(n)} & \text{with probability } 1 - a(X^{(n)}, Z^*). \end{cases}$$

Output: Sample $\{X^{(n)}\}_{n=1}^N$.

As previously mentioned, given a test function $h : E \to \mathbb{R}$, we can use the output $\{X^{(n)}\}_{n=1}^N$ of the Metropolis Hastings algorithm to approximate

$$\mathcal{I}_f[h] = \int_E h(x) f(x)\, dx \approx \frac{1}{N} \sum_{n=1}^N h(X^{(n)}) =: \mathcal{I}_f^{\text{MH}}[h].$$

Remark 4.1.

1. *To implement the Metropolis Hastings algorithm we need to be able to sample $q(x, \cdot\,)$ for $x \in E$ and evaluate the acceptance probability $a(x, z)$ for $x, z \in E$. A priori, the only necessary condition on the proposal is that*

$$E \subset \bigcup_{x \in E} \textit{support}\big(q(x, \cdot\,)\big).$$

Importantly, the target f appears in the acceptance probability only as a ratio, and therefore the algorithm can be implemented even if f is only known up to a normalizing constant.

2. *If $q(x,z) = q(z,x)$ for all $x, z \in E$, then $a(x,z) = \min\left\{1, \frac{f(z)}{f(x)}\right\}$. Moves to regions of higher target density are always accepted, while moves to regions of lower but non-zero target density are accepted with positive probability in order to ensure exploration of the state space E. If q is not symmetric, moves for which $q(z,x) \geq q(x,z)$ are favored, as they are more likely to be undone by chance.*

3. *The accept/reject step can be implemented as follows:*

   - *Draw $U^* \sim \text{Unif}(0,1)$ independently from Z^*.*
   - *Update*

$$X^{(n+1)} = \begin{cases} Z^* & \text{if} \quad U^* < a(X^{(n)}, Z^*), \\ X^{(n)} & \text{if} \quad a(X^{(n)}, Z^*) < U^*. \end{cases}$$

Pros and Cons: The Metropolis Hastings algorithm can be implemented without knowing the normalizing constant of the target, and is extremely flexible due to the freedom in the choice of proposal kernel q. It has important advantages over all the algorithms covered in Chapters 2 and 3. In contrast to rejection sampling, no samples are discarded; in contrast to importance sampling, MCMC samples are given equal weights, avoiding weight collapse; in contrast to ABC, in many cases MCMC is provably accurate in the large N limit. The Metropolis Hastings algorithm also has some caveats. First, it can be very hard (often impossible) to tell if the chain is close to stationarity. As opposed to rejection sampling and importance sampling, MCMC samples are typically positively correlated, which, as established in Lemma A.36, results in larger variance estimates than independent or negatively correlated samples. Additionally, it is often challenging to choose an appropriate proposal kernel, and the ergodic behavior of the algorithm depends on that choice. In particular, it is often necessary to tune the proposal by trial and error (the Gibbs sampler, studied in Chapter 5, is an exception). Finally, the Metropolis Hastings algorithm cannot be naturally parallelized.

The Metropolis Hastings algorithm implicitly defines a Markov kernel $p_{\text{MH}}(x,z)$ which gives the probability (or probability density) of finding the $(n+1)$-th sample at location $z \in E$ given that the n-th sample was at $x \in E$.

Lemma 4.2. *The Metropolis Hastings Markov kernel is given by*

$$p_{\text{MH}}(x,z) = q(x,z)a(x,z) + \delta_x(z)r(x),$$

where

$$r(x) = \begin{cases} \sum_{y \in E} q(x,z)\big(1 - a(x,z)\big) & \text{if } E \text{ is discrete,} \\ \int_E q(x,z)\big(1 - a(x,z)\big)\, dz & \text{if } E \text{ is continuous,} \end{cases}$$

and $\delta_x(z)$ denotes a Dirac mass at x. (Note: p_{MH} is not absolutely continuous with respect to Lebesgue measure when E is continuous, since the probability of moving from x to x is positive.)

Proof. We only prove the discrete case. The chain moves to a new state z if the state z was proposed and accepted, which happens with probability $q(x,z)a(x,z)$. This is the probability of moving from x to z if $x \neq z$. A move from x to x may occur in two different ways:

1. Propose x as new state and accept it, which happens with probability $q(x,x)a(x,x)$.
2. Propose any $z \in E$ and reject it, which happens with probability

$$r(x) = \sum_{z \in E} q(x,z)\big(1 - a(x,z)\big).$$

Putting everything together

$$p_{\text{MH}}(x,z) = q(x,z)a(x,z) + \delta_x(z)r(x). \qquad \square$$

As part of the proof of the previous lemma, we have shown that if $x \neq z$, then

$$p_{\text{MH}}(x,z) = q(x,z)a(x,z).$$

A consequence of this identity is the detailed balance of p_{MH} with respect to f.

Theorem 4.3 (Detailed Balance of Metropolis Hastings). *The Metropolis Hastings kernel p_{MH} satisfies detailed balance with respect to f.*

Proof. We need to show that

$$f(x)p_{\text{MH}}(x,z) = f(z)p_{\text{MH}}(z,x), \quad \forall x, z \in E.$$

For $x = z$ the equality is trivial. If $x \neq z$

$$\begin{aligned} f(x)p_{\text{MH}}(x,z) &= f(x)q(x,z)a(x,z) \\ &= \min\Big\{f(x)q(x,z), f(z)q(z,x)\Big\}. \end{aligned}$$

The right-hand side is symmetric in x and z, and therefore the result follows. $\square$

Note that since detailed balance implies general balance, Theorem 4.3 implies that f is an invariant distribution for p_{MH}.

4.2 ▪ Proposal Kernels

We can classify Metropolis Hastings algorithms based on the form of their proposal kernels. In this section, we study two types of proposal kernels that lead to the independence sampler and the random walk Metropolis Hastings algorithm. Other proposal kernels will be studied in subsequent chapters.

4.2.1 ▪ Independence Sampler

Independence samplers are Metropolis Hastings algorithms where the proposal Markov kernel does not include information on the current state of the chain. This is to say that, for some distribution g on E,

$$q_{\text{Indep}}(x,z) = g(z),$$

i.e. the proposal Markov kernel $q_{\text{Indep}}(x, z)$ does not depend on the current state x or the chain.

Algorithm 4.2 Independence Sampler

Input: Target distribution f, initial distribution π_0, proposal $g(z)$, sample size N.

1: Define the Markov kernel

$$q_{\text{Indep}}(x, z) := g(z).$$

2: Run Metropolis Hastings (Algorithm 4.1) with inputs f, π_0, $q_{\text{Indep}}(x, z)$, N.

Output: Sample $\{X^{(n)}\}_{n=1}^{N}$.

Note that the acceptance probability of the independence sampler reduces to

$$a(x, z) = \min\left\{1, \frac{f(z)}{f(x)}\frac{g(x)}{g(z)}\right\}.$$

By making analogy to importance sampling, we can define $w(x) := \frac{f(x)}{g(x)}$ and express the acceptance probability as follows:

$$a(x, z) = \min\left\{1, \frac{w(z)}{w(x)}\right\}.$$

Thus, moves $x \mapsto z$ for which $w(z) \geq w(x)$ are always accepted.

In practice, independence samplers typically perform poorly. However, they are an important ingredient of the MCMC toolbox because their theoretical properties are well understood. To gain intuition and to illustrate the flavor of the theory, we compare the independence sampler with rejection sampling. Precisely, we establish a relation between the acceptance probability for rejection sampling and the acceptance rate of the independence sampler at stationarity.

Theorem 4.4 (Acceptance Rate of Independence Sampler). *Suppose that the target f and the proposal distribution g for an independence sampler satisfy, for some $M \geq 1$,*

$$f(x) \leq Mg(x), \qquad \forall x \in E. \tag{4.1}$$

Then, the independence sampler at stationarity has acceptance rate greater than $1/M$.

Proof. Let $X \sim f$ and let $Z \sim g$. We work in the continuous setting so that $\mathbb{P}\big(f(z)g(x) = f(x)g(z)\big) = 0$. We need to show that

$$\mathbb{E}\big[a(X, Z)\big] = \mathbb{E}\left[\min\left\{1, \frac{f(Z)}{f(X)}\frac{g(X)}{g(Z)}\right\}\right] \geq \frac{1}{M}.$$

We express the minimum with indicator functions

$$\min\left\{1, \frac{f(Z)}{f(X)}\frac{g(X)}{g(Z)}\right\} = \mathbf{1}_{\left\{\frac{f(Z)}{f(X)}\frac{g(X)}{g(Z)} > 1\right\}} + \frac{f(Z)}{f(X)}\frac{g(X)}{g(Z)}\mathbf{1}_{\left\{\frac{f(Z)}{f(X)}\frac{g(X)}{g(Z)} \leq 1\right\}}$$

to obtain that

$$\begin{aligned}\mathbb{E}\big[a(X, Z)\big] = &\int\int \mathbf{1}_{\{f(z)g(x) > f(x)g(z)\}} f(x)g(z)\, dxdz \\ &+ \int\int \frac{f(z)}{f(x)}\frac{g(x)}{g(z)}\mathbf{1}_{\{f(z)g(x) \leq f(x)g(z)\}} f(x)g(z)\, dxdz.\end{aligned}$$

Now use symmetry and the bound (4.1) to deduce that

$$\begin{aligned}\mathbb{E}\big[a(X,Z)\big] &= 2\int\int \mathbf{1}_{\{f(z)g(x)>f(x)g(z)\}} f(x)g(z)\,dxdz \\ &\geq 2\int\int \mathbf{1}_{\{f(z)g(x)>f(x)g(z)\}} f(x)\frac{f(z)}{M}\,dxdz \\ &= \frac{2}{M}\,\mathbb{P}\Big(f(X_2)g(X_1) > f(X_1)g(X_2)\Big) \\ &= \frac{2}{M}\frac{1}{2} = \frac{1}{M},\end{aligned}$$

where $X_1, X_2 \overset{\text{i.i.d.}}{\sim} f$. □

Pros and Cons: Theorem 4.4 shows that the acceptance rate of the independence sampler at stationarity is higher than the acceptance rate $1/M$ of rejection sampling. Moreover, contrary to rejection sampling, the independence sampler does not throw away rejected draws. However, the independence sampler produces correlated draws and in practice is initialized far from stationarity —several iterations will be needed to approximately reach stationarity (see Theorem 4.5 below). Among Metropolis Hastings algorithms, the independence sampler is rarely used in practice and is mainly of theoretical importance.

As for rejection sampling, it is advisable to choose the independence sampler proposal g to be close to the target f, while at the same time being easy to sample from. Note that in the extreme case where $g = f$, the algorithm outputs draws from f, the acceptance probability is 1, and the Markov chain reaches stationarity immediately. However, choosing $g = f$ is clearly not useful in the interesting case where sampling f directly is not possible. The following result establishes uniform ergodicity of the independence sampler. The notions of total variation distance and uniform ergodicity are reviewed in Definitions A.23 and A.33.

Theorem 4.5 (Uniform Ergodicity of Independence Sampler). *Let p_{Indep} be the Markov kernel of the independence sampler with proposal kernel $q_{\text{Indep}}(x,z) = g(z)$. Suppose that there is $M \geq 1$ such that $f(x) \leq Mg(x)$ for every $x \in E$. Then, for any $x \in E$,*

$$d_{\text{TV}}\big(p^n_{\text{Indep}}(x,\cdot), f\big) \leq \Big(1 - \frac{1}{M}\Big)^n.$$

Proof. We only provide a sketch proof. If $x \neq z$,

$$p_{\text{Indep}}(x,z) = q_{\text{Indep}}(x,z)a(x,z) = g(z)\min\Big\{1, \frac{f(z)}{f(x)}\frac{g(x)}{g(z)}\Big\} = \min\Big\{g(z), \frac{g(x)f(z)}{f(x)}\Big\} \geq \frac{f(z)}{M}.$$

The inequality $p_{\text{Indep}}(x,z) \geq \frac{f(z)}{M}$ also holds when $x = z$. This bound allows us to conclude the result with the same coupling argument used to show ergodicity of finite state space Markov chains in Theorem A.25. A sketch is given below.

Let $\{B_n\}_{n=0}^{\infty}$ be a sequence of i.i.d. Bernoulli($\frac{1}{M}$) random variables, independent of all other randomness, and define

$$W_{n+1} \sim \begin{cases} s(W_n,\cdot) & \text{if } B_n = 0, \\ r(W_n,\cdot) & \text{if } B_n = 1, \end{cases}$$

where $r(x,z) = f(z)$ and

$$s(x,z) = \frac{p_{\text{Indep}}(x,z) - \frac{1}{M}f(z)}{1 - \frac{1}{M}}.$$

The bound $p_{\text{Indep}}(x,z) \geq \frac{f(z)}{M}$ ensures that s defines a Markov kernel. Moreover, $\{W_n\}$ has transition kernel $p_{\text{Indep}}(x,z)$. The rest of the proof is identical to that of Theorem A.25. □

Notice the common underlying structure between Theorem 4.5 and Theorem A.25. In both cases we used a lower bound on the kernel P in the sense that there exist $\delta > 0$ and a probability measure ν such that

$$P^m(x,A) \geq \delta\nu(A), \qquad \forall x \in E,\, \forall A \subset E.$$

Indeed, this condition is equivalent to ergodicity of the chain. Whenever it holds,

$$d_{\text{TV}}(P^n(x,\cdot), f) \leq (1-\delta)^{[\frac{n}{m}]}.$$

Remark 4.6. *It can be shown that if* $\operatorname{ess\,inf}\frac{g(z)}{f(z)} = 0$*, then the independence sampler is not even geometrically ergodic.*

Scaling for the Independence Sampler Suppose we want to use the independence sampler with a proposal distribution chosen from some parametric family $\{g_\theta : \theta \in \Theta\}$. For example, we may have decided to use a normal distribution as our proposal, and we want to determine an appropriate mean and variance. This general type of problem is called "scaling" of Metropolis Hastings methods.

Recall that we have shown that if $\frac{f(x)}{g(x)} \leq M$, then the independence sampler is uniformly ergodic with rate $1 - \frac{1}{M}$. This suggests choosing the parameter θ which minimizes M_θ under the constraint that $\frac{f(x)}{g_\theta(x)} \leq M_\theta$. Note that this is the same constrained optimization we would like to (approximately) solve for in rejection sampling. In many problems of interest, finding an upper bound for the ratio f/g is unfeasible. In such cases, one can monitor the acceptance rate of the independence sampler with various θ's, and choose the proposal g_θ that gives the highest acceptance rate.

4.2.2 ▪ Random Walk Metropolis Hastings

A Random Walk Metropolis Hastings (RWMH) algorithm is a particular type of Metropolis Hastings method, where the proposal Markov kernel is chosen to be of the form

$$q_{\text{RWMH}}(x,z) = g(z-x)$$

for some distribution g.

Algorithm 4.3 Random Walk Metropolis Hastings (RWMH)

Input: Target distribution f, initial distribution π_0, proposal distribution g, sample size N.

1: Define the Markov kernel

$$q_{\text{RWMH}}(x,z) := g(z-x).$$

2: Run Metropolis Hastings (Algorithm 4.1) with inputs f, π_0, $q_{\text{RWMH}}(x,z)$, N.

Output: Sample $\{X^{(n)}\}_{n=1}^N$.

The name RWMH comes from the fact that proposals are made according to a random walk: proposed moves Z^* may be obtained by sampling $\xi^* \sim g$ and setting

$$Z^* = X^{(n)} + \xi^*.$$

The acceptance probability is

$$a(x, z) = \min\left\{1, \frac{f(z)}{f(x)} \frac{g(x-z)}{g(z-x)}\right\}.$$

The original paper by Metropolis et al. proposed an RWMH with g symmetric about 0, which simplifies the expression of the acceptance probability to

$$a(x, z) = \min\left\{1, \frac{f(z)}{f(x)}\right\}.$$

Common choices of distribution g are normal, Student's t-distribution, and uniform (in some bounded subset, when E is unbounded).

Remark 4.7. *If $E = \mathbb{R}^d$ and $q_{\text{RWMH}}(x, z) = g(x - z) = g(z - x)$ for every $x, z \in E$, then the RWMH kernel is not uniformly ergodic for any f. However, if the target f is log-concave in the tails and symmetric, and $g > 0$ is continuous, then the RWMH kernel is geometrically ergodic. If f is not symmetric, the conclusion still holds if additionally $g(x) \leq \ell e^{-\alpha|x|}$, where $\ell > 0$ and α is some constant which quantifies the concavity of $\log(f)$.*

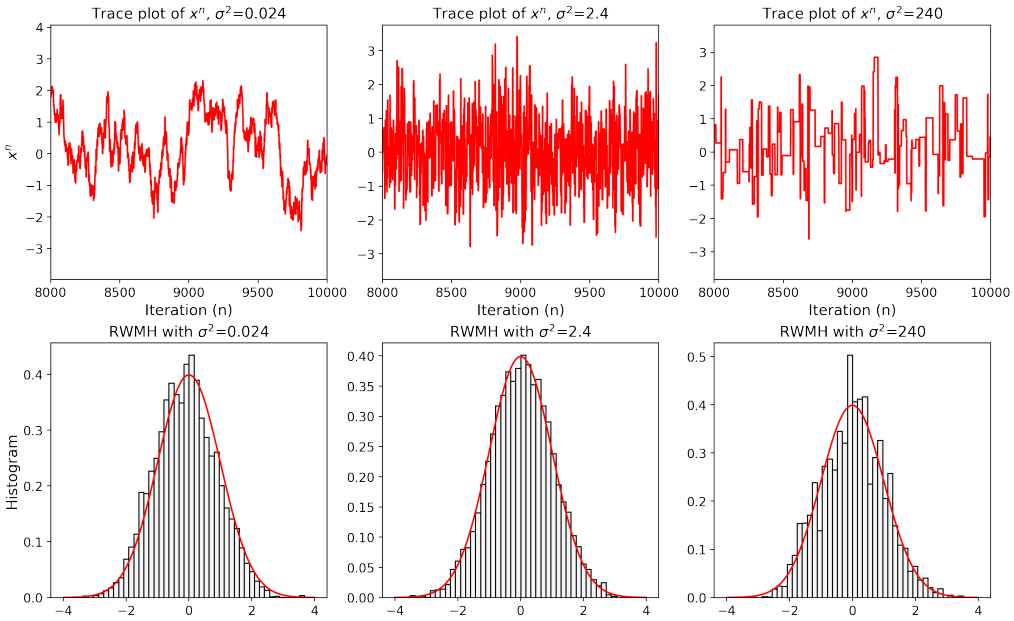


Figure 4.1. *RWMH with standard Gaussian target $f = \mathcal{N}(0, 1)$ and three proposals $g_\sigma = \mathcal{N}(0, \sigma^2)$ with $\sigma^2 = 0.024, 2.4, 240$. The choice $\sigma^2 = 2.4$ leads to good mixing of the chain, whereas $\sigma^2 = 0.024$ leads to low exploration and $\sigma^2 = 240$ leads to many rejections.*

Scaling for RWMH While for the independence sampler it is advantageous to maximize the expected acceptance probability, this is *not* the case for RWMH. Consider for intuition the case where $f = \mathcal{N}(0,1)$ and $g_\sigma = \mathcal{N}(0,\sigma^2)$, as in Figure 4.1. Then,

- Small σ^2 leads to high acceptance rate but poor exploration of the state space.
- Large σ^2 leads to good exploration but low acceptance rate.

There are some widely applicable guidelines for tuning proposals.

1. First, in the simple case of a normal target with a normal distributed random walk proposal, the optimal choice of variance σ^2 can be determined by minimizing the integrated autocorrelation time of the associated chains. The optimal value was found to be $\sigma^2 = 2.4$, with corresponding acceptance rate

$$\alpha = \frac{2}{\pi}\arctan\Big(\frac{2}{\sigma}\Big) = 0.44.$$

   This result has motivated the general recommendation of aiming at an acceptance rate around $1/2$ for low dimensional problems.

2. Second, analysis of RWMH in the setting

$$f_d(x) = f(x_1) \times \cdots \times f(x_d), \quad x = (x_1, \ldots, x_d) \in \mathbb{R}^d,$$
$$g_d(x) = \mathcal{N}(0, \sigma_d^2 I_d),$$

   reveals two important insights.

   First, in the large-d asymptotic, σ_d^2 should be scaled as $1/d$; this in turn implies that the convergence time of the algorithm scales like $\mathcal{O}(d)$.

   Second, in a precise sense and under suitable assumptions, the optimal acceptance probability for large d is roughly 0.234. This result has motivated the general recommendation of aiming at an acceptance rate around $1/4$ for high dimensional problems.

4.3 ▪ Implementation of Markov Chain Monte Carlo Methods

4.3.1 ▪ Initialization Bias and Burn-In

MCMC chains are usually initialized outside stationarity (if we could sample the initial draw from the target we would not be doing MCMC!). To reduce the initialization bias caused by the effect of the starting value, the first M draws may be discarded. Estimation is then based on the states visited after time M:

$$\mathcal{I}_f[h] \approx \frac{1}{N-M}\sum_{n=M+1}^{N} h(X^{(n)}).$$

The initial phase up to time M is called the transient phase or burn-in period. How do we decide on the length of the burn-in period? A first step would be to examine the output of the chain by eye. This is a very crude method, but is very quick and cheap. However, this method should be followed up by a more sophisticated analysis.

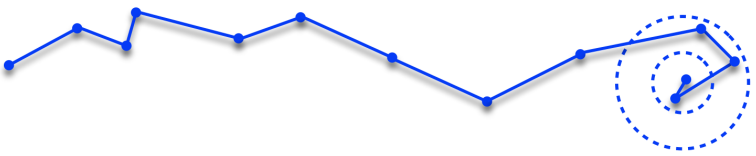


Figure 4.2. *The MCMC chain is typically started outside stationarity, and it may take several iterations to reach a region of high target density.*

4.3.2 ▪ Assessing Convergence

How to assess convergence has been a fiercely discussed topic in the literature. Some of the disagreement is due to the fact that there are different types of convergence. First, there is the issue of whether the distribution of the chain is close to the stationary distribution. Second, there is the issue of how well the chain explores the state space. Generally, chains that do not explore the state space well (we say, mix slowly) tend to approach stationarity more slowly. But note that even if the chain is at stationarity, ergodic averages of a slow mixing chain typically result in inaccurate estimates, a third issue that is discussed under the general keyword of convergence. All three issues are related to each other but are not exactly the same, which can be confusing. For example, a chain with a bi-modal stationary distribution may not have reached equilibrium yet, as it only ever has visited the region around one of the modes. Now suppose we are interested in estimating a function that is equal for both modes. Then, it can happen that the corresponding ergodic average converges much faster than the chain itself.

There are two general approaches to assess convergence to stationarity: convergence rate calculation and convergence diagnostics.

1. **Convergence rate calculation**: When we looked at the speed of convergence for the independence sampler, we essentially performed a convergence rate calculation. The advantage of this approach is, of course, that it is exact. In general, these calculations can be difficult and will only apply to specific cases. Because they are exact methods, they can be overly pessimistic as they take into account any worst-case scenario even if it has a very small probability of occurring. At times they can be so pessimistic that the suggested burn-in period simply becomes impractical.

2. **Convergence diagnostics**: This is the most widely used method. Convergence diagnostics examine the output of the chain and try to detect a feature that may suggest divergence from stationarity. They are usually relatively easy to implement. However, while they may indicate that convergence has not been reached, they are not able to guarantee convergence.

We will now examine some of the most commonly used diagnostics.

Geweke's Method Geweke proposed a diagnostic method based on spectral analysis of time series. Consider two subsequences of a run of your MCMC, one consisting of states immediately after burn-in and the other consisting of states at the very end of the run. Suppose that the run is of length N and we want to examine if a burn-in of M steps is sufficient. Consider ergodic averages

$$\mathcal{I}_f^A[h] := \frac{1}{N_A} \sum_{n=M+1}^{M+N_A} h(X^{(n)}), \qquad \mathcal{I}_f^B[h] := \frac{1}{N_B} \sum_{n=N-N_B}^{N} h(X^{(n)})$$

with $M + N_A < N - N_B$. If the chain has converged, then the two ergodic averages should be similar. How similar they should be can be quantified using an appropriate central limit theorem. The variance of ergodic averages under stationarity is determined by the spectral density of the time series $\{h(X^{(n)})\}_{n=1}^{\infty}$, defined as

$$S(\omega) = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \gamma_k \exp(ik\omega),$$

where $\gamma_k = \mathrm{Cov}\big(h(X^{(0)}), h(X^{(k)})\big)$ is the k-lag autocovariance of the time series (see Definition A.35 in Appendix A). The following asymptotic result holds. If the ratios N_A/N and N_B/N are fixed with $(N_A + N_B)/(N - M) < 1$, then, under stationarity and as $N \to \infty$,

$$\frac{\mathcal{I}_f^A[h] - \mathcal{I}_f^B[h]}{\sqrt{\frac{1}{N_A}\widehat{S}_A(0) + \frac{1}{N_B}\widehat{S}_B(0)}} \to \mathcal{N}(0,1),$$

where $\widehat{S}_A(0)$ and $\widehat{S}_B(0)$ are spectral estimates of the variances. We can use the above asymptotics to test whether the means of the two sequences are equal subject to variation. Geweke originally suggested taking the values $N_A = N/10$ and $N_B = N/2$.

Gelman and Rubin's Method This method uses several chains started in initial states that are over-dispersed compared to the stationary distribution. The idea is that if there is initialization bias, then the chains will be close to different modes while under the stationary regime the chains should behave similarly. If the chain tends to get stuck in a mode, then the path appears to be experiencing stable fluctuations, but the chain has not converged yet. In this case, we speak of *meta-stability*, which is often caused by multi-modality of the stationary distribution. Gelman and Rubin apply concepts from ANOVA techniques (and thus rely on Normal asymptotics). We assume each of a total of J chains are run for $2N$ iterations where the first N iterations are classified as burn-in. We first compute the variance of the means of the chains:

$$B = \frac{1}{J-1} \sum_{j=1}^{J} \Big(\mathcal{I}_f^j[h] - \bar{\mathcal{I}}_f^j[h] \Big)^2.$$

Here we run J chains, $\mathcal{I}_f^j[h]$ is the ergodic average of the j-th chain based on the last N iterations, and $\bar{\mathcal{I}}_f^j[h]$ is the average of the ergodic averages of the J chains. This is interpreted as the between chain variance. As $N \to \infty$ the between chain variance will tend to zero.

Then, we compute the within chain variance W, that is, the mean of the variance of each chain:

$$W = \frac{1}{J} \sum_{j=1}^{J} \frac{1}{N-1} \sum_{n=N+1}^{2N} \Big(h(X_j^{(n)}) - \mathcal{I}_f^j[h] \Big)^2,$$

where $X_j^{(n)}$ denotes the n-th iterate of the j-th chain. We calculate the weighted average of both variance estimates

$$V = \frac{N-1}{N} W + B$$

to obtain an estimate of the target variance. We then monitor $R = \sqrt{V/W}$, which is called the potential scale reduction factor. R tends to be larger than one and will converge towards one as stationarity is reached. Using Normal asymptotics, one can show that R has an F-distribution and the hypothesis of $R = 1$ can be tested. As a rule of thumb, if the 0.975-quantile is less than 1.2, no lack of convergence has been detected.

Raftery and Lewis' Method In this diagnostic, a posterior quantile p and an acceptable tolerance r for error in computing p are specified. In addition, the user specifies the desired probability α of obtaining an estimate for p within the prescribed error tolerance. Raftery and Lewis suggested a way to estimate the number N of iterations and the burn-in period M that are necessary to satisfy the prescribed conditions.

4.4 ▪ Discussion and Bibliography

The Metropolis Hastings algorithm was introduced in [162] restricted to the case of Gaussian random walk proposals; the extension to user-chosen proposal kernel was proposed in [108]. The paper [100] introduced an adaptive strategy, where the proposal kernel is updated along the process using the full information gathered so far. Historical overviews of the Metropolis Hastings algorithm and related MCMC methods include [191, 61]. Many books [88, 37, 192] and survey articles [36, 224] are devoted to the Metropolis Hastings algorithm; adaptive strategies are surveyed in [13, 199].

As shown in Theorem 4.3, the Metropolis Hastings algorithm is designed to ensure detailed balance with respect to the target; in other words, Markov chains generated using the Metropolis Hastings algorithm are *reversible*. The paper [26] shows that the algorithm can be modified to generate *non-reversible* chains that may have better properties in terms of mixing behavior or asymptotic variance. Uniform ergodicity of the independence sampler and geometric ergodicity of RWMH were established in [161]. A comparison between rejection sampling, importance sampling, and the independence sampler can be found in [144]. The study of optimal scaling for Metropolis Hastings algorithms, which originates from [195, 202] and is reviewed in [197], is still an active area of research [239]. The convergence diagnostics by Raftery, Geweke, and Gelman and Rubin were introduced in [188, 84, 81]. We refer to [167, 240, 229, 228] for other important examples of convergence diagnostics, and to [38] for an early survey on this topic.

An important feature of the Metropolis Hastings algorithm is that it can be implemented without knowledge of the normalizing constant of the target. However, in many applications such as Bayesian inverse problems [218], machine learning with tall data [16], and genetics [17] evaluating the unnormalized target density to compute the Metropolis Hastings acceptance probability can be expensive. If a *deterministic* cheap-to-evaluate approximation of the target is available, this issue can be alleviated using delayed-acceptance Metropolis Hastings algorithms [50, 69, 68, 211]. The idea is similar to that behind the envelope rejection sampling algorithm studied in Chapter 2: first, we do an accept/reject step with the cheap-to-evaluate target approximation; if the move is accepted, we do a second accept/reject step with the expensive-to-evaluate target. If a *stochastic* unbiased estimator of the unnormalized target density is available, pseudo-marginal MCMC algorithms [17, 12] can be employed. Particular instances of pseudo-marginal MCMC algorithms are particle MCMC algorithms [10] that rely on unbiased estimates computed via particle filters (see Chapter 9). Particle MCMC algorithms are powerful algorithms for parameter estimation in hidden Markov models, see e.g. [93].

In this chapter, we introduced the Metropolis Hastings algorithm on a countable or uncountable state space $E \subset \mathbb{R}^d$. Reversible jump MCMC algorithms [96] allow to sample posterior distributions on spaces of varying dimension; this important extension of the Metropolis Hastings framework enables Bayesian inference in applications where the number of parameters in the model is unknown. An adaptive, delayed-rejection algorithm for reversible jump MCMC was introduced in [97]. Another important extension of the Metropolis Hastings framework concerns its formulation in general state spaces [225, 198], which enables for instance function space sampling [52]. A common limitation of MCMC methodology is that it is not easy to parallelize due to its serial structure. We refer to [41, 91, 210] for attempts to parallelize Metropolis Hastings algorithms. These approaches often propose several moves in each iteration, an idea also found

in [147]. Finally, we remark that MCMC algorithms that rely on piecewise deterministic Markov processes are studied for instance in [74, 25, 32].

Chapter 5
Gibbs Sampling

The Gibbs sampler is a Markov chain Monte Carlo algorithm for multivariate target distributions. Gibbs samplers operate coordinate-wise: each new draw differs from the previous one in only a subset of variables, which are updated by sampling from the distribution of those variables conditioned to all others. This idea is extremely powerful when sampling conditionals is easier than sampling the joint distribution. In many problems where conditionals are known, Gibbs samplers avoid the curse of dimension by sampling these low dimensional conditional distributions instead of directly sampling the joint target distribution.

Gibbs samplers were conceived and developed independently of the Metropolis Hastings algorithms studied in Chapter 4, and their design and analysis have a unique flavor. However, Gibbs samplers can be seen as a particular instance of the Metropolis Hastings framework, where the proposal kernel is defined by so-called *full conditionals* of the target distribution. Remarkably, such a choice of proposal kernel implies that proposed moves are always accepted. Hence, in contrast to other Metropolis Hastings algorithms that rely on a step-size parameter to control the acceptance rate, Gibbs samplers require no tuning. On the other hand, the parameterization of the problem can severely impact the behavior of the algorithm. Specifically, Gibbs samplers may converge slowly if the variables are strongly correlated, or, relatedly, when the target is poorly conditioned. Thus, care should be taken to conveniently parameterize the variables of interest and understand their dependencies before implementing the Gibbs sampler.

This chapter is organized as follows. Section 5.1 introduces the idea behind the Gibbs sampler through a simple example. The main algorithm is presented in Section 5.2, where we also give the formal definition of full conditionals, show that Gibbs samplers belong to the Metropolis Hastings family, and prove that choosing full conditionals as proposal kernels results in moves that are always accepted. In Section 5.3 we study the convergence of Gibbs samplers in finite state spaces and for Gaussian target distributions, where we show that the algorithm converges slowly if the variables are highly correlated. Section 5.4 closes with bibliographical remarks.

5.1 ▪ Motivating Example

Consider a likelihood model

$$f(y|\mu,\sigma) = \frac{1}{(2\pi\sigma^2)^{K/2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{k=1}^{K}(y^{(k)}-\mu)^2\right),$$

where $y = \{y^{(k)}\}_{k=1}^{K}$ is the observed data.

We take a Bayesian approach and choose the following priors for the precision $\tau = \frac{1}{\sigma^2}$ and mean μ:

$$\tau \sim \text{Gamma}(\alpha, \beta),$$
$$\mu \sim \mathcal{N}(m, s^2).$$

The posterior for μ, τ is

$$f(\mu, \tau|y) \propto \exp\left(-\frac{1}{2s^2}(\mu - m)^2\right) \exp\left(-\frac{\tau}{2}\sum_{k=1}^{K}(y^{(k)} - \mu)^2\right) \exp\big(-\beta\tau\big)\tau^{\alpha-1+\frac{K}{2}}.$$

There is no closed form expression for the normalizing constant. Note that:

1. Conditional on τ,

$$\mu|\tau \sim \mathcal{N}\left(\frac{K\bar{y}\tau + ms^{-2}}{K\tau + s^{-2}}, \frac{1}{K\tau + s^{-2}}\right),$$

where $\bar{y}$ is the sample average.

2. Conditional on μ,

$$\tau|\mu \sim \text{Gamma}\left(\alpha + \frac{K}{2}, \beta + \frac{1}{2}\sum_{k=1}^{K}(y^{(k)} - \mu)^2\right).$$

Since both of these conditional distributions are standard, it seems natural to use them as proposal distributions. Suppose that $(\mu^{(n)}, \tau^{(n)})$ are given. Inspired by the Metropolis Hastings algorithm studied in Chapter 4, a natural idea would be to obtain $(\mu^{(n+1)}, \tau^{(n+1)})$ as follows:

1. Propose

$$\mu^* \sim \mathcal{N}\left(\frac{K\bar{y}\tau^{(n)} + ms^{-2}}{K\tau^{(n)} + s^{-2}}, \frac{1}{K\tau^{(n)} + s^{-2}}\right).$$

2. Accept $\mu^{(n+1)} = \mu^*$ with probability $a\big((\tau^{(n)}, \mu^{(n)}), (\tau^{(n)}, \mu^*)\big)$.
Otherwise, set $\mu^{(n+1)} = \mu^{(n)}$.

3. Propose

$$\tau^* \sim \text{Gamma}\left(\alpha + \frac{K}{2}, \beta + \frac{1}{2}\sum_{k=1}^{K}(y^{(k)} - \mu^{(n+1)})^2\right).$$

4. Accept $\tau^{(n+1)} = \tau^*$ with probability $a\big((\tau^{(n)}, \mu^{(n+1)}), (\tau^*, \mu^{(n+1)})\big)$.
Otherwise set $\tau^{(n+1)} = \tau^{(n)}$.

The fact that the normalizing constant of the posterior is unknown does not obstruct the implementation of this natural scheme. Moreover, the theory we will develop in the next section will show that the acceptance probability in steps 2 and 4 is always 1. Therefore, we could simplify the algorithm as follows:

1. Sample

$$\mu^{(n+1)} \sim \mathcal{N}\left(\frac{K\bar{y}\tau^{(n)} + ms^{-2}}{K\tau^{(n)} + s^{-2}}, \frac{1}{K\tau^{(n)} + s^{-2}}\right).$$

2. Sample

$$\tau^{(n+1)} \sim \text{Gamma}\left(\alpha + \frac{K}{2}, \beta + \frac{1}{2}\sum_{k=1}^{K}(y^{(k)} - \mu^{(n+1)})^2\right).$$

We have derived a Gibbs sampler for the posterior distribution $f(\mu, \tau|y)$.

5.2 ▪ Full Conditionals and the Gibbs Sampler

We next define the concept of full conditionals. For clarity, we give the definition in both discrete and continuous cases.

Definition 5.1 (Full Conditionals).

i) If E is discrete and $f(x) = f(x_1, \dots, x_d)$ is the p.m.f. of a random vector $X = (X_1, \dots, X_d)$ taking values in $E \times \cdots \times E = E^d$. We denote

$$f(x_{-j}) = \sum_{\xi \in E} f(x_1, \dots, x_{j-1}, \xi, x_{j+1}, \dots, x_d).$$

The j-th full conditional p.m.f. is:

$$\begin{aligned} f_j(x_j|x_{-j}) = f_j(x_j|x_i, i \neq j) &= \mathbb{P}(X_j = x_j | X_i = x_i, i \neq j) \\ &= \frac{\mathbb{P}(X_j = x_j, X_i = x_i, i \neq j)}{\mathbb{P}(X_i = x_i, i \neq j)} \\ &= \frac{f(x_1, \dots, x_{j-1}, x_j, x_{j+1}, \dots x_d)}{f(x_{-j})}. \end{aligned}$$

ii) If E is continuous and $f(x) = f(x_1, \dots, x_d)$ is the p.d.f. of the random vector $X = (X_1, \dots, X_d)$ taking values on $E \times \cdots \times E = E^d$, we denote

$$f(x_{-j}) = \int_E f(x_1, \dots, x_{j-1}, \xi, x_{j-1}, \dots, x_d)\, d\xi,$$

and define the j-th full conditional density

$$f_j(x_j|x_{-j}) = f_j(x_j|x_i, i \neq j) = \frac{f(x_1, \dots, x_{j-1}, x_j, x_{j+1}, \dots, x_d)}{f(x_{-j})}.$$ □

The distribution defined by $f_j(x_j|x_i, i \neq j)$ is called the j-th full conditional distribution of f. We will only need to know these distributions up to a normalizing constant, and so terms that do not depend on x_j can be ignored. In particular, it is useful to note that full conditionals are proportional to the joint density.

Example 5.2 (Full Conditionals of Multivariate Gaussians). Consider a multivariate Gaussian random vector $X = (X_1, \dots, X_d) \sim \mathcal{N}(\mu, H^{-1})$. Then, $X_j|X_{-j} \sim \mathcal{N}(\nu_j, H_{jj}^{-1})$, with

$$\nu_j = \mu_j - H_{jj}^{-1} \sum_{k \neq j} H_{jk}(x_k - \mu_k).$$

To check this, recall that a univariate normal random variable with mean α and precision β has p.d.f. proportional to

$$\exp\Big(-\frac{1}{2}x^2\beta + x\beta\alpha\Big). \tag{5.1}$$

Assume for the moment that $\mu = 0 \in \mathbb{R}^d$. Using that the j-th full conditional is proportional to the joint density,

$$f_j(x_j|x_{-j}) \propto \exp\Big(-\frac{1}{2}x^T H x\Big) \propto \exp\Big(-\frac{1}{2}x_j^2 H_{jj} - x_j \sum_{k \neq j} H_{jk} x_k\Big). \tag{5.2}$$

Thus $X_j|X_{-j}$ is univariate normal. Comparing the coefficients of the quadratic terms in Equations (5.1) and (5.2) we see that the precision is H_{jj}, and comparing the coefficients for the linear term we see that the mean is ν_j. Finally, if X has mean μ, then applying the above argument to $X - \mu$ gives the result. □

The Gibbs sampler is based on sampling full conditionals. It is not immediately obvious that full conditionals (unlike marginals) can specify a distribution uniquely, provided they are consistent. Sufficient and necessary conditions are given by the Hammersley-Clifford theorem, which for simplicity we only present in the 2-dimensional case.

Theorem 5.3 (Hammersley–Clifford 2-dimensional Case). *If* $\int \frac{f(x_2|x_1)}{f(x_1|x_2)}\,dx_2$ *exists, then the joint density associated with* $f(x_2|x_1)$ *and* $f(x_1|x_2)$ *is given by*

$$f(x_1, x_2) = \frac{f(x_2|x_1)}{\int \frac{f(x_2|x_1)}{f(x_1|x_2)}\,dx_2}.$$

Proof. Since $f(x_2|x_1)f(x_1) = f(x_1|x_2)f(x_2)$,

$$\int \frac{f(x_2|x_1)}{f(x_1|x_2)}\,dx_2 = \int \frac{f(x_2)}{f(x_1)}\,dx_2 = \frac{1}{f(x_1)},$$

and so the claim follows if the integral above exists. □

We are now ready to introduce the Gibbs sampler.

Algorithm 5.1 Gibbs Sampler

Input: target $f(x_1, \ldots, x_d)$, initialization $X^{(0)} = \left(X_1^{(0)}, \ldots, X_d^{(0)}\right)$, sample size N.

For $n = 1, \ldots, N$ do:

1: Choose $j \in \{1, \ldots, d\}$ (randomly or sequentially).

2: Sample $X_j^{(n)} \sim f_j\left(x_j|X_i^{(n-1)}, i \neq j\right)$ and set

$$X^{(n)} = \left(X_1^{(n-1)}, \ldots, X_{j-1}^{(n-1)}, X_j^{(n)}, X_{j+1}^{(n-1)}, \ldots, X_d^{(n-1)}\right).$$

Output: Sample $\{X^{(n)}\}_{n=1}^N$.

Pros and Cons: An important advantage of the Gibbs sampler is that there are no parameters to choose and tune. The Gibbs sampler is particularly useful in problems where the target distribution cannot be sampled directly, but its full conditionals are easy to sample from; e.g. in hierarchical Bayesian models, mixture models, and Bayesian missing data problems. Updating only one variable at a time makes the implementation scalable to high dimensional settings. However, the convergence can be slow when the individual variables are strongly correlated. Moreover, as other methods based on Markov chain sampling, the Gibbs sampler is not parallelizable.

There are two main ways to improve the performance of the Gibbs sampler when the variables are correlated.

1. Reparametrize the distribution so that the new variables are less correlated.

2. Block variables that are highly correlated. Suppose for example that X_1, X_2 are highly correlated and we want to sample from the distribution of (X_1, X_2, X_3). Then, we could do:
   - Sample $X_1, X_2 | X_3$.
   - Sample $X_3 | X_1, X_2$.

We conclude this section showing that the Gibbs sampler is a Metropolis Hastings algorithm.

Theorem 5.4 (Gibbs Sampler is a Metropolis Hastings Algorithm). *The Gibbs sampler is a Metropolis Hastings algorithm that uses full conditionals as proposals. Proposed moves are always accepted.*

Proof. Let $x = (x_1, \dots, x_d)$ and $z = (z_1, \dots, z_d)$ with $x_i = z_i$ for $i \neq j$.

$$\begin{aligned} a(x,z) &= \min\left\{1, \frac{f(z)}{f(x)} \frac{f_j(x_j|z_i, i \neq j)}{f_j(z_j|x_i, i \neq j)}\right\} \\ &= \min\left\{1, \frac{f(z)}{f(x)} \frac{f(x)/f(z_{-j})}{f(z)/f(x_{-j})}\right\} \quad \text{(by definition of full conditional)} \\ &= \min\left\{1, \frac{f(z)}{f(x)} \frac{f(x)/f(x_{-j})}{f(z)/f(x_{-j})}\right\} \qquad \big(f(z_{-j}) = f(x_{-j}) \text{ because } x_i = z_i \, \forall i \neq j\big) \\ &= 1. \end{aligned}$$

□

The Gibbs sampler is a particular example of a Metropolis Hastings algorithm, and therefore its transition kernel satisfies detailed balance with respect to f. We show this directly as an exercise in the following proposition.

Proposition 5.5. *Each transition of the Gibbs sampler satisfies detailed balance with respect to the target distribution f.*

Proof. Denote by p_{GS} the transition kernel of the Gibbs sampler and let

$$\begin{aligned} x &= (x_1, \dots, x_d), \\ z &= (x_1, \dots, x_{j-1}, \xi, x_{j-1}, \dots, x_d). \end{aligned}$$

Then,

$$\begin{aligned} f(x) p_{\text{GS}}(x,z) &= f(x) \frac{f(z)}{f(x_{-j})} = f(z) \frac{f(x)}{f(x_{-j})} \\ &= f(z) f_j(x_j|x_i, i \neq j) = f(z) p_{\text{GS}}(z,x). \end{aligned}$$

□

Example 5.6 (2D Slice Sampler). Recall that when we introduced rejection sampling in Chapter 2 we observed that given a density $f(x)$, $x \in \mathbb{R}$, we can represent it as the marginal of $f(x,u) = \mathbf{1}_{\{0<u<f(x)\}}$.

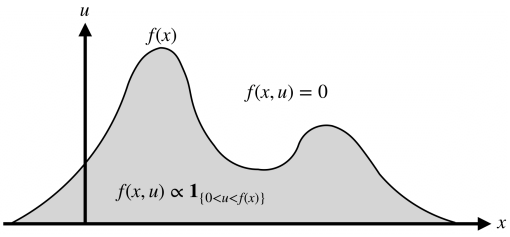


Figure 5.1. $f(x,u) = \boldsymbol{1}_{\{0<u<f(x)\}}$.

The full conditionals are

$$f(x|u) \propto \mathbf{1}_{\{0<u<f(x)\}},$$
$$f(u|x) \propto \mathbf{1}_{\{0<u<f(x)\}}.$$

We can use the Gibbs sampler to obtain samples from $f(x,u)$. Keeping the x-variable, we obtain samples from $f(x)$. This specific type of Gibbs sampler is called the 2D slice sampler. □

5.3 ▪ Convergence of the Gibbs Sampler

As we know, detailed balance implies that the Gibbs sampler has the desired invariant distribution. Therefore, if the chain is ergodic, then its distribution converges to the target. The following lemma gives a sufficient condition for irreducibility.

Lemma 5.7. *Suppose that the target $f(x_1,\ldots,x_d)$ satisfies:*

$$\underbrace{f_i(x_i)}_{\textit{Marginal}} > 0 \quad \forall i \in \{1,\ldots,d\} \Longrightarrow f(x_1,\ldots,x_d) > 0.$$

Then, the Gibbs sampler is f-irreducible.

Example 5.8 (Reducible Gibbs Sampler). Consider the following distribution in $\mathbb{R}^2$:

$$f(x_1,x_2) = \frac{1}{2}\mathbf{1}_{\{0\le x_1\le 1, 0\le x_2\le 1\}} + \frac{1}{2}\mathbf{1}_{\{-1\le x_1\le 0, -1\le x_2\le 0\}},$$

which is depicted in Figure 5.2. The full conditionals are:

$$f(x_1|x_2) = \begin{cases} \mathbf{1}_{\{x_1\in[0,1]\}} & x_2 \in (0,1), \\ \mathbf{1}_{\{x_1\in[-1,0]\}} & x_2 \in (-1,0), \end{cases}$$
$$f(x_2|x_1) = \begin{cases} \mathbf{1}_{\{x_2\in(0,1)\}} & x_1 \in (0,1), \\ \mathbf{1}_{\{x_2\in(-1,0)\}} & x_1 \in (-1,0). \end{cases}$$

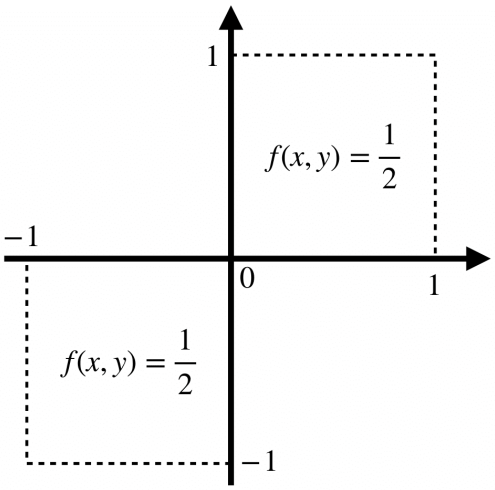


Figure 5.2. *Target density for which the Gibbs sampler is reducible.*

If we used a Gibbs sampler, it would be reducible: if we start in one of the two squares, it would never reach the other square. Note that the condition of the previous lemma fails here:

$$f_1(x_1) = \frac{1}{2}\mathbf{1}_{\{-1\le x_1\le 1\}},$$
$$f_2(x_2) = \frac{1}{2}\mathbf{1}_{\{-1\le x_2\le 1\}},$$
$$f(-0.5, 0.5) = 0 \quad \text{but} \quad f_1(-0.5) > 0, \quad f_2(0.5) > 0. \qquad \square$$

Remark 5.9. *It can be shown that Harris recurrence holds if the transition kernel is absolutely continuous with respect to an appropriate dominating measure.*

5.3.1 ▪ Convergence for Finite State Distributions

Convergence of the Gibbs sampler in a finite state space can be established using the results on ergodicity of Markov chains reviewed in Appendix A. Proposition 5.5 ensures that the target is the invariant distribution of the chain; convergence to the target follows if we guarantee that the chain is irreducible and aperiodic.

Theorem 5.10 (Convergence of Gibbs Sampler in Finite State Space). *Consider a target p.m.f. $f(x)$ on a finite state space $E = \{1, \dots, d\}$, and assume that $f(x) > 0$ for all $x \in E$. Suppose we run a Gibbs sampler on f by sampling each of the full conditionals f_j in sequential order. Then, there is $\epsilon > 0$ such that, for any $x \in E$,*

$$d_{\text{TV}}\big(p^n_{\text{GS}}(x, \cdot), f\big) \le \big(1 - \epsilon\big)^n,$$

where p_{GS} denotes the Markov kernel defined by one swipe of sampling all full conditionals f_j, $1 \le j \le d$.

Proof. The assumption that $f(x) > 0$ for all $x \in E$ implies that all full conditionals are positive. Thus, the Markov kernel p_{GS} is bounded below by a positive constant and the result follows from Theorem A.25 on ergodicity of Markov chains in finite state space. □

5.3.2 ▪ Convergence for Gaussian Distributions

Now we extend our objective to continuous distributions. Instead of investigating a general continuous distribution, we will focus on multivariate Gaussian targets $f = \mathcal{N}(\mu, \Sigma) = \mathcal{N}(\mu, H^{-1})$. Moreover, we restrict our analysis to the following vanilla Gibbs sampler that makes deterministic block updates:

Algorithm 5.2 Deterministic Update Gibbs Sampler (DUGS)

Input: Target $f(x_1, \ldots, x_d)$, initialization $X^{(0)} = \big(X_1^{(0)}, \ldots, X_d^{(0)}\big)$, sample size N.
for $n = 1, \ldots, N$ **do**
  for $j = 1, \ldots, d$ **do**
    Sample $X_j^{(n)} \sim f_j\big(X_j | X_1^{(n)}, \ldots, X_{j-1}^{(n)}, X_{j+1}^{(n-1)}, \ldots, X_d^{(n-1)}\big)$.
    Update the j-th coordinate by setting

$$X^{(n)} = \Big(X_1^{(n)}, \ldots, X_{j-1}^{(n)}, X_j^{(n)}, X_{j+1}^{(n-1)}, \ldots, X_d^{(n-1)}\Big).$$

  end for
  Record sample $X^{(n)}$.
end for
Output: Sample $\{X^{(n)}\}_{n=1}^N$.

We first find a compact way to write the deterministic Gibbs updates. We decompose the precision matrix $H = H_L + H_D + H_U$ into its lower-triangular, diagonal, and upper-triangular components. We have the following lemma.

Lemma 5.11. *Let $\{X^{(n)}\}_{n\geq 0}$ be the Gibbs output with target $f = \mathcal{N}(\mu, \Sigma) = \mathcal{N}(\mu, H^{-1})$ and initial state $X^{(0)}$. Then, $\{X^{(n)}\}_{n\geq 0}$ satisfies the following recurrence:*

$$X^{(n+1)} - \mu = -(H_L + H_D)^{-1} H_U (X^{(n)} - \mu) + (H_L + H_D)^{-1} U^{(n)}, \tag{5.3}$$

where $U^{(n)} \sim \mathcal{N}(0, H_D)$.

Proof. Denote the state at iteration n by $x = (x_1, x_2, \ldots, x_d)$ and the state at iteration $n+1$, after a Gibbs update of all coordinates, by $x' = (x'_1, x'_2, \ldots, x'_d)$. Using Example 5.2 we can write the Gibbs update on block j as

$$H_{jj}(x'_j - \nu_j) = u_j,$$

where $u_j \sim \mathcal{N}(0, H_{jj})$ with

$$\nu_j = \mu_j - H_{jj}^{-1} \sum_{i<j} H_{ji}(x'_i - \mu_i) - H_{jj}^{-1} \sum_{i>j} H_{ji}(x_i - \mu_i)$$

since we are conditioning on $x'_1, x'_2, \ldots, x'_{j-1}, x_{j+1}, \ldots, x_d$. Expanding out ν_j gives

$$H_{jj}(x'_j - \mu_j) + [H_{j1}; \ldots; H_{jj-1}]^T (x'_{1:j-1} - \mu_{1:j-1}) + [H_{jj+1}; \ldots; H_{jd}]^T (x_{j+1:d} - \mu_{j+1:d}) = u_j.$$

[4] $x_{1:j}$ stands for $[x_1; \ldots; x_j]$.

Therefore, a full Gibbs update can be written in vectorized form as

$$(H_L + H_D)(x' - \mu) + H_U(x - \mu) = u,$$

where $u \sim \mathcal{N}(0, H_D)$ as desired. □

The matrix

$$B = -(H_L + H_D)^{-1} H_U$$

is known as the Gauss-Seidel companion matrix. The Gauss-Seidel recurrence in (5.3) belongs to a larger family of iterative techniques to solve linear systems of equations via matrix splitting. It is clear that a sufficient condition for convergence is that $\rho(B) < 1$. This condition holds provided that the matrix H is symmetric and positive definite.

In the following theorem, we will use that if $f = \mathcal{N}(\mu, \Sigma)$ and $\widetilde{f} = \mathcal{N}(\widetilde{\mu}, \widetilde{\Sigma})$, then

$$d_{\chi^2}(f \,\|\, \widetilde{f}) = \frac{|W|}{\sqrt{|2W - I|}} e^{u^T (2W - I)^{-1} \Sigma^{-1} u} \tag{5.4}$$

with $W = \widetilde{\Sigma}\Sigma^{-1}$ and $u = \mu - \widetilde{\mu}$.

Theorem 5.12 (Convergence of Gibbs Sampler for Gaussian Targets). *Let the target distribution of the Deterministic Update Gibbs Sampler (DUGS) be $f = \mathcal{N}(\mu, H^{-1}) = \mathcal{N}(\mu, \Sigma)$, and let p_{DUGS} denote the Markov kernel defined by one full swipe of the DUGS scheme. Then,*

$$\mathbb{E}_f \Big[d_{\chi^2}\big(p^n_{\text{DUGS}}(x^{(0)}, \cdot) \| f\big) \Big] \sim o\big(\rho(B)^n\big),$$

where $B = -(H_L + H_D)^{-1} H_U$.

Proof. We rewrite the recurrence relation of Gibbs updates derived in (5.3) as

$$X^{(n)} - \mu = B(X^{(n-1)} - \mu) + z^{(n)},$$

where $z^{(n)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma_z)$ for some Σ_z that we will determine in what follows. Taking expectation on both sides gives

$$\mathbb{E}\big[X^{(n)} - \mu\big] = B\mathbb{E}\big[X^{(n-1)} - \mu\big].$$

On the other hand, we know that $\mathcal{N}(\mu, \Sigma)$ is the invariant distribution, so $\mathbb{V}\big[x\big] = \mathbb{V}\big[Bx\big] + \mathbb{V}\big[z\big]$, where x has the target distribution and z is the noise in the recurrence equation. This implies that $\Sigma_z = \Sigma - B\Sigma B^T$. Taking the variance on both sides of the original recurrence relation yields

$$\begin{aligned}
\mathbb{V}\big[X^{(n)}\big] &= \mathbb{V}\big[BX^{(n-1)}\big] + \Sigma_z \\
&= B\mathbb{V}\big[X^{(n-1)}\big]B^T + \Sigma - B\Sigma B^T \\
&= \Sigma + B\Big(\mathbb{V}\big[X^{(n-1)}\big] - \Sigma\Big)B^T.
\end{aligned}$$

Let μ_n and Σ_n be the mean and variance of the conditional distribution $X^{(n)}|x^{(0)}$. Then,

$$\begin{aligned}
\mu_n &= \mu + B^n(x^{(0)} - \mu), \\
\Sigma_n &= \Sigma + B^n(0 - \Sigma)(B^T)^n = \Sigma - B^n\Sigma(B^T)^n,
\end{aligned}$$

because $\Sigma^{(0)} = 0$. Using Equation (5.4), we have

$$d_{\chi^2}\big(p^n_{\text{DUGS}}(x^{(0)}, \cdot) \| f\big) = \frac{|W|}{\sqrt{|2W - I|}} e^{u^T (\Sigma^{(n)})^{-1} (2W - I)^{-1} u} - 1,$$

with

$$W = \Sigma_n \Sigma^{-1} = I - B^n \Sigma (B^T)^n,$$
$$u = B^n (x^{(0)} - \mu).$$

Observe that the only $x^{(0)}$-dependence lies in the exponent. Integrating over $x^{(0)} \sim f$ gives

$$\begin{aligned}\mathbb{E}_f \left[d_{\chi^2}(p^n_{\text{DUGS}}(x^{(0)} \| f) \right] + 1 &= \frac{|W|}{\sqrt{|2W - I|}} \int e^{u^T (\Sigma^{(n)})^{-1} (2W - I)^{-1} u} f(x) dx \\ &= \frac{|W|}{\sqrt{|2W - I|}} \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} \int e^{\frac{1}{2}(x-\mu)^T (2(B^T)^t \Sigma^{-1} W^{-1} (2W - I)^{-1} B^t - \Sigma^{-1})(x-\mu)} \\ &= \frac{|W|}{\sqrt{|2W - I||U|}},\end{aligned}$$

where $U = I - 2(B^T)^n \Sigma^{-1} W^{-1} (2W - I)^{-1} B^n \Sigma$. From the definition of W, we obtain the desired rate $\rho(B)$. □

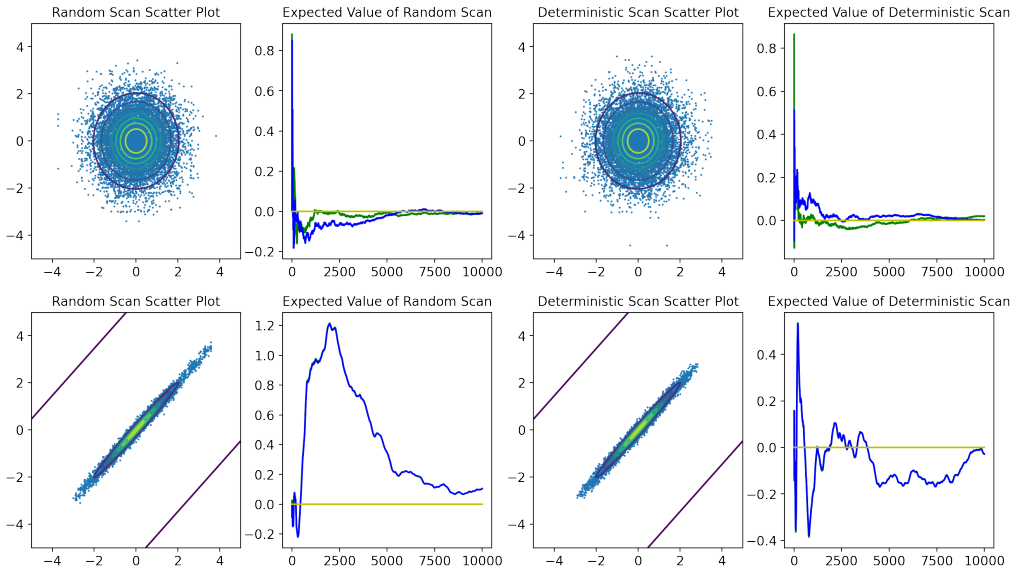


Figure 5.3. *Gibbs samplers for a zero-mean bivariate Gaussian distribution with covariance matrix* $[1, \alpha; \alpha, 1]$. *Top row:* $\alpha = 0.01$. *Bottom row:* $\alpha = 0.99$.

5.4 ▪ Discussion and Bibliography

The Gibbs sampler was introduced in [82]. The paper [80] further popularized the approach, making apparent its usefulness for Bayesian inference. The derivation of full conditionals for Gaussians in Example 5.2 can be found for instance in [205]. Example 5.6 briefly introduced the slice sampler for univariate targets. We refer to [169] for multivariate extensions and to [166] for slice samplers for function space sampling. A comparison between random and deterministic updates can be found in [8]. Simple conditions on the target to guarantee convergence were established in [201], and geometric ergodicity of the Gibbs sampler was further studied in [196].

The proof of convergence for Gaussian targets is taken from [200], which also contains fundamental results for the understanding of blocking and other variants of the Gibbs sampler. Gibbs samplers for hierarchical models have been widely studied, see e.g. [182] and references therein. Finally, adaptive Gibbs samplers are investigated in [48].

Chapter 6

Langevin Monte Carlo

This chapter introduces Langevin Monte Carlo, a family of sampling algorithms that shares numerous similarities with gradient descent optimization algorithms. Gradient descent minimizes an objective function $V : \mathbb{R}^d \to \mathbb{R}$ by taking steps in the direction of $-\nabla V$; similarly, Langevin Monte Carlo samples from a target distribution $f \propto \exp(-V)$ by moving, on average, in the direction of $-\nabla V$. In the same way that gradient descent can quickly find a local minimum of V, Langevin algorithms can quickly find and explore a mode of the target distribution. Consequently, gradient descent and Langevin Monte Carlo are well-suited for global optimization of convex objective functions and sampling of log-concave distributions. On the other hand, for non-convex objectives gradient descent can struggle to escape local minima, and for multimodal targets Langevin Monte Carlo may require many iterations to adequately explore the region around each mode.

We will study two Langevin sampling algorithms: the Unadjusted Langevin Algorithm (ULA) and the Metropolis Adjusted Langevin Algorithm (MALA). Our presentation of ULA highlights the analogy with gradient descent optimization. In a Gaussian setting, we will show that ULA is biased: the limit distribution of draws from ULA does not agree with the target. Nonetheless, this bias can be reduced by using a smaller step-size. MALA provides a natural approach to remove the bias of ULA by leveraging a Metropolis Hastings accept/reject mechanism. In addition to the optimization perspective on Langevin Monte Carlo algorithms that we emphasize in Sections 6.1 and 6.2, we will provide in Section 6.3 an alternative perspective deriving ULA by Euler-Maruyama discretization of the Langevin stochastic differential equation. This perspective predates the optimization one and motivates the name of the algorithms. As we shall see, the Langevin equation has the key property of leaving the target invariant; however, this property is no longer satisfied after discretization, explaining the bias of ULA. In this light, the Metropolis Hastings accept/reject step in MALA removes the Euler-Maruyama discretization error, ensuring convergence to the target. The chapter concludes in Section 6.4 with bibliographical remarks.

6.1 ▪ Unadjusted Langevin Algorithm

A simple but effective approach to minimize an objective function $V : \mathbb{R}^d \to \mathbb{R}$ is via the gradient descent algorithm, which starting from an initial guess $x_0 \in \mathbb{R}^d$ recursively defines

$$x_{n+1} = x_n - \epsilon \nabla V(x_n), \qquad n \geq 0. \tag{6.1}$$

Under convexity assumptions on V, the deterministic iterates $\{x_n\}_{n=1}^{\infty}$ converge to the minimizer of V provided that the step-size $\epsilon > 0$ is sufficiently small. The Unadjusted Langevin Algorithm

(ULA) relies on a stochastic variation of the deterministic update (6.1) to obtain approximate samples from a target distribution $f(x) \propto \exp\big(-V(x)\big)$, see Figure 6.1. Starting from a random variable $X^{(0)}$, draws $\{X^{(n)}\}_{n=1}^{\infty}$ from ULA satisfy

$$\mathbb{E}\big[X^{(n+1)}|X^{(n)}\big] = X^{(n)} - \epsilon\nabla V\big(X^{(n)}\big), \qquad n \geq 0.$$

Therefore, on average, ULA moves $X^{(n)} \mapsto X^{(n+1)}$ follow the direction of $-\nabla V\big(X^{(n)}\big)$, decreasing the value of V and increasing the value of the target $f \propto \exp(-V)$ provided that the step-size is sufficiently small. The method is summarized in Algorithm 6.1.

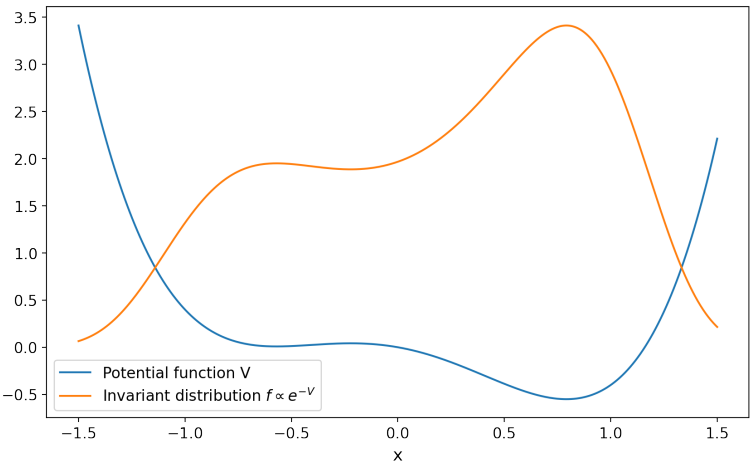


Figure 6.1. *Objective function* $V(x) = x^4 - x^2 - 0.4x$ *and corresponding target density* $f(x) \propto \exp\big(-V(x)\big)$. *Low values of* V *correspond to large values of* f.

Algorithm 6.1 Unadjusted Langevin Algorithm (ULA)

Input: Target $f(x) \propto \exp\big(-V(x)\big)$, initial distribution π_0, step-size $\epsilon > 0$, sample size N.

1: **Initial draw:** Sample $X^{(0)} \sim \pi_0$.

2: **Subsequent draws:** For $n = 0, 1, \ldots, N-1$ set

$$X^{(n+1)} = X^{(n)} - \epsilon\nabla V\big(X^{(n)}\big) + \sqrt{2\epsilon}\xi^{(n)}, \qquad \xi^{(n)} \sim \mathcal{N}(0, I).$$

Output: Sample $\{X^{(n)}\}_{n=1}^{N}$.

Pros and Cons: ULA can be implemented without knowing the normalizing constant of the target distribution. Moreover, by taking steps $X^{(n)} \mapsto X^{(n+1)}$ that are, on average, in the direction of $-\nabla V\big(X^{(n)}\big)$, ULA is able to quickly find and explore regions of high target density. Under convexity assumptions on V and for sufficiently small step-size ϵ, draws from ULA are, after a burn-in period, approximately distributed according to the target. However, the algorithm is biased in that, for fixed ϵ, the distribution of the n-th sample does not converge to the target in the large n asymptotic. Another caveat of ULA is that taking steps in the direction of $-\nabla V\big(X^{(n)}\big)$ can be counterproductive if the target is multi-modal. In that case, ULA can get stuck in one mode without adequately exploring others. Finally, ULA requires evaluating ∇V, which can be expensive.

We will analyze ULA for Gaussian target $f = \mathcal{N}(\mu, \Sigma)$. In this setting, similar to the Gibbs sampler in Chapter 5, draws from ULA satisfy a simple recursion. As in Chapter 5, we will denote by $H = \Sigma^{-1}$ the precision matrix.

Lemma 6.1. *Let $\{X^{(n)}\}_{n\geq 1}$ be the output of ULA with target $f = \mathcal{N}(\mu, \Sigma) = \mathcal{N}(\mu, H^{-1})$ and initial state $X^{(0)}$. It holds that*

$$X^{(n+1)} - \mu = (I - \epsilon H)^{n+1}(X^{(0)} - \mu) + \sqrt{2\epsilon}\sum_{i=0}^{n}(I - \epsilon H)^{n-i}\xi^{(i)}. \tag{6.2}$$

Suppose further that $\pi_0 = \mathcal{N}(\mu_0, \sigma_0^2 I)$ for some $\mu_0 \in \mathbb{R}^d$ and $\sigma_0^2 > 0$. Then, $X^{(n)} \sim \mathcal{N}(\mu_n, \Sigma_n)$, where

$$\mu_n = \mu + (I - \epsilon H)^n(\mu_0 - \mu), \tag{6.3}$$

$$\Sigma_n = \Big(H - \frac{\epsilon}{2}H^2\Big)^{-1} + (I - \epsilon H)^{2n}\bigg(\sigma_0^2 I - \Big(H - \frac{\epsilon}{2}H^2\Big)^{-1}\bigg). \tag{6.4}$$

Proof. Due to the Gaussian target assumption, we can set $V(x) = \frac{1}{2}(x-\mu)^T H(x-\mu)$, which implies that $\nabla V(x) = H(x-\mu)$. Therefore,

$$\begin{aligned} X^{(n+1)} - \mu &= X^{(n)} - \epsilon\nabla V(X^{(n)}) + \sqrt{2\epsilon}\,\xi^{(n)} - \mu \\ &= (I - \epsilon H)(X^{(n)} - \mu) + \sqrt{2\epsilon}\,\xi^{(n)} \\ &= (I - \epsilon H)^{n+1}(X^{(0)} - \mu) + \sqrt{2\epsilon}\sum_{i=0}^{n}(I - \epsilon H)^{n-i}\xi^{(i)}, \end{aligned}$$

thus establishing (6.2). Under the additional assumption that $\pi_0 = \mathcal{N}(\mu_0, \sigma_0^2 I)$, it follows from (6.2) that each $X^{(n)}$ has Gaussian distribution. To compute the mean, using that $\mathbb{E}[X^{(0)}] = \mu_0$ and that $\mathbb{E}[\xi^{(i)}] = 0$ for all $i \geq 0$,

$$\begin{aligned} \mu_n - \mu &= \mathbb{E}\Big[(I - \epsilon H)^n(X^{(0)} - \mu) + \sqrt{2\epsilon}\sum_{i=0}^{n-1}(I - \epsilon H)^{n-i-1}\xi^{(i)}\Big] \\ &= (I - \epsilon H)^n(\mu_0 - \mu), \end{aligned}$$

which proves (6.3). To compute the covariance, note first that

$$\begin{aligned} X^{(n)} - \mu_n &= (I - \epsilon H)^n(X^{(0)} - \mu) + \sqrt{2\epsilon}\sum_{i=0}^{n-1}(I - \epsilon H)^{n-i-1}\xi^{(i)} - (\mu_n - \mu) \\ &= (I - \epsilon H)^n(X^{(0)} - \mu_0) + \sqrt{2\epsilon}\sum_{i=0}^{n-1}(I - \epsilon H)^{n-i-1}\xi^{(i)}. \end{aligned}$$

Hence, using that $\mathbb{E}[(X^{(0)} - \mu_0)(X^{(0)} - \mu_0)^T] = \sigma_0^2 I$, that $\mathbb{E}[\xi^{(i)}(\xi^{(i)})^T] = I$, and the independence of $X^{(0)}$ and of each $\xi^{(i)}$ from all other randomness, we deduce that

$$\begin{aligned} \Sigma_n &= \mathbb{E}\big[(X^{(n)} - \mu_n)(X^{(n)} - \mu_n)^T\big] \\ &= \sigma_0^2(I - \epsilon H)^{2n} + 2\epsilon\sum_{i=0}^{n-1}(I - \epsilon H)^{2i} \\ &= \Big(H - \frac{\epsilon}{2}H^2\Big)^{-1} + (I - \epsilon H)^{2n}\bigg(\sigma_0^2 I - \Big(H - \frac{\epsilon}{2}H^2\Big)^{-1}\bigg), \end{aligned}$$

where in the last line we used the formula for the partial sum of the geometric series $\sum_{i=0}^{n-1} A^i = (I-A)^{-1} - A^n(I-A)^{-1}$. We have hence established (6.4), completing the proof. □

The key takeaway from Lemma 6.1 is that, for fixed and sufficiently small ϵ, we have

$$\mu_n \xrightarrow{n\to\infty} \mu,$$
$$\Sigma_n \xrightarrow{n\to\infty} \Sigma_\epsilon := \Big(H - \frac{\epsilon}{2}H^2\Big)^{-1}.$$

The next theorem shows that, in Wasserstein distance, $\pi_n := \mathcal{N}(\mu_n, \Sigma_n)$ converges exponentially to $f_\epsilon := \mathcal{N}(\mu, \Sigma_\epsilon)$ and that f_ϵ is order ϵ apart from f. We recall that the Wasserstein distance between Gaussians $\pi = \mathcal{N}(\mu, \Sigma)$ and $\widetilde{\pi} = \mathcal{N}(\widetilde{\mu}, \widetilde{\Sigma})$ is given by

$$W_2(\pi, \widetilde{\pi})^2 = \|\mu - \widetilde{\mu}\|^2 + \|\Sigma^{1/2} - \widetilde{\Sigma}^{1/2}\|_F^2$$

provided that Σ and $\widetilde{\Sigma}$ commute. Here $\|A\|_F = \sqrt{\mathrm{Tr}(A^T A)}$ denotes the Frobenius norm. We will denote by $\lambda_{\min}(H)$ and $\lambda_{\max}(H)$ the smallest and largest eigenvalues of H.

Theorem 6.2 (Convergence of ULA for Gaussian Targets). *Let $\{X^{(n)}\}_{n\geq 1}$ be the output of ULA with target $f = \mathcal{N}(\mu, \Sigma) = \mathcal{N}(\mu, H^{-1})$ and initial distribution $\pi_0 = \mathcal{N}(\mu_0, \sigma_0^2 I)$. Let π_n denote the distribution of $X^{(n)}$. Suppose that $\sigma_0^2 \leq \lambda_{\max}(H)^{-1}$. Then, there is a constant c which depends on μ_0, Σ_0, μ, and Σ such that*

$$W_2(\pi_n, f) \leq c\big(1 - \epsilon\lambda_{\min}(H)\big)^n + \frac{\epsilon}{4}\sqrt{\mathrm{Tr}(H)} + \mathcal{O}(\epsilon^2), \qquad n = 1, 2, \ldots$$

Proof. By triangle inequality, $W_2(\pi_n, f) \leq W_2(\pi_n, f_\epsilon) + W_2(f_\epsilon, f)$. The first term represents the distance between the distribution of $X^{(n)}$ and the limit distribution of ULA, and the second term represents the distance between the limit distribution of ULA and the target. We will bound both terms in turn. We denote by c a constant that depends on μ_0, Σ_0, μ, and Σ and which may change from line to line.

To bound $W_2(\pi_n, f_\epsilon)$, we need to control the deviation between the means and covariances of $\pi_n = \mathcal{N}(\mu_n, \Sigma_n)$ and $f_\epsilon = \mathcal{N}(\mu, \Sigma_\epsilon)$. For the means,

$$\begin{aligned}\|\mu_n - \mu\|^2 &= \|(I - \epsilon H)^n(\mu_0 - \mu)\|^2 \\ &\leq \max\{|1 - \epsilon\lambda_{\min}(H)|, |1 - \epsilon\lambda_{\max}(H)|\}^{2n}\|\mu_0 - \mu\|^2 \\ &= c\big(1 - \epsilon\lambda_{\min}(H)\big)^{2n},\end{aligned}$$

where we used the Cauchy-Schwarz inequality and that, for all ϵ sufficiently small, $\max\{|1 - \epsilon\lambda_{\min}(H)|, |1 - \epsilon\lambda_{\max}(H)|\} = |1 - \epsilon\lambda_{\min}(H)|$.

For the covariances, let $Q\mathrm{diag}(h_1, \ldots, h_d)Q^T$ be the eigendecomposition of H, where $\{h_i\}_{i=1}^d$ are the eigenvalues of H and Q is orthonormal. Then,

$$\Sigma_\epsilon = \Big(H - \frac{\epsilon}{2}H^2\Big)^{-1} = Q\mathrm{diag}\big(\alpha_1, \ldots, \alpha_d\big)Q^T,$$
$$\Sigma_n = Q\mathrm{diag}\big(\alpha_1 - \beta_1, \ldots, \alpha_d - \beta_d\big)Q^T,$$

where $\alpha_i = \Big(h_i - \frac{\epsilon}{2}h_i^2\Big)^{-1}$ and $\beta_i = (1 - \epsilon h_i)^{2n}(\alpha_i - \sigma_0^2)$. Notice that, for all sufficiently small ϵ, we have that $\frac{\epsilon}{2}h_i < 1$, and hence that $\alpha_i > 0$. Furthermore, the assumption on σ_0^2 ensures that

$\sigma_0^2 \leq \alpha_i$, and hence that $\beta_i \geq 0$. Consequently, for all ϵ sufficiently small,

$$\begin{aligned}\|\Sigma_n^{1/2} - \Sigma_\epsilon^{1/2}\|_F^2 &= \sum_{i=1}^d \Big((\alpha_i - \beta_i)^{1/2} - \alpha_i^{1/2}\Big)^2 = \sum_{i=1}^d \Big(\alpha_i - \beta_i + \alpha_i - 2\sqrt{\alpha_i(\alpha_i - \beta_i)}\Big) \\ &\leq \sum_{i=1}^d \Big(\alpha_i - \beta_i + \alpha_i - 2\sqrt{(\alpha_i - \beta_i)(\alpha_i - \beta_i)}\Big) = \sum_{i=1}^d \beta_i \\ &= \sum_{i=1}^d (1 - \epsilon h_i)^{2n}(\alpha_i - \sigma_0^2) \leq c\big(1 - \epsilon\lambda_{\min}(H)\big)^{2n},\end{aligned}$$

where for the last inequality we use that $\mathrm{Tr}(\Sigma_\epsilon - \Sigma_0) \to \mathrm{Tr}(\Sigma - \Sigma_0)$ as $\epsilon \to 0$, which implies that $\mathrm{Tr}(\Sigma_\epsilon - \Sigma_0)$ is uniformly bounded for all sufficiently small ϵ. Combining the bounds for the means and the covariances,

$$W_2(\pi_n, f_\epsilon)^2 \leq c\big(1 - \epsilon\lambda_{\min}(H)\big)^{2n}. \tag{6.5}$$

Next, we bound $W_2(f_\epsilon, f)$. Now the means agree, and we deduce as before that

$$\begin{aligned}W_2(f_\epsilon, f)^2 &= \sum_{i=1}^d \frac{1}{h_i}\left(1 - \frac{1}{\sqrt{1 - \frac{\epsilon}{2}h_i}}\right)^2 \\ &= \sum_{i=1}^d \frac{1}{h_i}\varphi\Big(\frac{\epsilon}{2}h_i\Big),\end{aligned}$$

where $\varphi(z) := \Big(1 - \frac{1}{\sqrt{1-z}}\Big)^2$. By Taylor expansion around zero, one can show that, for small z, $\varphi(z) = \frac{z^2}{4} + \mathcal{O}(z^3)$. Hence, for small ϵ,

$$W_2(f_\epsilon, f)^2 \leq \frac{\epsilon^2}{16}\sum_{i=1}^d h_i + \mathcal{O}(\epsilon^3) = \frac{\epsilon^2}{16}\mathrm{Tr}(H) + \mathcal{O}(\epsilon^3). \tag{6.6}$$

Taking square roots in (6.5) and (6.6) and using triangle inequality completes the proof. □

Choice of Step-Size The proof of Theorem 6.2 decomposes the Wasserstein distance between π_n and f into two terms. The first term, which corresponds to the distance between π_n and f_ϵ, decreases exponentially fast with the number n of iterations. The second term, which corresponds to the distance between f_ϵ and f, is a systematic bias of order ϵ. Notice that a smaller step-size reduces the bias, but it makes slower the convergence of π_n to the biased limit f_ϵ. Thus, it is recommended to take the largest ϵ such that the error in the approximation $f \approx f_\epsilon$ can be tolerated. While not considered here, time-varying step-sizes can also be used; in particular, a common approach is to take bold large steps for a few iterations to ensure fast exploration, and then gradually decrease the step-size to reduce the bias.

6.2 ▪ Metropolis Adjusted Langevin Algorithm

In the previous section, we have shown that the limit distribution of the Markov chain $\{X^{(n)}\}_{n=1}^N$ defined by ULA does not agree with the target, even in a simple Gaussian setting. In other words, the target $f \propto \exp(-V)$ is not an invariant distribution for the Markov kernel

$$q_{\text{Lgv}}(x, \cdot) = \mathcal{N}\big(x - \epsilon\nabla V(x), 2\epsilon I\big)$$

that defines the distribution of $X^{(n+1)}$ given that $X^{(n)} = x$. However, the proof of Theorem 6.2 suggests that, for small step-size ϵ, the limit distribution f_ϵ of ULA is order ϵ apart from f, so that the kernel q_{Lgv} *approximately* preserves the target. This motivates the idea of using q_{Lgv} as a proposal Markov kernel within the Metropolis Hastings algorithm studied in Chapter 4. The key advantage of this choice of proposal is that since q_{Lgv} approximately preserves the target, the accept/reject mechanism in the Metropolis Hastings framework only needs to account for a small correction to ensure convergence to the target. The resulting method, known as the Metropolis Adjusted Langeving Algorithm (MALA), is summarized below.

Algorithm 6.2 Metropolis Adjusted Langevin Algorithm (MALA)

Input: Target $f(x) \propto \exp\big(-V(x)\big)$, initial distribution π_0, step-size $\epsilon > 0$, sample size N.

1: Define the Markov kernel

$$q_{\text{Lgv}}(x, \cdot) := \mathcal{N}\big(x - \epsilon \nabla V(x), 2\epsilon I\big).$$

2: Run Metropolis Hastings (Algorithm 4.1) with inputs f, π_0, q_{Lgv}, N.

Output: Sample $\{X^{(n)}\}_{n=1}^N$.

Pros and Cons: MALA removes the bias of ULA at the price of introducing a Metropolis Hastings accept/reject step. MALA shares many of the pros and cons of ULA. They are both well suited to sample log-concave targets but can struggle to explore multi-modal distributions. Like ULA, MALA requires evaluating ∇V, which can be expensive. Compared to RWMH proposals from Chapter 4, MALA can scale better to high dimensional problems, particularly so for log-concave targets. However, tuning the step-size parameter ϵ is more delicate than for RWMH; existing theory shows that the step-size should be smaller in the transient than in the stationary phase of the chain.

Choice of Step-Size As we saw in Section 6.1, the limit distribution of ULA iterates depends on the choice of step-size. The ability of ULA to produce samples that are approximately distributed according to the target hinges on convexity of V and on choosing a small step-size. On the other hand, MALA corrects the bias of ULA via a Metropolis Hastings accept/reject mechanism. For MALA, using a larger step-size leads to more rejections but faster exploration of the state space. As for ULA, time-varying step-sizes can be used.

Classical theoretical results for MALA developed in the same large-d scaling setting that we considered for RWMH in Chapter 4 revealed two key insights.

1. The step-size ϵ should be taken to decrease with d at rate $d^{-1/3}$ in order to ensure that the acceptance rate remains bounded away from 0 and 1. Consequently, the number of iterations needed to get close to equilibrium is $\mathcal{O}(d^{1/3})$, a significant improvement over $\mathcal{O}(d)$ for RWMH. Though this fluctuates depending on the problem, many problems will have a cost per iteration of $\mathcal{O}(d)$ in both cases, so the overall complexity will be $\mathcal{O}(d^{4/3})$ compared to $\mathcal{O}(d^2)$.

2. The optimal acceptance rate for MALA is ≈ 0.574, compared to the considerably lower ≈ 0.234 for RWMH.

More recent results show enhanced scalability of MALA under various convexity assumptions on the potential V, see Section 6.4. A heuristic reason for the improvement of MALA over RWMH is that MALA, unlike RWMH, incorporates the target distribution in the proposal kernel through the gradient ∇V.

6.3 ▪ Langevin Equation and its Numerical Solution

In Section 6.1 we introduced ULA by analogy to the gradient descent optimization algorithm. Another insightful interpretation of ULA is to view it as a discretization of the Langevin stochastic differential equation. In fact, this perspective predates the optimization one and explains the name of the algorithm. Here, we summarize some key ideas without delving into the important technical issues that are needed for a rigorous treatment of this subject.

Given a potential $V : \mathbb{R}^d \to \mathbb{R}$, the (overdamped) Langevin equation is given by

$$dX(t) = -\nabla V\big(X(t)\big)\, dt + \sqrt{2}\, dW(t), \quad 0 \le t < \infty, \tag{6.7}$$

where W denotes a standard Brownian motion in $\mathbb{R}^d$. The solution to (6.7) is a continuous-time stochastic process $\{X(t)\}_{t\ge 0}$. Given a final time $T > 0$, a *trajectory* of this process is a random function

$$\begin{aligned} [0, T] &\to \mathbb{R}^d \\ t &\mapsto X(t). \end{aligned}$$

Figure 6.2 shows a trajectory of Langevin dynamics.

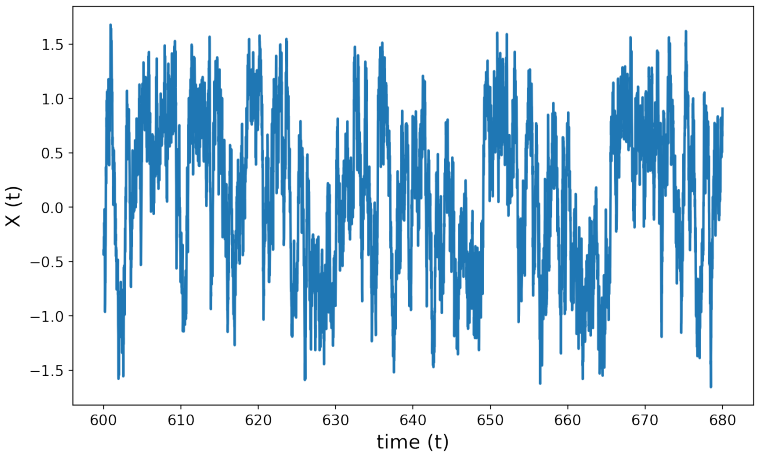


Figure 6.2. *Random trajectory of Langevin equation with the potential $V(x) = x^4 - x^2 - 0.4x$ shown in Figure 6.1. The process spends most of its time around the modes of the invariant distribution $f \propto \exp(-V)$.*

The p.d.f. $\pi_t(x) = \pi(x, t)$ of $X(t)$ defined by (6.7) satisfies the Fokker-Planck equation

$$\begin{aligned} \frac{\partial \pi}{\partial t} &= \operatorname{div}\big(\nabla V \pi + \nabla \pi\big), \\ \pi(x, 0) &= \pi_0(x). \end{aligned}$$

Notice that for $f \propto \exp(-V)$, we have that $\nabla V = -\nabla \log f = \frac{-\nabla f}{f}$, which implies that

$$\operatorname{div}\big(\nabla V f + \nabla f\big) = 0.$$

Thus, f is an invariant distribution for the Fokker-Planck equation, which means that if $X(0) \sim f$, then $X(t) \sim f$ for all $t \ge 0$. Moreover, the key property of Langevin equation that makes

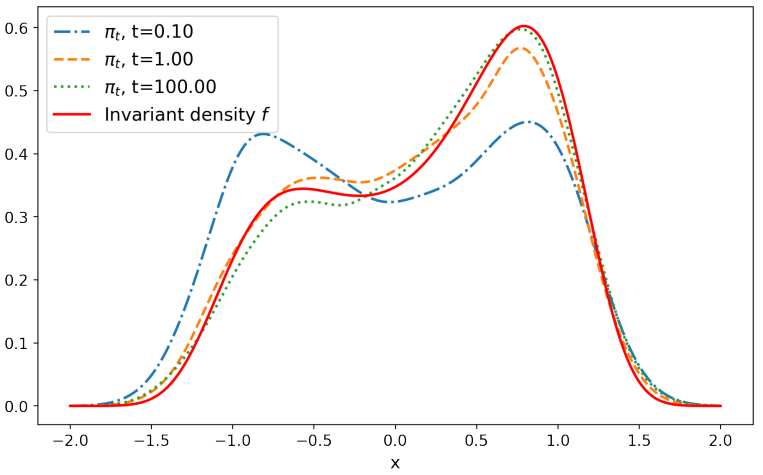


Figure 6.3. *Histograms of the distribution π_t of $X(t)$ for Langevin equation with the potential $V(x) = x^4 - x^2 - 0.4x$ in Figure 6.1. For large t, π_t is close to the target density $f \propto \exp(-V)$.*

it appealing for sampling is that, under mild asymptotic growth conditions on the potential V, it defines an ergodic process. This means that, for large t, $X(t)$ is approximately distributed like f, regardless of the initial distribution π_0 of $X(0)$. An illustration is given in Figure 6.3.

Although any given trajectory may spend large amounts of time hovering around some local minima of V, if T is large enough and $A \subset E = \mathbb{R}^d$, the proportion of time that the trajectory spends in A will approximately be $\int_A f(x)\,dx$. Such sample path ergodicity inspires a natural way to estimate expectations with respect to f :

$$I_f[h] := \int_E h(x)f(x)\,dx \approx \frac{1}{T}\int_0^T h\big(X(t)\big)\,dt, \tag{6.8}$$

where h is a given test function. The quality of the approximation of the spatial average via the temporal average along a trajectory improves as the end-time T increases.

In practice, Langevin equation (6.7) can only be solved analytically in simple cases, e.g. for quadratic potentials V that result in linear stochastic differential equations. Outside these cases, the Euler-Muruyama method is a standard approach to find numerical solutions. Let $0 < t_1 < \cdots < t_N = T$ with $t_n - t_{n-1} = \epsilon$, $n = 1, \ldots, N$. Applied to the Langevin equation (6.7), Euler-Maruyama approximates $X(t_n)$ with $X^{(n)}$, where

$$X^{(n+1)} = X^{(n)} - \epsilon\nabla V\big(X^{(n)}\big) + \sqrt{2\epsilon}\,\xi^{(n)}, \qquad \xi^{(n)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I).$$

In other words, ULA agrees with an Euler-Maruyama discretization of Langevin equation. In analogy with (6.8), one can then approximate expectations with respect to the target using the discretized process

$$I_f[h] := \int_E h(x)f(x)\,dx \approx \frac{1}{N}\sum_{n=1}^{N} h\big(X^{(n)}\big).$$

An important observation is that Euler-Maruyama does not preserve the ergodic property of Langevin equation, i.e. the invariant distribution for the chain $\{X^{(n)}\}_{n\geq 1}$ will no longer be f. However, we can use the Markov kernel

$$q_{\text{Lgv}}(x, \cdot) = \mathcal{N}\big(x + \epsilon\nabla \log f(x), 2\epsilon I\big)$$

implicitly defined by ULA as a proposal kernel within a Metropolis Hastings algorithm. The use of this proposal kernel along with its correction via the standard Metropolis Hastings acceptance probability

$$a(x,z) := \min\left\{1, \frac{f(z)}{f(x)} \frac{q_{\text{Lgv}}(z,x)}{q_{\text{Lgv}}(x,z)}\right\}$$

defines MALA. In this light, the Metropolis Hastings accept/reject step in MALA can be interpreted as removing the bias introduced by Euler-Maruyama discretization of Langevin equation.

6.4 ▪ Discussion and Bibliography

The idea of using continuous-time Langevin for sampling a given target distribution can be found in [20] and [98], but it was only developed in [202] where the MALA algorithm was first proposed and analyzed. This chapter emphasized the optimization viewpoint on Langevin Monte Carlo considered for instance in [78, 79, 236, 64]. These works establish convergence guarantees for Langevin equation and discretizations thereof under convexity assumptions akin to those required for convergence of gradient descent optimization [34]. The choice of step-size for MALA as well as its optimal scaling is discussed in [197]. Recent works show improvement in the scaling with dimension under appropriate convexity assumptions [139].

As for optimization algorithms, *preconditioning* Langevin Monte Carlo algorithms can speed-up their convergence. Preconditioning improves the conditioning of the target and may be implemented, for instance, by leveraging an ensemble of particles [137, 77, 44]. Relatedly, many algorithms leverage ideas from Riemannian geometry to adapt Langevin dynamics to the geometry induced by the target distribution [89, 183, 79].

An important caveat of Langevin Monte Carlo algorithms is the need to evaluate gradients, which in many applications is computationally expensive. For this reason, there has been significant interest in studying Langevin Monte Carlo algorithms that use approximate gradients [53]. In this direction, Stochastic Gradient Langevin Dynamics (SGLD) [235] provides an important extension to ULA and MALA. Consider the setting of Bayesian inference for a parameter θ under the assumption of i.i.d. data $\{y_i\}_{i=1}^K$ giving rise to a target posterior distribution of the form

$$f(\theta|y_1,\dots,y_K) \propto f(\theta)\prod_{i=1}^{K} f(y_i|\theta).$$

Evaluating the log-likelihood involves computing a sum over K terms, which is expensive for large K. This motivates subsampling $k \ll K$ data points uniformly at random at each iteration, which leads to the SGLD update rule

$$\theta^{(n+1)} = \theta^{(n)} + \epsilon_n\left\{\nabla \log\big(f(\theta^{(n)})\big) + \frac{K}{k}\sum_{i=1}^{k} \nabla \log\big(f(y_i^{(n)}|\theta^{(n)})\big)\right\} + \sqrt{2\epsilon_n}\,\xi^{(n)}.$$

Here $y_i^{(n)}$ denotes the i-th data point of the k subsampled points during the n-th iteration, and $\xi^{(n)} \sim \mathcal{N}(0,I)$. Further, the step-size ε_n is chosen such that $\sum_{n=1}^{\infty}\epsilon_n = \infty$ and $\sum_{n=1}^{\infty}\epsilon_n^2 < \infty$, ensuring that we can sample from this chain *without* the need for accepting and rejecting. Note that $\epsilon_n = 1/n$ satisfies the requirements on the step size. Another approach to design gradient-free Langevin Monte Carlo algorithms is to rely on an ensemble of interacting particles coupled by their empirical covariance to approximate gradients and precondition the dynamics [137, 77, 44].

Section 6.3 provided a brief and informal introduction to Langevin equation. For rigorous but accessible introductions to stochastic differential equations, we refer to the textbooks

[176, 72, 153], and for more details on Langevin equation and its use for sampling we refer to [184]. Numerical solutions are covered in [125, 111]. From a theoretical perspective, several works [138, 190] have shown that including a non-reversible linear drift term in Langevin equation speeds-up the convergence to the stationary distribution in continuous time; however, the non-reversible diffusion can be more expensive to discretize [79, 78], making less obvious the algorithmic value of this idea. A general recipe for building preconditioned non-reversible Langevin-type diffusions can be found in [150], while [151] discusses Langevin methods for sampling in hidden Markov models.

Chapter 7
Annealing Strategies

This chapter presents annealing strategies for optimization and sampling. In metallurgy, annealing is a technique involving heating and cooling of a material to alter its physical properties. In the same spirit, the annealing strategies in this chapter alter a given objective function or target distribution to facilitate optimization and sampling. For optimization, we will *cool* the objective so that it is more peaked around its global optima; for sampling, we will *heat* the target so that it is more flat and hence easier to sample from.

The first algorithm we will study is *simulated annealing*, a probabilistic technique for global optimization. Simulated annealing is particularly powerful when it is preferable to find an approximate global optimum rather than an exact local optimum. The main idea behind the algorithm is to sample a sequence of target distributions that gradually become more peaked around the global optimum. Sampling is typically performed via a Metropolis Hastings algorithm, and, consequently, simulated annealing can be applied to both discrete and continuous optimization. Indeed, simulated annealing does not require computing derivatives of the objective and has been applied to solve challenging large scale discrete optimization problems.

While for optimization we alter our objective so that it becomes more peaked, for sampling we will flatten the target distribution to facilitate exploration. The main motivation to do so is that sampling multi-modal distributions where regions of high probability are separated by regions of low probability can be challenging for Markov chain Monte Carlo algorithms that rely on local moves. We will study two strategies to leverage auxiliary heated target densities to speed-up exploration: simulated tempering and parallel tempering. *Simulated tempering* relies on an augmented target which includes an auxiliary variable that indexes temperatures of heated target distributions. On the other hand, *parallel tempering* —also known as replica exchange Markov chain Monte Carlo— runs in parallel several chains at different temperatures and relies on a Metropolis Hastings mechanism to propose swaps between the states of the chains. Augmenting the target to speed-up mixing and running multiple chains that communicate with each other are two important ideas that underpin many sampling algorithms and convergence diagnostics.

This chapter is organized as follows. Section 7.1 introduces annealing strategies for optimization, leading to the simulated annealing algorithm. Section 7.2 describes strategies for sampling, with a focus on simulated tempering and parallel tempering. Section 7.3 closes with bibliographical remarks.

7.1 ▪ Annealing Strategies for Optimization

In this section, we first use sampling to find the mode of a given target distribution and then introduce the simulated annealing optimization algorithm to minimize a given objective function.

7.1.1 ▪ Finding the Mode of a Distribution

Let f be the distribution of interest. The mode of f is the set $\mathcal{M} = \{\xi : f(\xi) \geq f(x) \text{ for all } x \in E\}$ of *global* maxima of f.[5] Naively, one can approximate the mode of f by sampling f and choosing the draw with higher density with highest density:

1: Generate $X^{(n)} \sim f, \quad 1 \leq n \leq N$.
2: Output $\text{argmax}_{1 \leq n \leq N} f(X^{(n)})$.

This approach is not very efficient. We would like to have a higher density of samples close to the mode, rather than samples distributed like f. This motivates us to modify the distribution f into a new distribution with the same mode as f but higher density around it. One way to achieve this goal is to consider

$$f_\beta(x) \propto f(x)^\beta \tag{7.1}$$

for large values of $\beta > 0$. The new distribution f_β is more peaked than f and has the same mode. We will tacitly assume without further notice that $\int f(x)^\beta \, dx < \infty$, so that $f(x)^\beta$ can be normalized into a p.d.f.

Example 7.1 (Gaussian Target). Consider a Gaussian density $f = \mathcal{N}(\mu, \sigma^2)$ with mode μ. In this case

$$f_\beta(x) \propto \exp\left(-\frac{(x-\mu)^2}{2\sigma^2/\beta}\right),$$

which shows that $f_\beta(x)$ is a Gaussian density $\mathcal{N}(\mu, \sigma^2/\beta)$ with the same mode μ as f. The larger β is chosen, the smaller the variance, and the more concentrated f_β is around its mode. □

Example 7.2 (Interpretation of β as Inverse Temperature). Consider as in Chapter 6 an unnormalized target density of the form $f(x) \propto \exp\big(-V(x)\big)$, where $V : \mathbb{R}^d \to \mathbb{R}$ is a given potential function. Then, $f_\beta(x) \propto \exp\big(-\beta V(x)\big)$. Following the ideas in Chapter 6, the Langevin equation

$$dX(t) = -\nabla V\big(X(t)\big)\, dt + \sqrt{2\beta^{-1}}\, dW(t), \;\; 0 \leq t < \infty, \tag{7.2}$$

has f_β as invariant distribution under mild assumptions on V. The random effect on the trajectories stemming from the white noise term dW is more pronounced when β^{-1} is large. On the other hand, for large β the term $\sqrt{2\beta^{-1}}\, dW(t)$ is small and the Langevin dynamics resembles the deterministic dynamics of the gradient system $\dot{x} = -\nabla V(x)$. In physical applications, β is interpreted as the *inverse temperature* of the system due to the connection between temperature and kinetic energy of the system by which particles move faster when the temperature of the system is higher. □

We can use any of the sampling algorithms considered in previous chapters to sample f_β. However, sampling f_β typically becomes harder for larger β. Consider for instance the Random

[5] We are mainly interested in cases where the mode comprises a single element $\mathcal{M} = \{\xi\}$, in which case we abuse our terminology and refer to the element ξ rather than the set $\{\xi\}$ as the mode.

Walk Metropolis Hastings in Algorithm 4.3 with a proposal distribution g symmetric around 0. The probability of accepting a move from $X^{(n)}$ to Z^* would be

$$\min\left\{1, \frac{f_\beta(Z^*)}{f_\beta(X^{(n)})}\right\} = \min\left\{1, \left(\frac{f(Z^*)}{f(X^{(n)})}\right)^\beta\right\}, \tag{7.3}$$

which does not depend on the (usually unknown) normalization constant of f_β. It is however difficult to sample from f_β for large values of β. For $\beta \to \infty$ the probability of accepting a newly proposed Z^* becomes 1 if $f(Z^*) > f(X^{(n)})$ and 0 otherwise. For this reason, a typical scenario is that $X^{(n)}$ converges to *local* extrema of f, not necessarily a mode (global extrema).

Example 7.3 (Discrete Target and Convergence to Local Maxima). Consider the p.m.f. $f(x)$ supported on $\{1, 2, \ldots, 5\}$ defined by

$$f(x) = \begin{cases} 0.4 & x = 2, \\ 0.3 & x = 4, \\ 0.1 & x = 1, 3, 5. \end{cases}$$

and shown in Figure 7.3. Clearly the mode is $x = 2$. Assume we sample $f_\beta(x) \propto f(x)^\beta$ with a random walk proposal defined as follows:

$$Z^* = X^{(n)} + \xi^*,$$

where

$$\begin{aligned} &\mathbb{P}(\xi^* = \pm 1) = 0.5 \quad \text{if } X^{(n)} \in \{2, 3, 4\}, \\ &\mathbb{P}(\xi^* = -1) = 1 \quad \text{if } X^{(n)} = 5, \\ &\mathbb{P}(\xi^* = +1) = 1 \quad \text{if } X^{(n)} = 1. \end{aligned}$$

Note that for $\beta \to \infty$ the probability of accepting a move from 4 to 3 converges to 0 since $f(4) > f(3)$. As the Markov chain can only move from 4 to 2 via 3, it cannot escape the local mode at $x = 4$ for $\beta \to \infty$. □

The above discussion suggests that for large β the distribution f_β is concentrated around the mode but is hard to sample from. This motivates considering a time-changing sequence of target distributions f_{β_n} with an increasing sequence $\{\beta_n\}_{n=0}^\infty$ of inverse temperatures. This is the key idea behind the simulated annealing algorithm we study next.

7.1.2 ▪ Minimizing an Objective Function

We now turn to the problem of finding the global minimum of an objective function $V : E \to \mathbb{R}$. Equivalently,[6] we seek to find the mode of the distribution

$$f(x) \propto \exp\bigl(-V(x)\bigr).$$

As in Subsection 7.1.1, we can raise f to different powers to obtain

$$f_{\beta_n}(x) \propto \exp\bigl(-\beta_n V(x)\bigr),$$

where $\{\beta_n\}_{n=1}^N$ is a given sequence of inverse temperatures. The simulated annealing algorithm proceeds as follows:

[6] This equivalence makes the implicit assumption that $\int \exp\bigl(-V(x)\bigr)\, dx < \infty$, so that $\exp\bigl(-V(x)\bigr)$ can be normalized into a p.d.f.

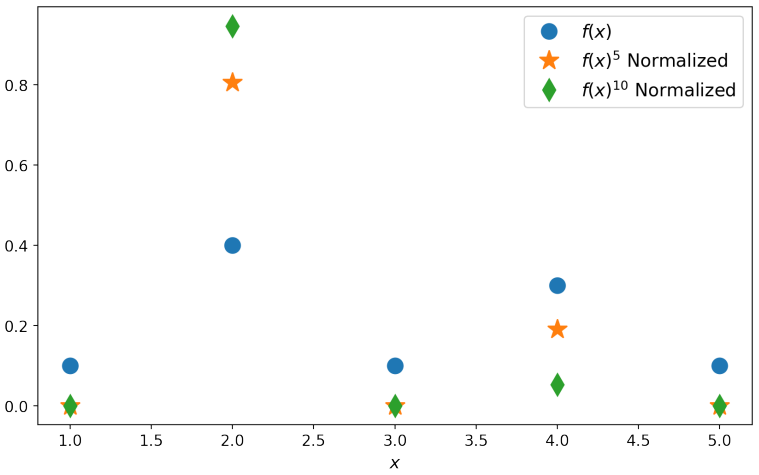


Figure 7.1. *An illustration of the target p.m.f. in Example 7.3 along with* $f_\beta(x) \propto f(x)^\beta$ *with* $\beta = 5, 10$.

Algorithm 7.1 Simulated Annealing

Input: Function to minimize V, proposal Markov kernel $q(x, z)$, initialization $X^{(0)}$, sample size N, sequence of inverse temperatures $\{\beta_n\}_{n=1}^N$.

1: For $n = 0, 1, \ldots, N-1$ do:
2: Draw $Z^* \sim q(X^{(n)}, \cdot)$.
3: Update

$$X^{(n+1)} = \begin{cases} Z^* & \text{with probability } a := \min\left\{1, \frac{\exp(-\beta_{n+1}V(Z^*))}{\exp\left(-\beta_{n+1}V(X^{(n)})\right)} \frac{q(Z^*, X^{(n)})}{q(X^{(n)}, Z^*)}\right\}, \\ X^{(n)} & \text{with probability } 1-a. \end{cases}$$

4: **Output**: Sample $\{X^{(n)}\}_{n=1}^N$ that can be used to minimize V by $\text{argmin}_{1 \le n \le N} V(X^{(n)})$.

First and foremost, simulated annealing is a highly versatile optimization algorithm that can be applied for discrete and continuous optimization without requiring first or second-order derivatives of the objective. Notice that the accept/reject mechanism agrees with that in the Metropolis Hastings framework in Algorithm 4.1, taking f_{β_n} to be the target distribution for the n-th draw. As in the Metropolis Hastings framework, there is freedom in the choice of proposal Markov kernel. Since a key feature of simulated annealing is its potential for derivative-free implementation, random walk proposal kernels that do not require gradient information on V have been traditionally favored over the Langevin proposal kernels considered in Chapter 6. If a RWMH with symmetric proposal is used, i.e. $Z^* = X^{(n)} + \xi^*$ with the distribution of ξ^* being symmetric around the origin, then the acceptance probability simplifies to

$$a(X^{(n)}, Z^*) = \min\left\{1, \exp\left(-\beta_n\left(V(Z^*) - V(X^{(n)})\right)\right)\right\},$$

giving the standard implementation of simulated annealing.

Choice of Inverse Temperatures The convergence analysis of simulated annealing is challenging. The algorithm usually converges to a local minimum, but whether it converges to a

global one depends on the choice of inverse temperatures. Two types of tempering have been widely studied:

- **Logarithmic tempering:** For given $\beta_0 > 0$ set $\beta_n := \frac{\log(1+n)}{\beta_0}$, $n \geq 1$. The inverse temperatures increase slowly enough that global convergence can be established for certain special cases. Simulated annealing with logarithmic tempering behaves similarly to an exhaustive search, and so while it satisfies nice theoretical guarantees, it is typically slow to converge and has little practical relevance.
- **Geometric tempering:** $\beta_n = \alpha^n \beta_0$ for some $\beta_0 > 0$ and $\alpha > 1$.

Pros and Cons: Simulated annealing can be an effective discrete or continuous optimization algorithm when convexity or smoothness assumptions invoked by first and second-order deterministic optimization algorithms are not satisfied. Importantly, simulated annealing does not require derivatives and can be applied in high dimension. The algorithm converges under mild assumptions when logarithmic tempering is used, but in practice it is usually implemented with geometric tempering, for which few theoretical guarantees are available. Additionally, the choice of the temperatures can have a significant impact on the behavior of the algorithm.

7.2 ▪ Annealing Strategies for Sampling

Sampling multi-modal target distributions that have regions of high probability separated by regions of low probability can be challenging. Annealing strategies seek to improve sampling by introducing auxiliary *tempered* distributions that bridge regions of low probability.

7.2.1 ▪ Simulated Tempering

Let $\beta_1 = 1 > \beta_2 > \cdots > \beta_K$ be K *decreasing* inverse temperatures and let $f_{\beta_k}(x) \propto f(x)^{\beta_k}$. Here $f = f_{\beta_1}$ is the target distribution of interest and, for $1 \leq k \leq K$, the density f_{β_k} is a *flattened* version of f, which is easier to sample. Simulated tempering considers an augmented target distribution

$$f^{\text{ST}}(x, k) \propto c_k f_{\beta_k}(x)$$

over the variable of interest x and the discrete auxiliary variable k which indexes the inverse temperatures. Samples from the distribution of interest f can then be obtained by keeping the x-variable of the samples $(X^{(n)}, k^{(n)})$ with $k^{(n)} = 1$.

A simple way to design a Markov chain Monte Carlo algorithm for the extended distribution f^{ST} is by alternating between updates of the x and k components. Moves for the latter can be proposed using the simple kernel

$$\begin{aligned} r(k, k+1) = r(k, k-1) &= \frac{1}{2}, \qquad k \in \{2, \ldots, K-1\}, \\ r(K, K-1) = r(1, 2) &= 1. \end{aligned}$$

The simulated tempering algorithm then proceeds as follows:

Algorithm 7.2 Simulated Tempering

Input: Inverse temperatures $\beta_1 = 1 > \cdots > \beta_K$, proposal Markov kernel $q(x, z)$, proposal Markov kernel $r(k, \ell)$, initialization $(X^{(0)}, q^{(0)})$, sample size N.

1: For $n = 0, 1, \ldots, N-1$ do:

2: Draw $\ell^* \sim r(k^{(n)}, \cdot)$.

3: Update

$$k^{(n+1)} = \begin{cases} \ell^* & \text{with probability } a_{k,\ell} := \min\left\{1, \frac{f^{\mathrm{ST}}(X^{(n)}, \ell^*)}{f^{\mathrm{ST}}(X^{(n)}, k^{(n)})} \frac{r(\ell^*, k^{(n)})}{r(k^{(n)}, \ell^*)}\right\}, \\ k^{(n)} & \text{with probability } 1 - a_{k,\ell}. \end{cases}$$

4: Draw $Z^* \sim q(X^{(n)}, \cdot)$.

5: Update

$$X^{(n+1)} = \begin{cases} Z^* & \text{with probability } a_{X,Z} := \min\left\{1, \frac{f^{\mathrm{ST}}(Z^*, k^{(n+1)})}{f^{\mathrm{ST}}(X^{(n)}, k^{(n+1)})} \frac{q(Z^*, X^{(n)})}{q(X^{(n)}, Z^*)}\right\}, \\ X^{(n)} & \text{with probability } 1 - a_{X,Z}. \end{cases}$$

6: **Output**: Sample $\{(X^{(n)}, k^{(n)})\}_{n=1}^{N}$.

The idea is that for low inverse temperature the chain will easily move across the state space, and by the time the augmented chain returns to the temperature of interest $\beta_1 = 1$ the x variable will have moved to a different region of the state space, accelerating the mixing.

Pros and Cons: Simulated tempering faces two implementation issues. First, how should we choose the constants c_k? It is generally recommended that the temperatures are chosen so that the extended chain spends roughly the same amount of time at each temperature. If all the f_{β_k} can be normalized, then the constants c_k should then be chosen equal. However, in practice these normalizing constants are typically unknown and need to be estimated using an alternative adaptive algorithm (e.g. Wang-Landau). The second implementation issue is how to choose the inverse temperatures. There is a trade-off between having sufficient acceptance probability for moving from one temperature to another, and not having too many temperature levels. There is a vast literature on this issue, notably in physics.

7.2.2 ▪ Parallel Tempering

Parallel tempering also uses a decreasing sequence of inverse temperatures β_k. However, it runs K chains in parallel such that the k-th chain has stationary distribution f_{β_k}. The chains are allowed to communicate by a swapping mechanism, so that chains with larger inverse temperatures benefit from the enhanced mixing of the chains with lower inverse temperature. Precisely, after an updating each of the K chains, we pick a pair of chains and propose to swap their states. This proposed swapping is followed by an accept/reject step to retain the correct invariant distribution. As in simulated tempering, we set $\beta_1 = 1$ so that $f_{\beta_1} = f$. We denote by $X_k^{(n)}$ the n-th sample from the k-th chain.

Algorithm 7.3 Parallel Tempering

Input: Inverse temperatures $\beta_1 = 1 > \cdots > \beta_K$, proposal Markov kernels $\{q_k(x,z)\}_{k=1}^K$, initializations $\{X_k^{(0)}\}_{k=1}^K$, sample size N.

For $n = 0, 1, \ldots, N-1$ do:

For $k = 1, \ldots, K$: generate $\tilde{X}_k^{(n+1)}$ by doing a Metropolis Hastings step (including accept/reject) with current state $X_k^{(n)}$, proposal kernel q_k, and target f_{β_k}.

Choose $\ell, m \in \{1, \ldots, K\}$ with $\ell \neq m$ uniformly at random.

For $k \notin \{\ell, m\}$ set $X_k^{(n+1)} = \tilde{X}_k^{(n+1)}$.

Attempt a swap of states between the ℓ-th and the m-th chains:

$$\left(X_\ell^{(n+1)}, X_m^{(n+1)}\right) = \begin{cases} \left(\tilde{X}_m^{(n+1)}, \tilde{X}_\ell^{(n+1)}\right) & \text{with probability } a_{\ell,m}, \\ \left(\tilde{X}_\ell^{(n+1)}, \tilde{X}_m^{(n+1)}\right) & \text{with probability } 1 - a_{\ell,m}, \end{cases}$$

where

$$a_{\ell,m} := \min\left\{1, \frac{f_{\beta_\ell}\left(\tilde{X}_m^{(n+1)}\right) f_{\beta_m}\left(\tilde{X}_\ell^{(n+1)}\right)}{f_{\beta_m}\left(\tilde{X}_m^{(n+1)}\right) f_{\beta_\ell}\left(\tilde{X}_\ell^{(n+1)}\right)}\right\}.$$

Output: Sample $= \{X_1^{(n)}\}_{n=1}^N$.

Parallel tempering requires proposal Markov kernels $\{q_k\}_{k=1}^K$ to update each chain. A popular choice is to use the Langevin proposals described in Chapter 6. Precisely, for the k-th chain one can define a proposal Markov kernel q_k by proposing moves $X_k^{(n)} \mapsto Z_k^*$ according to

$$Z_k^* = X_k^{(n)} + \epsilon_k \nabla \log f\left(X_k^{(n)}\right) + \sqrt{2\epsilon_k \beta_k^{-1}}\xi_k^*,$$

where $\xi_k^* \sim \mathcal{N}(0, I)$ and $\epsilon_k > 0$ is a given step size. Notice that large β_k^{-1} results in large random moves that can help explore the state space. Such moves may still be accepted with high probability, since they are targeting the flattened distribution f_{β_k}.

Pros and Cons: In contrast to simulated tempering, parallel tempering does not require estimating the normalizing constants c_k. While parallel tempering involves running K chains in parallel, doing so is not a significant obstacle with modern parallel computing. The question of how to choose the number K of chains and the temperatures β_k has been widely studied, and there is some evidence that geometric scaling of the temperatures may be optimal under mild assumptions.

7.3 ▪ Discussion and Bibliography

The simulated annealing algorithm was introduced in [124]. Geman and Geman [82] conjectured and Gidas [87] rigorously showed convergence with logarithmic tempering in finite state space. Later, Hajek [103] showed that choosing the inverse temperatures via logarithmic tempering is a sufficient and necessary condition for global convergence. Further early results were reviewed and developed in [227, 220, 1, 2]. For more recent expositions, we refer to [109, 149].

Simulated and parallel tempering were introduced in [219, 154], and further developed in [85, 114]. For a review on parallel tempering, we refer to [66]. The question of how to choose the temperatures has been widely studied, and recent theory suggests that a geometric sequence of temperature ratios is optimal under mild assumptions [63]. Another important question is

how often one should propose swaps between chains. In a continuous time setting, recent theory indicates the potential advantage of frequent swaps, formalized via the infinite swapping limit for parallel tempering [62].

Augmenting the target to speed-up mixing and running multiple chains that communicate with each other are two important ideas that underpin many sampling algorithms and convergence diagnostics (see e.g. Hamiltonian Monte Carlo in Chapter 8, Box-Muller sampling method in Chapter 3, slice sampler in Chapter 5, Gelman and Rubin's diagnostic in Chapter 4). Likewise, introducing a sequence of auxiliary target densities to gradually reach a desired target is a powerful idea that underpins many sequential Monte Carlo algorithms (see particle filters in Chapter 9). Algorithms that rely on tempered transitions for sampling multi-modal distributions are discussed in [168].

Chapter 8

Hamiltonian Monte Carlo

The key idea of Hamiltonian Monte Carlo (HMC) is to extend the state space and exploit some properties of Hamiltonian dynamics to avoid the local behavior of RWMH and Langevin Monte Carlo. We write our target as

$$f(q) \propto \exp\big(-V(q)\big), \qquad q \in \mathbb{R}^d,$$

and introduce a distribution f^H in extended space $\mathbb{R}^{2d}$

$$\begin{aligned} f^H(q,p) &\coloneqq \frac{1}{Z}\exp\big(-V(q)\big)\exp\big(-K(p)\big) \\ &= \frac{1}{Z}\exp\big(-H(q,p)\big), \end{aligned} \tag{8.1}$$

where $K(p)$ satisfies $\int_{\mathbb{R}^d}\exp\big(-K(p)\big)\,dp < \infty$ and

$$H(q,p) := V(q) + K(p)$$

is called the Hamiltonian. We call the measure μ^H with Lebesgue density f^H the canonical measure of the Hamiltonian H. All the ingredients just introduced have both a physical and a statistical interpretation.

- $q \in \mathbb{R}^d$ is interpreted as position $\equiv$ variables of interest.
- $p \in \mathbb{R}^d$ is interpreted as momentum $\equiv$ artificial variables that are not part of our statistical problem.
- V is interpreted as potential energy, and is determined by the unextended target f.
- K is interpreted as kinetic energy, and in HMC is part of the algorithmic design, a standard choice being $K(p) = \frac{1}{2}p^T M^{-1} p$ for some symmetric positive definite $M \in \mathbb{R}^{d\times d}$.
- μ^H is interpreted as a Gibbs measure that governs the distribution of (q,p) over an ensemble of many copies of the system in contact with a heat bath at a constant temperature.

Clearly f is the marginal of f^H over the position variables, and therefore we can obtain samples from f by sampling f^H and throwing away the momentum variables. At first sight, it is not clear how sampling f^H rather than directly sampling f can help. The main idea is that extending the state space allows us to exploit several properties of Hamiltonian dynamics, proposing bolder moves in state space while not hindering the acceptance rate and the mixing of the chain.

This chapter is organized as follows. Hamiltonian dynamics are reviewed in Section 8.1, emphasizing three properties that are important to the design of sampling algorithms. The HMC algorithm and an idealized version of it are introduced in Section 8.2. We focus on showing that these algorithms are exact, meaning that they leave the target distribution invariant. Section 8.3 contains bibliographical remarks, including references that address the geometric ergodicity of HMC.

8.1 ▪ Hamiltonian Dynamics

Given a Hamiltonian $H(q,p)$, the corresponding Hamilton equations are

$$\begin{aligned} \frac{dq_i}{dt} &= \frac{\partial H}{\partial p_i}, & 1 \le i \le d, \\ \frac{dp_i}{dt} &= -\frac{\partial H}{\partial q_i}, & 1 \le i \le d. \end{aligned} \tag{8.2}$$

We vectorize equations (8.2) as follows

$$\frac{dx}{dt} = J^{-1}\nabla H(x), \qquad J := \begin{bmatrix} 0 & -I_{d\times d} \\ I_{d\times d} & 0 \end{bmatrix} \in \mathbb{R}^{2d\times 2d}, \quad x := \begin{bmatrix} q \\ p \end{bmatrix} \in \mathbb{R}^{2d},$$

where

$$\nabla H = \left[\frac{\partial H}{\partial q_1}, \dots, \frac{\partial H}{\partial q_d}, \frac{\partial H}{\partial p_1}, \dots, \frac{\partial H}{\partial p_d}\right]^T, \qquad \frac{d}{dt}\begin{bmatrix} q \\ p \end{bmatrix} = \left[\frac{dq_1}{dt}, \dots, \frac{dq_d}{dt}, \frac{dp_1}{dt}, \dots, \frac{dp_d}{dt}\right]^T.$$

The matrix J plays a central role in the Hamiltonian formalism. Note that J is skew-symmetric, meaning that $J^T = -J$. Later we will use the following properties of invertible skew-symmetric matrices:

1. J^{-1} is also skew-symmetric, since $J^{-1} = -(J^T)^{-1} = -(J^{-1})^T$.
2. The corresponding quadratic form is zero, since $x^T Jx = x^T J^T x = -x^T Jx$.

Hamilton equations (8.2) define a flow in phase space: For $t \ge 0$, we denote by $\varphi_t : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ the t-flow map, so that $\varphi_t(x) \in \mathbb{R}^{2d}$ is the solution to Hamilton equations at time t with initial condition $x \in \mathbb{R}^{2d}$. It follows directly from the definition that $\varphi_0(x) = x$, i.e. φ_0 is the identity map. Furthermore, for $t, s \ge 0$ we have that $\varphi_{s+t}(x) = \varphi_s \circ \varphi_t(x)$, since the solution at time $t+s$ with initial condition x can be found by first solving up to time t to obtain $x(t) := \varphi_t(x)$, and then solving Hamilton equations for an additional time s with initial condition $x(t)$ to obtain $x(t+s) := \varphi_s\big(x(t)\big) = \varphi_s \circ \varphi_t(x)$. The concept of flow map of a dynamical system is helpful in exploring the qualitative behavior of numerical methods to solve them, since one can think of numerical methods as approximations of the flow map. In particular, we will see that in order to ensure that Hamiltonian Monte Carlo algorithms leave the extended target invariant, it is essential to use numerical solvers that preserve key properties of the flow map.

Example 8.1 (Harmonic Oscillator). Consider the case $d = 1$, $V(q) = \frac{q^2}{2}$, and $K(p) = \frac{p^2}{2}$. Then, the Hamilton equations are

$$\begin{aligned} \frac{dq}{dt} &= p, \\ \frac{dp}{dt} &= -q. \end{aligned}$$

The general solutions have the form

$$q(t) = r\cos(a+t),$$
$$p(t) = -r\sin(a+t),$$

for some constants a and r, which shows that φ_t is a clockwise rotation in the $q-p$ plane. □

Hamiltonian dynamics dovetail nicely with MCMC because they satisfy three important properties: conservation of the Hamiltonian, symplecticity, and reversibility. We next explain these properties and how they are relevant to the construction of MCMC algorithms.

8.1.1 ▪ Conservation of Hamiltonian

By the chain rule and the fact that J is skew-symmetric, we have that

$$\frac{dH}{dt} = \nabla H^T \frac{dx}{dt} = \nabla H^T J^{-1} \nabla H = 0, \qquad x = \begin{bmatrix} q \\ p \end{bmatrix}.$$

Recall the definition of f^H in (8.1). The value of f^H depends on $x = (q,p)$ only through H. Therefore, conservation of the Hamiltonian implies that if we start at any $x_0 = (q_0, p_0)$ and move with Hamiltonian dynamics, we move along level sets of f^H. In other words, we have $H(x_0) = H\big(\varphi_s(x_0)\big)$ for all $s > 0$. This property, together with the preservation of the

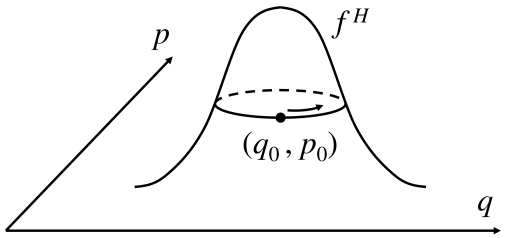


Figure 8.1. *Movement along a level set of f^H starting from (q_0, p_0).*

canonical measure that we describe next, will allow us to propose moves that are far but likely to be accepted.

8.1.2 ▪ Symplecticity and Preservation of the Canonical Measure

The flow map $x(s) \overset{\varphi_t}{\longmapsto} x(t+s)$ preserves volume. This is a direct consequence of the symplecticity of Hamiltonian dynamics. We will provide the mathematical definition of symplecticity and prove the symplecticity of Hamiltonian dynamics using a theorem by Poincaré.

Definition 8.2 (Symplectic Maps). *A linear map $L : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ is symplectic if $L^T J L = J$. A continuously differentiable map $\psi : U \subset \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ is symplectic if its Jacobian is symplectic everywhere. That is, if*

$$\psi'(x)^T J \psi'(x) = J, \qquad \forall x \in \mathbb{R}^{2d}.$$ □

Remark 8.3. *In $\mathbb{R}^2$ symplecticity of linear maps is equivalent to area preservation of parallelograms; in $\mathbb{R}^{2d}$ it is equivalent to conservation of the oriented areas of the projections of parallelograms into the coordinate planes (q_i, p_i).*

Theorem 8.4 (Hamiltonian Flows are Symplectic). *Let H be a $\mathcal{C}^2$ Hamiltonian. Then the corresponding flow map φ_t is symplectic for all t.*

Proof. Let $\varphi(x,t) := \varphi_t(x)$ denote the flow map of time t starting from x. We need to show that, for any t and arbitrary $x \in \mathbb{R}^{2d}$,

$$\frac{\partial}{\partial x}\varphi(x,t)^T J \frac{\partial}{\partial x}\varphi(x,t) = J. \tag{8.3}$$

First, the equality holds for $t = 0$, since $\varphi(x,0)$ is the identity map on $\mathbb{R}^{2d}$. Therefore, the proof will be complete if we show that the left-hand side in equation (8.3) is constant in time. To that end, recall that $\frac{\partial}{\partial t}\varphi(x,t) = J^{-1}\nabla H\big(\varphi(x,t)\big)$ and so, using that H is twice continuously differentiable, we have

$$\begin{aligned}
\frac{\partial^2}{\partial t \partial x}\varphi(x,t) &= \frac{\partial^2}{\partial x \partial t}\varphi(x,t) \\
&= \frac{\partial}{\partial x} J^{-1}\nabla H\big(\varphi(x,t)\big) \\
&= J^{-1}\nabla^2 H\big(\varphi(x,t)\big)\frac{\partial}{\partial x}\varphi(x,t),
\end{aligned}$$

where ∇^2 denotes the Hessian. This partial result and the skew-symmetry of J give that

$$\begin{aligned}
&\frac{\partial}{\partial t}\left(\frac{\partial}{\partial x}\varphi(x,t)^T J \frac{\partial}{\partial x}\varphi(x,t)\right) \\
&= \frac{\partial^2}{\partial t \partial x}\varphi(x,t)^T J \frac{\partial}{\partial x}\varphi(x,t) + \frac{\partial}{\partial x}\varphi(x,t)^T J \frac{\partial}{\partial t \partial x}\varphi(x,t) \\
&= \frac{\partial}{\partial x}\varphi(x,t)^T \nabla^2 H\big(\varphi(x,t)\big) J^{-T} J \frac{\partial}{\partial x}\varphi(x,t) + \frac{\partial}{\partial x}\varphi(x,t)^T J J^{-1}\nabla^2 H\big(\varphi(x,t)\big)\frac{\partial}{\partial x}\varphi(x,t) \\
&= -\frac{\partial}{\partial x}\varphi(x,t)^T \nabla^2 H\big(\varphi(x,t)\big)\frac{\partial}{\partial x}\varphi(x,t) + \frac{\partial}{\partial x}\varphi(x,t)^T \nabla^2 H\big(\varphi(x,t)\big)\frac{\partial}{\partial x}\varphi(x,t) \\
&= 0,
\end{aligned}$$

and the proof is complete. □

Taking the determinant at both sides of the equality

$$\big(\varphi_t'\big)^T J \varphi_t = J$$

shows that the Jacobian determinant of φ_t has absolute value 1. Using this property and the conservation of the Hamiltonian, we obtain the following important result:

Theorem 8.5 (Preservation of Canonical Measure). *The flow φ_t preserves the canonical measure of H.*

Proof. Let D be a Lebesgue measurable set. We have

$$\begin{aligned}\mu^H\big(\varphi_t^{-1}(D)\big) &= \int_{\varphi_{-t}(D)} \frac{1}{Z} e^{-H(x)}\, dx \\ &= \int_D \frac{1}{Z} e^{-H(\varphi_{-t}(x))} \left|\varphi'_{-t}(x)\right| \, dx \\ &= \int_D \frac{1}{Z} e^{-H(x)}\, dx \\ &= \mu^H(D).\end{aligned}$$

□

8.1.3 ▪ Time Reversibility

The term time reversible comes from classical mechanics. Let $q(t)$ and $p(t)$ be the height and momentum of a pendulum of mass m at time t. Suppose we release the pendulum at $q(0) = q_0$ with momentum $p(0) = v_0 m$ in a frictionless space. By conservation of energy, we know that the pendulum will come back to the same position q_0 at some time t, and its velocity will be $-v_0$. The $q - p$ time plot is symmetric i.e. the system is time reversible. Here we give a definition of time reversibility in the context of dynamical systems.

Definition 8.6 (Involution and Reversibility). *A linear map S is an involution if $S^2 = S \circ S$ is the identity map. A bijection φ is called reversible with respect to an involution S if $S \circ \varphi = \varphi^{-1} \circ S$. A differential equation is called reversible with respect to an involution S if, for every fixed t, its flow φ_t is reversible with respect to S.*

The inverse of the flow φ_t of a differential equation $\frac{dx}{dt} = u(x)$ can be intuitively viewed as the flow map φ_t of the system $\frac{dx}{dt} = -u(x)$. We can give an equivalent characterization of time reversibility:

Theorem 8.7 (Characterization of Reversibility). *A system of differential equations $\frac{dx}{dt} = u(x)$ is time reversible with respect to an involution S if and only if $S \circ u = -u \circ S$.*

Proof. Suppose $\frac{dx}{dt} = u(x)$ is time reversible with respect to S. Using linearity of S, we have

$$\begin{aligned}S \circ u(x) &= \lim_{\epsilon \to 0} \frac{1}{\epsilon}\Big(S \circ \varphi_{t+\epsilon}(x) - S \circ \varphi_t(x)\Big) \\ &= \lim_{\epsilon \to 0} \frac{1}{\epsilon}\Big(\varphi_{t+\epsilon}^{-1} \circ S(x) - \varphi_t^{-1} \circ S(x)\Big) \\ &= \frac{d}{dt}\varphi_t^{-1}\big(S(x)\big) \\ &= (-u) \circ S(x).\end{aligned}$$

The other direction follows directly from the remark above. □

Corollary 8.8 (Reversibility of Hamiltonian Systems). *Hamiltonian systems with Hamiltonian $H(q,p) = V(q) + K(p)$ such that $K(p)$ is an even function are reversible with respect to the momentum flip involution $S : (q,p) \mapsto (q,-p)$.*

Example 8.9 (Gaussian Momentum Variables and Reversibility). The kinetic energy $K(p) = \frac{1}{2}p^T M^{-1}p$ is an even function, since

$$K(-p) = \frac{1}{2}(-p)^T M^{-1}(-p) = \frac{1}{2}p^T M^{-1}p = K(p).$$

Therefore, Corollary 8.8 implies that Hamiltonian systems with Hamiltonian $H(q,p) = V(q) + \frac{1}{2}p^T M^{-1}p$ are always time reversible with respect to momentum flip. Notice that with the choice of kinetic energy $K(p) = \frac{1}{2}p^T M^{-1}p$, the marginal marginal of $f^H(q,p) \propto \exp\big(-H(q,p)\big)$ over the momentum variables is Gaussian $\mathcal{N}(0, M)$. □

8.2 ▪ From Hamiltonian Dynamics to a Sampling Algorithm

8.2.1 ▪ Idealized Hamiltonian Monte Carlo Algorithm

For the following algorithm, we consider a Hamiltonian of the form $H(q,p) = \frac{1}{2}p^T M^{-1}p + V(q)$. We would like to construct a Markov kernel that leaves the canonical measure of H invariant. Recall that our true target is the q-marginal with density proportional to e^{-V}; the Gaussian momentum p is just an auxiliary variable. As usual, we have access to the potential $V(q)$ (and its gradient) but not to the normalizing constant Z.

Algorithm 8.1 Idealized, Not-Implementable Hamiltonian Monte Carlo

Input: Initialization $(q^{(0)}, p^{(0)})$, duration parameter λ, sample size N.
For $n = 0, \dots, N-1$ do:
1: Momentum refreshment: Sample $p^* \sim \mathcal{N}(0, M)$.
2: State update: $(q^{(n+1)}, p^{(n+1)}) = \varphi_\lambda(q^{(n)}, p^*)$.
Output: Sample $\{q^{(n)}\}_{n=1}^N$.

Note that the idealized, not implementable HMC algorithm shown above presupposes that the flow map φ_λ can be evaluated, which is not true for most practical applications. However, it is important to note that if the flow map could be evaluated, there would be no need for an acceptance/rejection step to leave the target invariant; this is proved in Theorem 8.10. In practice, φ_λ is approximated with numerical integrators (which might not preserve total energy) and the acceptance/rejection step is used to counter that error. This will be made precise in Algorithm 8.2 and Theorem 8.12.

Theorem 8.10. *Let μ^H be the Gibbs distribution with energy $H(q,p) = \frac{1}{2}p^T M^{-1}p + V(q)$. The Markov kernel implicitly defined by Algorithm 8.1 leaves μ^H invariant.*

Proof. By the definition of the Hamiltonian, the momentum variables are $\mathcal{N}(0, M)$ and therefore μ^H is clearly invariant in the momentum refreshment step. On the other hand, Theorem 8.5 shows that the flow map φ_λ is measure-preserving with respect to the canonical measure of H. This implies that step 2 also preserves the distribution μ^H. □

8.2.2 ▪ Hamiltonian Monte Carlo Algorithm

Time reversibility, conservation of Hamiltonian, and symplecticity are properties that hold for Hamiltonian systems. In practice, for most problems of interest, Hamilton equations cannot be

solved in closed form and need to be discretized. Discretizations of Hamiltonian systems may preserve *exactly* some of these properties. For instance, the famous leapfrog method is reversible and symplectic. We remark that unfortunately a numerical solver for Hamilton equations cannot be simultaneously symplectic *and* conserve the Hamiltonian exactly.

Example 8.11 (The Leapfrog Integrator). The leapfrog method applied to Hamilton equations with $K(p) := \sum_{i=1}^{d} \frac{p_i^2}{2m_i}$ and step-size $\epsilon > 0$ is

$$
\begin{aligned}
p_i\left(t + \frac{\varepsilon}{2}\right) &= p_i(t) - \frac{\varepsilon}{2}\frac{\partial V}{\partial q_i}(q(t)) && \text{(half momentum step)} \\
q_i(t+\varepsilon) &= q_i(t) + \varepsilon \frac{p_i\left(t + \frac{\varepsilon}{2}\right)}{m_i} && \text{(full position step)} \\
p_i(t+\varepsilon) &= p_i\left(t + \frac{\varepsilon}{2}\right) - \frac{\varepsilon}{2}\frac{\partial V}{\partial q_i}(q(t+\varepsilon)). && \text{(half momentum step)}
\end{aligned}
$$

It is time reversible and conserves volume exactly. To discretize Hamilton equations for a time interval $[t, t+\lambda)$ of length λ, we may take L leapfrog steps with $\lambda = \epsilon L$. □

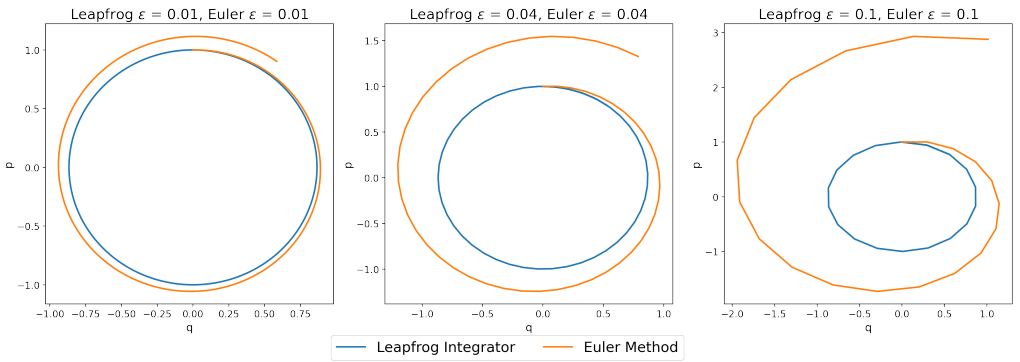


Figure 8.2. *Simulate the Hamiltonian* $\frac{3p^2}{2} + \frac{4q^2}{2}$ *using leapfrog and Euler method updates:* $p(t+\varepsilon) = p(t) + \varepsilon\frac{dp}{dt}, q(t+\varepsilon) = q(t) + \varepsilon\frac{dq}{dt}$.

Let Ψ_λ be a symplectic numerical integrator, time reversible with respect to the momentum flip involution, that approximates the flow map φ_λ. The Hamiltonian Monte Carlo algorithm proposes moves using Ψ_λ and corrects for the fact that Ψ_λ does not preserve the Hamiltonian by using an accept/reject mechanism.

Algorithm 8.2 Hamiltonian Monte Carlo (HMC)

Input: Initialization $(q^{(0)}, p^{(0)})$, duration parameter λ, sample size N.

For $n = 0, \dots, N-1$ do:

1: Momentum refreshment: Sample $p^{\text{NEW}} \sim \mathcal{N}(0, M)$.

2: Propose new state: $(q^*, p^*) = \Psi_\lambda\big(q^{(n)}, p^{\text{NEW}}\big)$.

3: Set

$$\big(q^{(n+1)}, p^{(n+1)}\big) = \begin{cases} (q^*, p^*) & \text{with probability } a := \min\Big\{1, e^{H(q^{(n)}, p^{\text{NEW}}) - H(q^*, p^*)}\Big\}, \\ \big(q^{(n)}, -p^{\text{NEW}}\big) & \text{with probability } 1 - a. \end{cases}$$

Output: Sample $\{q^{(n)}\}_{n=1}^N$.

Pros and Cons: Using Hamiltonian dynamics in an extended state space, HMC breaks away from the local behavior of RWMH and MALA proposals. As a result, HMC scales better to high-dimensional problems (see below for further discussion). However, HMC is often harder to implement and to tune than RWMH and MALA. Similar to MALA, HMC requires evaluation of the gradient of V, which can be expensive.

Scaling of HMC Classical theoretical results for HMC developed in the same large-d scaling setting that we considered for RWMH in Chapter 4 and for MALA in Chapter 6 revealed the scaling of the leapfrog step-size should be $\mathcal{O}(d^{-1/4})$. Thus, HMC requires $\mathcal{O}(d^{1/4})$ steps to traverse the state space, which should be compared to $\mathcal{O}(d^{1/3})$ for MALA and $\mathcal{O}(d^1)$ for RWMH. These analyses also show that the optimal acceptance rate for HMC is 0.651, compared to 0.574 for MALA and 0.234 for RWMH.

Our goal in the rest of this section is to understand why the acceptance probability in the HMC algorithm is the correct one to ensure that f^H invariant is an invariant distribution. To that end, we will make use of the time reversibility of the numerical integrator and its preservation of volume (otherwise a Jacobian would appear in the acceptance probability). The main theoretical result of this chapter is the following.

Theorem 8.12 (HMC Leaves Target Invariant). *Suppose that Ψ_λ is symplectic and reversible with respect to the momentum flip involution. Then the Markov kernel implicitly defined by the Hamiltonian Monte Carlo algorithm leaves invariant the canonical measure of H.*

Proof. Throughout the proof we denote by S the momentum flip involution. Clearly the momentum refreshment step preserves the canonical measure μ^H. It suffices to show that μ^H is invariant under steps 2 and 3. The corresponding Markov kernel is given by

$$\pi_x(dy) = a(x)\delta_{\Psi_\lambda(x)}(y)dy + \big(1 - a(x)\big)\delta_{S(x)}(y)dy,$$

where

$$a(x) := \min\Big\{1, e^{H(x) - H(\Psi_\lambda(x))}\Big\}.$$

Let $A \subset \mathbb{R}^{2d}$ be a Lebesgue measurable set. We need to show that

$$\int_{\mathbb{R}^{2d}} \pi_x(A)\mu^H(dx) = \int_A \mu^H(dx).$$

Expanding the left-hand side gives

$$\int_{\mathbb{R}^{2d}} \pi_x(A)\mu^H(dx) = \int_{\mathbb{R}^{2d}} a(x)\mathbf{1}_A\big(\Psi_\lambda(x)\big)f^H(x)\,dx + \int_{\mathbb{R}^{2d}} \big(1-a(x)\big)\mathbf{1}_A\big(S(x)\big)f^H(x)\,dx$$
$$= \int_{\mathbb{R}^{2d}} \mathbf{1}_A\big(S(x)\big)f^H(x)\,dx + \int_{\mathbb{R}^{2d}} a(x)\mathbf{1}_A\big(\Psi_\lambda(x)\big)f^H(x)\,dx - \int_{\mathbb{R}^{2d}} a(x)\mathbf{1}_A\big(S(x)\big)f^H(x)\,dx.$$

First, $\int_{\mathbb{R}^{2d}} \mathbf{1}_A\big(S(x)\big)f^H(x)\,dx = \int_A \mu^H(dx)$ since S preserves the canonical measure. Next, we use a change of variables in the third term to show that it is equal to the second one. Precisely, using that $S\circ S = id$, and that the determinant Jacobians of Ψ_λ and S are identically one due to the symplecticity of Ψ_λ and reversibility of S, we have

$$\int_{\mathbb{R}^{2d}} a(x)\mathbf{1}_A\big(S(x)\big)f^H(x)\,dx = \int_{\mathbb{R}^{2d}} a\big(S\circ\Psi_\lambda(x)\big)\mathbf{1}_A\big(\Psi_\lambda(x)\big)f^H\big(\Psi_\lambda(x)\big)\,dx.$$

Now, using that $\Psi_\lambda \circ S \circ \Psi_\lambda = S$ since Ψ_λ is time reversible with respect to S and also that $H\circ S = H$, we deduce that

$$\begin{aligned} a\big(S\circ\Psi_\lambda(x)\big) &= \min\Big\{1, e^{H(S\circ\Psi_\lambda(x))-H(\Psi_\lambda\circ S\circ\Psi_\lambda(x))}\Big\} \\ &= \min\Big\{1, e^{H(S\circ\Psi_\lambda(x))-H(S(x))}\Big\} \\ &= \min\Big\{1, e^{H(\Psi_\lambda(x))-H(x)}\Big\} \\ &= \frac{f^H(x)}{f^H\big(\Psi_\lambda(x)\big)}a(x). \end{aligned}$$

Substituting this back to the previous integral gives

$$\int_{\mathbb{R}^{2d}} a(x)\mathbf{1}_A\big(S(x)\big)f^H(x)\,dx = \int_{\mathbb{R}^{2d}} a(x)\mathbf{1}_A\big(\Psi_\lambda(x)\big)f^H(x)\,dx,$$

as desired. □

8.3 ▪ Discussion and Bibliography

HMC was proposed in the physics literature under the name Hybrid Monte Carlo [60]. The method was introduced to the statistics community through the work [171]. An accessible introduction to HMC is [170]; see also [23]. Theorem 8.4, which shows that Hamiltonian flows are symplectic, was proved in [186]. Further background on Hamiltonian dynamics and their numerical solution can be found in [209, 101]. Numerical integrators tailored to HMC are reviewed and compared in [31, 42].

In this chapter, we have seen that the Metropolis Hastings framework can be generalized by means of an involution map, which we take as a momentum flip for HMC methods. The use of involutions within the Metropolis Hastings algorithm was proposed in [11, 92], and has been leveraged to establish mixing rates for Hamiltonian Monte Carlo in [92]. The theory in [92], which relies on the weak Harris approach developed in [102], also covers implementations of HMC in infinite-dimensional Hilbert spaces [22]. The paper [30] relies instead on a novel coupling technique to analyze the convergence of HMC algorithms. Further convergence results can be found in [148, 65] and optimal scaling of HMC in high dimension was studied in [21]. Tuning HMC algorithms can be challenging. The No-U-Turn (NUTS) algorithm [113] introduces a computational framework to automatically set the duration parameter λ and the step-size ϵ.

Chapter 9

Sequential Monte Carlo

This chapter is devoted to Monte Carlo algorithms that leverage weighted samples to approximate a sequence of target distributions. Sequential Monte Carlo algorithms combine three main ingredients: (i) a Markov kernel to propagate a collection of particles; (ii) a weighting mechanism akin to importance sampling; and (iii) a resampling scheme to alleviate weight degeneracy. For a given problem, there is often significant flexibility in how to specify and implement these three components, which contributes to making sequential Monte Carlo an extremely rich family of algorithms with deep theoretical underpinnings.

We will focus on Bayesian filtering, where the targets represent the conditional law of a time-evolving hidden state given noisy observations. In this context, we will introduce two algorithms: the bootstrap particle filter and the optimal particle filter. Both algorithms rely on different Markov kernels to propagate particles, and, consequently, on different weighting mechanisms. We emphasize, however, that while the study of sequential Monte Carlo in Bayesian filtering provides a natural framework to introduce some key ideas, the methodology is applicable much more broadly. In some applications, the sequence of targets is not specified by the problem at hand but arises instead by artificially introducing a sequence of tempered targets, similar to the annealing strategies in Chapter 7. For instance, in applications to rare event sampling, the sequence of targets may be chosen so that they gradually concentrate around the event of interest.

As discussed in Chapter 3, the main caveat of sampling algorithms that rely on weighting mechanisms is that the variance of the weights becomes larger as target and proposal distributions become further apart, which can cause weight degeneracy in high-dimensional problems. Indeed, sequential Monte Carlo methods are extremely powerful in low and moderate dimension, but they often behave poorly in high dimension. The use of resampling schemes within sequential Monte Carlo algorithms is partly motivated by their weight degeneracy in high dimension; however, resampling cannot fully resolve this fundamental limitation of algorithms relying on importance weights.

This chapter is structured as follows. Section 9.1 overviews the Bayesian formulation of the filtering problem and introduces the sequence of target distributions we seek to approximate. Section 9.2 introduces the bootstrap particle filter and shows its convergence in the large particle limit using a numerical analysis argument reminiscent of Lax equivalence theorem. Section 9.3 introduces optimal particle filters, which partly alleviate the weight degeneracy of bootstrap particle filters. Section 9.4 closes with bibliographical remarks.

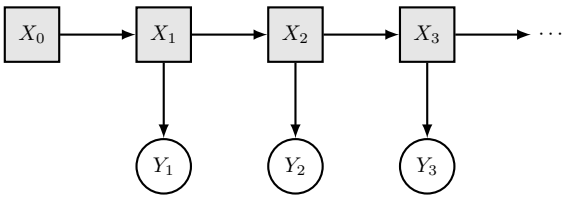


Figure 9.1. *Graphical representation of the assumed dependence structure.*

9.1 ▪ Bayesian Filtering

Hidden Markov models arise in applications where a latent or "hidden" Markov process $\{X_j\}_{j=0}^{\infty}$ needs to be estimated based on an *observed* process $\{Y_j\}_{j=1}^{\infty}$. For concreteness, we assume that the hidden and observed processes are specified as follows:

$$\begin{aligned}
&\text{(Initial condition)} && X_0 \sim f_0, \\
&\text{(Dynamics model)} && X_{j+1} = a(X_j) + \xi_{j+1}, && \xi_{j+1} \sim \mathcal{N}(0,\Sigma), \quad j = 0,1,\dots \\
&\text{(Observation model)} && Y_{j+1} = b(X_{j+1}) + \eta_{j+1}, && \eta_{j+1} \sim \mathcal{N}(0,\Gamma), \quad j = 0,1,\dots
\end{aligned}$$

where X_0, $\{\xi_j\}$, and $\{\eta_j\}$ are all independent; see Figure 9.1 for a representation of the resulting dependence structure of the hidden and observed processes. The initial distribution f_0, the dynamics map $a : \mathbb{R}^d \to \mathbb{R}^d$, the observation map $b : \mathbb{R}^d \to \mathbb{R}^k$, and the positive definite covariances Σ and Γ are all assumed to be known. We assume Gaussianity of ξ_j and η_j for pedagogical reasons, as it leads to concrete and familiar expressions for the Markov kernel and likelihood function implicitly defined, respectively, by the dynamics and observation models.

We are interested in estimating X_j given the observations $Y_{1:j} := \{Y_i\}_{i=1}^{j}$ available up to time j. In the Bayesian framework, inference is carried out via the posterior or *filtering distribution* of $X_j|Y_{1:j}$, denoted by f_j. We will study two sequential Monte Carlo algorithms to approximate f_j: the bootstrap particle filter and the optimal particle filter. An important feature of these algorithms is that they are *sequential*: at time $j+1$ they approximate the distribution f_{j+1} using an approximation of f_j and the new observation Y_{j+1}. To obtain a particle approximation of the filtering distribution, each algorithm relies on a different conceptual decomposition of the filtering step:

1. Bootstrap decomposition: in the bootstrap particle filter, particles are propagated through the dynamics model, and then Bayes' formula is applied to assimilate the new observation, leading to the following decomposition of the filtering step: $X_j|Y_{1:j} \longrightarrow X_{j+1}|Y_{1:j} \longrightarrow X_{j+1}|Y_{1:j+1}$.

2. Optimal decomposition: in the optimal particle filter, Bayes' formula is applied to particles before they are propagated through the dynamics, leading to the following decomposition of the filtering step: $X_j|Y_{1:j} \longrightarrow X_j|Y_{1:j+1} \longrightarrow X_{j+1}|Y_{1:j+1}$.

Throughout the chapter, we will use the notation

$$\pi(x) = \frac{1}{N}\sum_{n=1}^{N} \delta\big(x - X^{(n)}\big)$$

to represent the empirical measure defined by a sample $\{X^{(n)}\}_{n=1}^{N}$, so that, for sufficiently smooth test function h,

$$\mathcal{I}_\pi[h] = \mathbb{E}_\pi[h] = \frac{1}{N}\sum_{n=1}^{N} h\big(X^{(n)}\big).$$

We informally treat a distribution of this form as a p.d.f. Likewise, given normalized weights $\{w^{(n)}\}_{n=1}^N$ and a sample $\{X^{(n)}\}_{n=1}^N$, we will use the notation

$$\pi_w(x) = \sum_{n=1}^N w^{(n)} \delta\big(x - X^{(n)}\big)$$

to represent the weighted empirical measure defined by the requirement that, for sufficiently smooth test function h,

$$\mathcal{I}_{\pi_w}[h] = \mathbb{E}_{\pi_w}[h] = \sum_{n=1}^N w^{(n)} h\big(X^{(n)}\big).$$

Finally, for positive definite matrix A and vector v, we will denote by $|v|_A^2 := v^T A^{-1} v$ the Mahalanobis norm.

9.2 ▪ Bootstrap Particle Filter

In the bootstrap particle filter, we propagate particles through the dynamics model, and then apply Bayes' formula to weight each particle according to the likelihood function implied by the observation model. At each time-step, we resample new particles to avoid weight degeneracy.

Algorithm 9.1 Bootstrap Particle Filter

Input: Initial distribution $f_0^N = f_0$, observations $Y_{1:J}$, sample size N.

For $j = 0, \dots, J-1$ do for $1 \le n \le N$:

1: Draw $X_j^{(n)} \overset{\text{i.i.d.}}{\sim} f_j^N$.

2: Propagate each sample through the dynamics model:

$$\widehat{X}_{j+1}^{(n)} = a\big(X_j^{(n)}\big) + \xi_{j+1}^{(n)}, \qquad \xi_{j+1}^{(n)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma).$$

3: Weight each sample with the likelihood defined by the observation model:

$$\tilde{w}_{j+1}^{(n)} = \exp\Big(-\frac{1}{2} \big| Y_{j+1} - b\big(\widehat{X}_{j+1}^{(n)}\big) \big|_\Gamma^2 \Big).$$

4: Normalize the weights:

$$w_{j+1}^{(n)} = \frac{\tilde{w}_{j+1}^{(n)}}{\sum_{n=1}^N \tilde{w}_{j+1}^{(n)}}.$$

5: Set

$$f_{j+1}^N(x) = \sum_{n=1}^N w_{j+1}^{(n)} \delta\big(x - \widehat{X}_{j+1}^{(n)}\big).$$

Output: Approximation f_J^N to the filtering distribution f_J.

Pros and Cons: The bootstrap particle filter is a sequential Monte Carlo algorithm to approximate a time-evolving sequence of target distributions. It can be applied in general hidden Markov models and it is provably convergent in the large N limit. However, in high-dimensional settings the weights typically have a large variance, leading to a small effective sample size and poor performance.

Our goal now is to show that, for large N, f_J^N is close to f_J. To perform the analysis we first introduce three operators acting on probability distributions.

9.2.1 ▪ Prediction, Analysis, and Sampling Operators

The first operator we introduce, which we call the prediction operator, describes how p.d.f.s propagate through the Markov kernel specified by the dynamics model.

Definition 9.1 (Prediction Operator). *The* prediction *operator $\mathcal{P}$ acting on a p.d.f. π is given by*

$$(\mathcal{P}\pi)(x) = \int_{\mathbb{R}^d} p(\tilde{x}, x)\pi(\tilde{x})\, d\tilde{x},$$

where p is the Markov kernel associated with the dynamics model, namely

$$p(x, \tilde{x}) = \frac{1}{\sqrt{(2\pi)^d \det \Sigma}} \exp\Big(-\frac{1}{2}\big|\tilde{x} - a(x)\big|_\Sigma^2\Big). \tag{9.1}$$

The second operator we introduce, which we call the analysis operator, takes as input a p.d.f. π and produces as output the posterior p.d.f. obtained by taking π as the prior and using the likelihood function specified by the observation model.

Definition 9.2 (Analysis Operator). *The* analysis *operator $\mathcal{A}_j$ acting on a p.d.f. π is given by*

$$(\mathcal{A}_j\pi)(x) \propto \exp\Big(-\frac{1}{2}|Y_{j+1} - b(x)|_\Gamma^2\Big)\pi(x).$$

Denoting by $\widehat{f}_{j+1} := \mathcal{P}f_j$ the distribution of $X_{j+1}|Y_{1:j}$, we thus have that $f_{j+1} = \mathcal{A}_j\widehat{f}_{j+1}$. Consequently, we can write the filtering step as

$$f_{j+1} = \mathcal{A}_j\mathcal{P}f_j. \tag{9.2}$$

The operator $\mathcal{P}$ does not depend on j because our dynamics model is time-homogeneous. However, the operator $\mathcal{A}_j$ depends on the observed data Y_{j+1}, and thus on the time index j.

Remark 9.3. *The operator $\mathcal{A}_j$ acting on a distribution of the form*

$$\pi(x) = \frac{1}{N}\sum \delta\big(x - X^{(n)}\big)$$

gives

$$(\mathcal{A}_j\pi)(x) = \sum_{n=1}^{N} w^{(n)}\delta\big(x - X^{(n)}\big),$$

where

$$\tilde{w}^{(n)} = \exp\Big(-\frac{1}{2}\big|Y_{j+1} - b(x)\big|_\Gamma^2\Big)$$

and $w^{(n)}$ are the normalized weights.

The third operator we introduce, which we call the sampling operator, takes as input a p.d.f. π and outputs the empirical p.d.f. obtained by drawing N samples from π.

Definition 9.4 (Sampling Operator). *The* sampling *operator $\mathcal{S}^N$ acts on a distribution π by drawing N samples from π and producing a Dirac approximation*

$$(\mathcal{S}^N\pi)(x) = \frac{1}{N}\sum_{n=1}^{N}\delta\big(x - X^{(n)}\big), \quad X^{(n)} \overset{\text{i.i.d.}}{\sim} \pi.$$

Notice that $\mathcal{S}^N\pi$ is random probability distribution due to sampling from π. With the three operators we just defined, we can succinctly write the bootstrap particle filter as

$$f_{j+1}^N = \mathcal{A}_j\mathcal{S}^N\mathcal{P}f_j^N \tag{9.3}$$

In order to reconcile this way of writing the bootstrap particle filter with the algorithmic implementation in Algorithm 9.1, it is important to note that doing prediction and then sampling is equivalent to sampling first and then doing prediction.

9.2.2 ▪ Bounds for Prediction, Analysis, and Sampling

We now seek to obtain bounds for the prediction, analysis, and sampling operators. We will use the following distance between random probability distributions:

$$d(\pi,\pi') = \sup_{|h|_\infty\le 1}\sqrt{\mathbb{E}\big[(\mathbb{E}_\pi[h] - \mathbb{E}_{\pi'}[h])^2\big]}. \tag{9.4}$$

It can be verified that (9.4) indeed defines a valid distance in the space of random probability distributions, so that in particular it satisfies the triangle inequality. In what follows, the randomness in the distributions we consider will arise from sampling, and the outer expectation is with respect to such randomness.

We first show that the prediction operator is a contraction in this metric.

Lemma 9.5. *It holds that*

$$d(\mathcal{P}\pi,\mathcal{P}\pi') \le d(\pi,\pi').$$

Proof. Let $|h|_\infty \le 1$ and define

$$h^\dagger(x) = \int_{\mathbb{R}^d} p(x,\tilde{x})h(\tilde{x})\,d\tilde{x},$$

where recall that p denotes the transition kernel (9.1) of the dynamics model. Note that

$$|h^\dagger(x)| \le \int_{\mathbb{R}^d} p(x,\tilde{x})\,d\tilde{x} = 1$$

and

$$\begin{aligned}
\mathbb{E}_\pi[h^\dagger] &= \int_{\mathbb{R}^d} h^\dagger(x)\pi(x)\,dx \\
&= \int_{\mathbb{R}^d}\Big(\int_{\mathbb{R}^d} p(x,\tilde{x})h(\tilde{x})\,d\tilde{x}\Big)\pi(x)\,dx \\
&= \int_{\mathbb{R}^d}\Big(\int_{\mathbb{R}^d} p(x,\tilde{x})\pi(x)\,dx\Big)h(\tilde{x})\,d\tilde{x} \\
&= \int_{\mathbb{R}^d}(\mathcal{P}\pi)(\tilde{x})h(\tilde{x})\,d\tilde{x},
\end{aligned}$$

and so

$$\mathbb{E}_\pi[h^\dagger] = \mathbb{E}_{\mathcal{P}\pi}[h].$$

Finally

$$\begin{aligned} d(\mathcal{P}\pi, \mathcal{P}\pi') &= \sup_{|h|_\infty \le 1} \sqrt{\mathbb{E}\big[(\mathbb{E}_{\mathcal{P}\pi}[h] - \mathbb{E}_{\mathcal{P}\pi'}[h])^2\big]} \\ &\le \sup_{|h^\dagger| \le 1} \sqrt{\mathbb{E}\big[(\mathbb{E}_{\pi}[h^\dagger] - \mathbb{E}_{\pi'}[h^\dagger])^2\big]} \\ &= d(\pi, \pi'), \end{aligned}$$

as desired. □

We next establish a bound for the analysis operator under the following assumption:

Assumption 9.6. *There exists $\kappa \in (0,1)$ such that, for all $x \in \mathbb{R}^d$ and $j \in \{0, 1, \dots, J-1\}$,*

$$\kappa \le w_{j+1}(x) := \exp\Big(-\frac{1}{2}|Y_{j+1} - b(x)|_\Gamma^2\Big) \le \kappa^{-1}.$$

While this strong assumption can be significantly relaxed, it will allow us to streamline our theory. In particular, notice that the proof of the following result is nearly identical to that of Theorem 3.5 for autonormalized importance sampling.

Lemma 9.7. *Under Assumption 9.6,*

$$d(\mathcal{A}_j\pi, \mathcal{A}_j\pi') \le \frac{2}{\kappa^2} d(\pi, \pi').$$

Proof. Let h with $|h|_\infty \le 1$, then

$$\begin{aligned} \mathbb{E}_{\mathcal{A}\pi}[h] - \mathbb{E}_{\mathcal{A}\pi'}[h] =& \frac{\mathbb{E}_\pi[hw]}{\mathbb{E}_\pi[w]} - \frac{\mathbb{E}_{\pi'}[hw]}{\mathbb{E}_{\pi'}[w]} \\ =& \frac{\mathbb{E}_\pi[hw]}{\mathbb{E}_\pi[w]} - \frac{\mathbb{E}_{\pi'}[hw]}{\mathbb{E}_\pi[w]} + \frac{\mathbb{E}_{\pi'}[hw]}{\mathbb{E}_\pi[w]} - \frac{\mathbb{E}_{\pi'}[hw]}{\mathbb{E}_{\pi'}[w]} \\ =& \frac{1}{\kappa}\left(\frac{\mathbb{E}_\pi[h\kappa w] - \mathbb{E}_{\pi'}[h\kappa w]}{\mathbb{E}_\pi[w]} + \frac{\mathbb{E}_{\pi'}[hw]}{\mathbb{E}_{\pi'}[w]}\frac{\mathbb{E}_{\pi'}[\kappa w] - \mathbb{E}_\pi[\kappa w]}{\mathbb{E}_\pi[w]}\right). \end{aligned}$$

Since $|h|_\infty \le 1$, we have

$$|\mathbb{E}_{\mathcal{A}\pi'}[h]| = \left|\frac{\mathbb{E}_{\pi'}[hw]}{\mathbb{E}_{\pi'}[w]}\right| \le 1.$$

In addition, by Assumption 9.6, we have $\mathbb{E}_\pi[w] \ge \kappa$. Therefore,

$$|\mathbb{E}_{\mathcal{A}\pi}[h] - \mathbb{E}_{\mathcal{A}\pi'}[h]| \le \frac{1}{\kappa^2}\Big(|\mathbb{E}_\pi[h\kappa w] - \mathbb{E}_{\pi'}[h\kappa w]| + |\mathbb{E}_{\pi'}[\kappa w] - \mathbb{E}_\pi[\kappa w]|\Big),$$

and so

$$\begin{aligned} \mathbb{E}\Big[\big(\mathbb{E}_{\mathcal{A}\pi}[h] - \mathbb{E}_{\mathcal{A}\pi'}[h]\big)^2\Big] &\le \frac{2}{\kappa^4}\left(\mathbb{E}\Big[\big(\mathbb{E}_\pi[h\kappa w] - \mathbb{E}_{\pi'}[h\kappa w]\big)^2\Big] + \mathbb{E}\Big[\big(\mathbb{E}_{\pi'}[\kappa w] - \mathbb{E}_\pi[\kappa w]\big)^2\Big]\right) \\ &\le \frac{4}{\kappa^4} d(\pi, \pi')^2, \end{aligned}$$

where in the last line we have used that $|\kappa w| \leq 1$. Taking square roots and the supremum over $|h|_\infty \leq 1$ completes the proof. □

Lastly, the sampling operator can be bounded using Theorem 3.1 for classical Monte Carlo integration.

Lemma 9.8. *For any distribution π,*

$$d(\pi, \mathcal{S}^N \pi) \leq \frac{1}{\sqrt{N}}.$$

Proof. For any h with $|h|_\infty \leq 1$,

$$\mathbb{E}_{\mathcal{S}^N \pi}[h] = \frac{1}{N} \sum_{n=1}^{N} h\big(X^{(n)}\big),$$

where $X^{(n)} \overset{\text{i.i.d.}}{\sim} \pi$. Therefore, using Theorem 3.1 for classical Monte Carlo integration,

$$\mathbb{E}\Big[\big(\mathbb{E}_{\mathcal{S}^N \pi}[h] - \mathbb{E}_\pi[h]\big)^2\Big] = \mathbb{E}\Big[\big(\mathcal{I}_\pi^{\text{MC}}[h] - \mathcal{I}_\pi[h]\big)^2\Big] = \frac{\mathbb{V}_\pi[h]}{N} \leq \frac{1}{N}.$$

Taking square roots and the supremum over $|h|_\infty \leq 1$ gives the result. □

9.2.3 ▪ Error of Bootstrap Particle Filter

Recall from equations (9.2) and (9.3) that the filtering step and the bootstrap particle filter can be succinctly written as

$$f_{j+1} = \mathcal{A}_j \mathcal{P} f_j, \tag{9.5}$$
$$f_{j+1}^N = \mathcal{A}_j \mathcal{S}^N \mathcal{P} f_j^N. \tag{9.6}$$

With the bounds we derived for prediction, analysis, and sampling operators in Lemmas 9.5, 9.7, and 9.8, the following theorem controls the error of the bootstrap particle filter. The proof resembles the classical Lax equivalence framework in numerical analysis, where convergence is established by ensuring consistency and stability.

Theorem 9.9 (Error of Bootstrap Particle Filter). *Under Assumption 9.6, there exists a constant $c = c(J, \kappa)$ such that, for $j = 1, \ldots, J$,*

$$d(f_j, f_j^N) \leq \frac{c}{\sqrt{N}}.$$

Proof. Let $e_j = d(f_j, f_j^N)$, then using (9.5) and the triangle inequality for the distance d defined in (9.4), we have

$$\begin{aligned} e_{j+1} = d(f_{j+1}, f_{j+1}^N) &= d(\mathcal{A}_j \mathcal{P} f_j, \mathcal{A}_j \mathcal{S}^N \mathcal{P} f_j^N) \\ &\leq d(\mathcal{A}_j \mathcal{P} f_j, \mathcal{A}_j \mathcal{P} f_j^N) + d(\mathcal{A}_j \mathcal{P} f_j^N, \mathcal{A}_j \mathcal{S}^N \mathcal{P} f_j^N). \end{aligned}$$

The first term represents the growth of error over one time-iteration by propagating through the operator $\mathcal{A}_j\mathcal{P}$ which governs the true evolution of the filtering distribution, whereas the second term is a truncation error, i.e. the error incurred by the algorithm in one iteration. We then have

$$\begin{aligned} e_{j+1} &\leq \frac{2}{\kappa^2}\Big(d(\mathcal{P}f_j^N, \mathcal{P}f_j) + d(\mathcal{P}f_j^N, \mathcal{S}^N\mathcal{P}f_j^N)\Big) \\ &\leq \frac{2}{\kappa^2}\Big(e_j + \frac{1}{\sqrt{N}}\Big), \end{aligned}$$

where in the first inequality we used Lemma 9.7 for the analysis operator and in the second inequality we used Lemmas 9.5 and 9.8 for the prediction and sampling operators. Letting $\lambda = \frac{2}{\kappa^2}$, it then follows from induction that

$$e_j \leq \lambda^j e_0 + \frac{\lambda}{\sqrt{N}}\frac{1-\lambda^j}{1-\lambda},$$

where $e_0 = 0$ since $f_0^N = f_0$. Now, noting that the above expression is monotonically increasing in j completes the proof by setting $c = \frac{\lambda(1-\lambda^J)}{1-\lambda}$. □

9.3 ▪ Optimal Particle Filter

The bootstrap particle filter propagates particles through the dynamics and then assigns them weights according to the likelihood defined by the observation model. In contrast, the optimal particle filter stems from a different conceptual decomposition of the filtering step, where we first apply Bayes' formula and then propagate particles through a Markov kernel which depends on the new observation.

9.3.1 ▪ Derivation of the Optimal Filter Decomposition

Let $\mathbb{P}(\cdot|\cdot)$ denote conditional density. The following calculation illustrates the decomposition of the filtering step that the optimal particle filter approximates via particles:

$$\begin{aligned} \mathbb{P}(X_{j+1}|Y_{1:j+1}) &= \int_{\mathbb{R}^d} \mathbb{P}(X_j, X_{j+1}|Y_{1:j+1})\, dX_j \\ &= \int_{\mathbb{R}^d} \mathbb{P}(X_{j+1}|X_j, Y_{1:j+1})\, \mathbb{P}(X_j|Y_{1:j+1})\, dX_j \\ &= \int_{\mathbb{R}^d} \mathbb{P}(X_{j+1}|X_j, Y_{j+1})\, \mathbb{P}(X_j|Y_{1:j+1})\, dX_j \\ &= \int_{\mathbb{R}^d} \mathbb{P}(X_{j+1}|X_j, Y_{j+1}) \frac{\mathbb{P}(Y_{j+1}|X_j, Y_{1:j})}{\mathbb{P}(Y_{j+1}|Y_{1:j})}\, \mathbb{P}(X_j|Y_{1:j})\, dX_j \\ &= \int_{\mathbb{R}^d} \mathbb{P}(X_{j+1}|X_j, Y_{j+1}) \frac{\mathbb{P}(Y_{j+1}|X_j)}{\mathbb{P}(Y_{j+1}|Y_{1:j})}\, \mathbb{P}(X_j|Y_{1:j})\, dX_j \\ &= \mathcal{P}_j^{\text{OPF}} \mathcal{A}_j^{\text{OPF}}\, \mathbb{P}(X_j|Y_{1:j}), \end{aligned}$$

where

$$\begin{aligned} \mathcal{P}_j^{\text{OPF}}\pi(x_{j+1}) &= \int_{\mathbb{R}^d} \mathbb{P}(x_{j+1}|X_j, Y_{j+1})\pi(X_j)\, dX_j, \\ \mathcal{A}_j^{\text{OPF}}\pi(x_j) &\propto \mathbb{P}(Y_{j+1}|x_j)\pi(x_j). \end{aligned}$$

We have therefore derived the following decomposition of the filtering step:

$$f_{j+1} = \mathcal{P}_j^{\text{OPF}} \mathcal{A}_j^{\text{OPF}} f_j.$$

9.3.2 ▪ Implementation

In general, it is not possible to implement the optimal particle filter in the fully nonlinear setting due to two computational bottlenecks:

- There may not be a closed formula for evaluating the likelihood $\mathbb{P}(Y_{j+1}|X_j)$, making unfeasible the computation of the particle weights.

- It may not be possible to sample from the Markov kernel $\mathbb{P}(X_{j+1}|X_j, Y_{j+1})$, making unfeasible the propagation of particles.

However, when the observation function $b(\cdot)$ is linear, i.e. $b(\cdot) = H\cdot$ for some matrix H, both bottlenecks may be overcome. We thus consider the following setting, which arises in many applications:

$$\begin{aligned} X_{j+1} &= a(X_j) + \xi_{j+1}, & \xi_{j+1} &\sim \mathcal{N}(0, \Sigma), \\ Y_{j+1} &= HX_{j+1} + \eta_{j+1}, & \eta_{j+1} &\sim \mathcal{N}(0, \Gamma), \end{aligned}$$

where X_0, $\{\xi_j\}$, and $\{\eta_j\}$ are all assumed to be independent. First, note that

$$Y_{j+1} = Ha(X_j) + H\xi_{j+1} + \eta_{j+1},$$

which implies that $\mathbb{P}(Y_{j+1}|X_j) = \mathcal{N}\big(Ha(X_j), H\Sigma H^T + \Gamma\big)$. Second, note that

$$\mathbb{P}(X_{j+1}|X_j, Y_{j+1}) \propto \exp\bigg(-\frac{1}{2}\big|Y_{j+1} - HX_{j+1}\big|_\Gamma^2 - \frac{1}{2}\big|X_{j+1} - a(X_j)\big|_\Sigma^2\bigg),$$

which implies via completion of the square that $\mathbb{P}(X_{j+1}|X_j, Y_{j+1}) = \mathcal{N}(m_{j+1}, C)$, with

$$\begin{aligned} m_{j+1} &= (I - KH)a(X_j) + KY_{j+1}, \\ C &= (I - KH)\Sigma, \\ K &= \Sigma H^T S^{-1}, \\ S &= H\Sigma H^T + \Gamma. \end{aligned}$$

These formulas are similar to those arising in the Kalman filter algorithm. We are now ready to present a practical implementation of the optimal particle filter with linear observations.

Algorithm 9.2 Optimal Particle Filter

Input: Initial distribution f_0, observations $Y_{1:J}$, sample size N.

Initial sampling: Sample $X_0^{(1)}, \dots, X_0^{(N)} \overset{\text{i.i.d.}}{\sim} f_0$ so that $f_0^N = \mathcal{S}^N f_0$.

Subsequent sampling:

For $j = 0, \dots, J-1$, do for $1 \le n \le N$:

1: Set
$$\widehat{X}_{j+1}^{(n)} = (I - KH)a\big(X_j^{(n)}\big) + KY_{j+1} + \zeta_{j+1}^{(n)}, \qquad \zeta_{j+1}^{(n)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, C).$$

2: Set
$$\tilde{w}_{j+1}^{(n)} = \exp\Big(-\frac{1}{2}\big|Y_{j+1} - Ha\big(X_j^{(n)}\big)\big|_S^2\Big).$$

3: Normalize the weights:
$$w_{j+1}^{(n)} = \frac{\tilde{w}_{j+1}^{(n)}}{\sum_{n=1}^N \tilde{w}_{j+1}^{(n)}}.$$

4: Draw $X_{j+1}^{(n)} \overset{\text{i.i.d.}}{\sim} \sum_{n=1}^N w_{j+1}^{(n)} \delta\big(x - \widehat{X}_{j+1}^{(n)}\big)$.

5: Set
$$f_{j+1}^N(x) = \frac{1}{N} \sum_{n=1}^N \delta\big(x - X_{j+1}^{(n)}\big).$$

Output: Approximation f_J^N to the filtering distribution f_J.

Pros and Cons: Similar to the bootstrap particle filter, the optimal particle filter is convergent in the large N limit, but also suffers from small effective sample size in high dimensional settings. The variance of the weights is smaller, however, for the optimal particle filter than for the bootstrap particle filter. It is important to notice that, in contrast to the bootstrap filter, the optimal filter relies on Gaussian assumptions on the noise in the dynamics and observation models, and on linearity of the observations. Thus, it can only be deployed on a smaller class of problems.

9.4 ▪ Discussion and Bibliography

Sequential Monte Carlo methods are overviewed from an algorithmic viewpoint in [58, 59] and from a mathematical perspective in [54, 49]. The term "bootstrap" particle filter was coined in the seminal work [94] in reference to the bootstrap [70], a standard statistical procedure that also involves sampling with replacement. The optimal particle filter is discussed, and further references given, in [59]; see section IID. While the term "optimal" particle filter is common in the literature, the sense in which this filter is optimal is indeed rather weak. We refer to [49, Chapter 10], which utilizes instead the term "guided" particle filter, for further discussion on this topic; in particular, [49, Theorem 10.1] formalizes the criterion of *local optimality* and the ensuing discussion provides compelling criticisms of this notion of optimality.

The proof of convergence of the bootstrap particle filter presented here originates in [189] and can also be found in [207]. We refer to [207, Chapter 12] for a similar proof of convergence for a Gaussianized optimal particle filter, and to [118] for a convergence analysis of the optimal particle filter. Throughout much of this chapter, we considered for simplicity the case of Gaussian additive noise and linear observation operator. More general convergence and stability results for sequential Monte Carlo methods can be found, for instance, in [54] and [49, Chapter 11].

For problems in which the dynamics model is nonlinear and the state space of the hidden

process is low dimensional, particle filters are enormously successful. Generalizing them so that they work for the high-dimensional problems that arise, for example, in geophysical applications, provides a major challenge, since in those settings the particle weights typically concentrate on one, or a small number, of particles —see for instance [24, 215, 214, 3]. Weight collapse is intimately related to the fact that, as discussed in Chapter 3, importance sampling weights have large variance when the target and the proposal distributions are far apart. As noted in [216, 3], an advantage of the optimal particle filter over the bootstrap particle filter is that the optimal filter utilizes a proposal that is closer to the target, which helps alleviate weight collapse over each iteration of the filter. Resampling techniques to turn weighted samples into unweighted ones are essential to the success of particle filters over multiple iterations. We refer to [49, Chapter 9] for an in-depth exploration and comparison of resampling schemes in sequential Monte Carlo.

An important algorithmic idea in sequential Monte Carlo, as in many other algorithms studied in these notes, is to introduce auxiliary variables to accelerate computation [185, 118]. In addition, exploiting decay of correlations through *localization* is often needed when implementing particle filters for online estimation of discretizations of spatial fields. A review of local particle filters can be found in [73]. The paper [189] investigates, from a theoretical viewpoint, whether localization can help to beat the curse of dimension. Attempts to bridge particle filters with ensemble Kalman filters to alleviate the curse of dimension include [76, 217], and the relation between the collapse of ensemble and particle methods is investigated in the paper [165], which also emphasizes the importance of localization. Further theoretical investigations of localization for ensemble Kalman methods include [6, 5]. We close by pointing out that sequential Monte Carlo methods can be combined in several ways with algorithms studied in previous chapters, and we refer to [10] for a representative, important example.

Chapter 10

Variational Inference and Expectation Maximization

This chapter is concerned with two important computational techniques: Variational Inference (VI) and the Expectation Maximization (EM) algorithm. In contrast to the methods studied in previous chapters, VI and EM are deterministic algorithms which do not rely on sampling. The goal of VI is to find a tractable *variational distribution* that is close to a given intractable target distribution, so that quantities of interest that involve the target can be approximated using the variational distribution. The EM algorithm is a deterministic optimization method that is particularly useful when the objective function can be simplified by introducing auxiliary variables. This chapter showcases the unifying ideas underlying VI and the EM algorithm, and how these techniques can be combined with, or used as alternatives of, Monte Carlo methods.

VI is gaining popularity as an alternative to MCMC, and it is worth highlighting the conceptual differences between both approaches. MCMC algorithms approximate the target distribution by generating samples from a Markov chain that has the desired target as a limit distribution. However, in practice it is challenging to determine for how long one needs to run the chain to reach the stationary regime, and this can depend heavily on the choice of proposal kernel. MCMC methods also tend to be computationally intensive. On the other hand, VI is often much faster, but does not enjoy the same convergence guarantees, as it simply seeks a density "close enough" to the target from a given family. If the target cannot be well approximated within this family (which is often the case), accurate approximations will not be possible. Thus, VI is suited for instance for Bayesian inference with large datasets or for problems where we want to quickly explore many models; MCMC is suited to smaller data sets and scenarios where we are willing to pay a heavier computational cost to achieve a higher accuracy.

The main idea behind the EM algorithm is to introduce auxiliary variables to simplify the optimization of an intractable objective function. Each EM iteration consists of two steps, referred to as E-step and M-step. In the E-step, we compute an expectation to find a lower bound on the objective function. In the M-step, we maximize this lower bound to produce a new optimization iterate. If the expectation in the E-step is intractable, Monte Carlo methods can be used to approximate it. EM is widely used, for instance, to compute maximum likelihood estimators in applications where the likelihood function can be simplified by introducing latent variables. The role of EM in maximum likelihood estimation is similar to the role of Gibbs samplers in posterior sampling; indeed, EM can be used to initialize Gibbs samplers for Bayesian inference.

This chapter is organized as follows. Section 10.1 is concerned with VI, emphasizing a computationally efficient coordinate-wise implementation. Section 10.2 contains an introduction to the EM algorithm, deriving the method using the same ideas that underpin VI. We close in Section 10.3 with bibliographical remarks.

10.1 ▪ Variational Inference

The goal of Variational Inference (VI) is to find a tractable distribution g^* that is close to a given target distribution f, so that quantities of interest with respect to the target can be cheaply computed using the variational distribution g^*. For instance, once g^* is found, expected values with respect to the target can be approximated as follows:

$$\mathcal{I}_f[h] = \int h(x)f(x)\,dx \approx \int h(x)g^*(x)\,dx = \mathcal{I}_{g^*}[h],$$

which is useful if computing $\mathcal{I}_{g^*}[h]$ is cheaper than computing $\mathcal{I}_f[h]$.

The variational distribution g^* is defined as the (numerical) solution to an optimization problem. Precisely, one specifies a family $\mathscr{D}$ of tractable distributions and sets

$$g^* = \arg\min_{g\in\mathscr{D}} d_{\text{KL}}(g\|f)\,, \tag{10.1}$$

where d_{KL} is the Kullback-Leibler (KL) divergence between p.d.f.s. Thus, g^* is the p.d.f. in $\mathscr{D}$ which is closest to the target in KL-divergence, as represented in Figure 10.1. The family $\mathscr{D}$ of admissible distributions needs to be carefully chosen, since it determines the accuracy and computational cost of the methodology. We will introduce the KL-divergence in Subsection 10.1.1, and we will discuss in Subsection 10.1.2 the choice of the family $\mathscr{D}$ along with a Coordinate Ascent Variational Inference (CAVI) algorithm to solve the optimization problem (10.1).

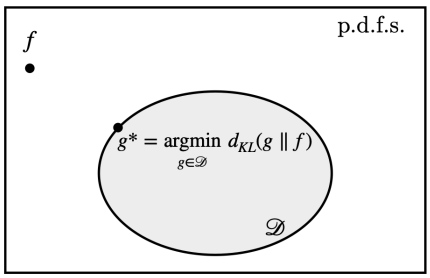


Figure 10.1. *Illustration of the variational distribution g^*.*

10.1.1 ▪ Kullback-Leibler Divergence and Evidence Lower-Bound

In this section, we motivate the choice of the KL-divergence to quantify closeness between p.d.f.s in (10.1). Recall that the KL-divergence between p.d.f.s g and f is given by

$$d_{\text{KL}}(g\|f) = \mathbb{E}_g\left[\log\left(\frac{g}{f}\right)\right],$$

provided that g is absolutely continuous with respect to f, and otherwise $d_{\text{KL}}(g\|f) = \infty$. Note that the KL-divergence satisfies $d_{\text{KL}}(g\|f) \geq 0$ with equality iff $g = f$. However, the KL-divergence is not symmetric, does not satisfy the triangle inequality, and may take value infinity. Therefore, it is not a distance in the space of p.d.f.s.

An important motivation to work with the KL-divergence is that the optimization problem of finding the p.d.f. that minimizes the KL-divergence with the target can be reformulated into an equivalent optimization problem which does not require the target to be normalized. Indeed, we

will show that minimizing KL-divergence is equivalent to maximizing the evidence lower-bound (ELBO), which we now introduce.

Definition 10.1 (Evidence Lower-Bound). *The ELBO of a p.d.f. g with respect to an unnormalized p.d.f. $\tilde{f}$ is given by*

$$\text{ELBO}(g) := \mathbb{E}_g[\log \tilde{f}] - \mathbb{E}_g[\log g]. \qquad \square$$

The next result shows that the KL-divergence is equal, up to an additive constant, to the negative ELBO. As a consequence, minimizing the KL-divergence is equivalent to maximizing the ELBO.

Theorem 10.2 (Relationship Between KL-Divergence and ELBO). *Let $f(x) = \tilde{f}(x)/c$ be a p.d.f. and let $\text{ELBO}(g)$ denote the ELBO of a p.d.f. g with respect to the unnormalized p.d.f. $\tilde{f}$. Then,*

$$d_{\text{KL}}(g\|f) = -\text{ELBO}(g) + \log c.$$

Proof. By direct calculation:

$$\begin{aligned} d_{\text{KL}}(g\|f) &= \mathbb{E}_g\left[\log\left(\frac{g}{f}\right)\right] \\ &= \mathbb{E}_g\left[\log g\right] - \mathbb{E}_g\left[\log f\right] \\ &= \mathbb{E}_g\left[\log g\right] - \mathbb{E}_g\left[\log\left(\frac{\tilde{f}}{c}\right)\right] \\ &= \mathbb{E}_g\left[\log g\right] - \mathbb{E}_g\left[\log \tilde{f}\right] + \mathbb{E}_g\left[\log c\right] \\ &= -\text{ELBO}(g) + \log c\,. \end{aligned}$$

$\square$

Theorem 10.2 implies that the optimization problem (10.1) can be reformulated in terms of maximizing the ELBO. The following corollary shows that the ELBO provides a lower-bound on the (log)-evidence, which motivates its name. We will discuss the interpretation of the normalizing constant c as model evidence in Example 10.4 below.

Corollary 10.3 (Lower Bound Property of ELBO). *In the setting of Theorem 10.2, it holds that*

$$\text{ELBO}(g) \leq \log c.$$

Proof. Since the KL-divergence is non-negative and $d_{\text{KL}}(g\|f) + \text{ELBO}(g) = \log c$, the result follows. $\square$

Example 10.4 (VI in Bayesian Statistics). In Bayesian statistics, inference over a parameter θ given data y is based on the posterior distribution

$$f(\theta|y) = \frac{1}{f(y)} f(y|\theta) f(\theta),$$

which combines the likelihood $f(y|\theta)$ with a prior distribution $f(\theta)$. Computing posterior expectations can be expensive when the dimension of θ is large. In such a case, computing the normalizing constant $f(y)$, called the evidence, can also be challenging, and it is natural to view $\tilde{f}(\theta) := f(y|\theta)f(\theta)$ as an unnormalized target distribution with unknown normalizing constant $c := f(y)$. Corollary 10.3 then shows that $\text{ELBO}(g) \leq \log f(y)$. This inequality explains the name *evidence lower-bound*: $\text{ELBO}(g)$ gives a lower bound on the log-evidence $\log f(y)$. In addition, this inequality motivates a crude but cheap model selection criteria: since a large evidence suggests good model fit, one may choose the model that gives the highest ELBO.

In this Bayesian context, it is insightful to rewrite the ELBO as follows

$$\begin{aligned}\text{ELBO}(g) &= \mathbb{E}_g\left[\log f(y|\theta)\right] + \mathbb{E}_g\left[\log f(\theta)\right] - \mathbb{E}_g\left[\log g(\theta)\right] \\ &= \mathbb{E}_g\left[\log f(y|\theta)\right] + \mathbb{E}_g\left[\log\left(\frac{f(\theta)}{g(\theta)}\right)\right] \\ &= \mathbb{E}_g\left[\log f(y|\theta)\right] - d_{\text{KL}}\big(g(\theta)\|f(\theta)\big).\end{aligned}$$

The first term in the last displayed equation is the expected log-likelihood, whereas the second is the negative KL-divergence between the prior and the variational density q. From the above formulation, we obtain an intuition that the optimal q should find a compromise between maximizing the expected log-likelihood and being close to the prior distribution. The first term promotes matching the data, and the second term promotes matching prior beliefs. □

10.1.2 ▪ Mean-Field Approximation and Coordinate Ascent Variational Inference

This section discusses the choice of the family $\mathscr{D}$ of admissible distributions in (10.1). Ideally, the family $\mathscr{D}$ should be chosen so that:

(i) there is $q \in \mathscr{D}$ that accurately approximates the target;

(ii) expectations with the variational distribution q^* can be computed efficiently. Thus, $\mathscr{D}$ should contain tractable distributions;

(iii) the optimization problem (10.1) can be solved efficiently.

In practice, a compromise needs to be made: enlarging the family $\mathscr{D}$ potentially helps in the approximation accuracy, but it typically leads to a harder optimization problem. A popular choice of $\mathscr{D}$ is the mean-field family, which sacrifices the ability to accurately represent target distributions with highly correlated coordinates, but leads to efficient optimization and inference algorithms.

Definition 10.5 (Mean-Field Family). *A p.d.f. $g(x)$ with $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ is said to belong to the mean-field family if its marginals are independent, that is, if it can be written in the form*

$$g(x) = \prod_{i=1}^{d} g_i(x_i)\,,$$

where g_i is a one dimensional p.d.f. over the i-th coordinate x_i.

In other words, the mean-field assumption requires that the variational density can be factorized as a product of distributions. We work with the case where g factorizes as a product

of one-dimensional distributions, but it is also possible to consider more general cases where it factorizes into a product of larger groups of coordinates.

Pros and Cons: As we shall see, solving the optimization problem (10.1) is relatively straightforward when using the mean-field family. While restrictive, the mean-field family still affords some flexibility as the marginals g_i are left unconstrained. If required, we can impose a specific parametric family on these marginals as well. The main caveat of the mean-field family is that it cannot capture the dependence structure of the different variables, and estimates computed with a mean-field distribution will be far off when the target has highly correlated variables. In that case, marginal variances can be severely over or under-estimated.

Remark 10.6. *The mean-field assumption has its roots in statistical physics, where 'mean-field' refers to simplifying difficult high-dimensional stochastic models by considering simpler ones which ignore second-order effects. While we focus on the mean-field variational family due to its popularity and relative simplicity, it is possible to go beyond this restrictive assumption by incorporating dependencies between the marginals or by adding new latent variables. Both of these approaches reduce the bias of VI, but can substantially increase the difficulty of the associated optimization problem* (10.1).

Under the mean-field approximation, each g_i can be optimized individually, and this approach is known as CAVI. We next prove a theorem that will lead to this technique. Let us set some notation. We denote by x_i the i-th component of $x \in \mathbb{R}^d$. We denote by $x_{-i} \in \mathbb{R}^{d-1}$ the vector obtained by removing the i-th component of x. To highlight the dependence of $f(x)$ on x_i and x_{-i}, we write $f(x_i, x_{-i})$ with abuse of notation. Finally, g_{-i} will denote a p.d.f. over the variables x_{-i}.

Theorem 10.7 (Coordinate-Wise ELBO Maximization). *Let $g(x) = \prod_{i=1}^d g_i(x_i)$ and let $g_{-i}(x_{-i}) = \prod_{j \neq i} g_j(x_j)$ for fixed g_j, $j \neq i$. Then, the g_i that maximizes* $\text{ELBO}(g)$ *is given by*

$$g_i^*(x_i) \propto \exp\Big(\mathbb{E}_{g_{-i}}\left[\log f(x_i, x_{-i})\right]\Big). \tag{10.2}$$

Proof. Throughout this proof, we denote by C a constant which does not depend on g_i and may change from line to line. We will show that

$$\text{ELBO}(g) = -d_{\text{KL}}(g_i \| g_i^*) + C,$$

where g_i^* is given by (10.2). This implies the desired result, since the g_i that minimizes the KL-divergence in the right-hand side is clearly $g_i = g_i^*$.

First, we show that

$$\text{ELBO}(g) = \mathbb{E}_{g_i}\left[\mathbb{E}_{g_{-i}}\left[\log\ f(x_i, x_{-i})\right] - \log g_i(x_i)\right] + C.$$

Indeed,

$$\begin{aligned}
\text{ELBO}(g) &= \mathbb{E}_g\left[\log\ f(x)\right] - \mathbb{E}_g\left[\log g(x)\right] \\
&= \mathbb{E}_g\left[\log\ f(x_i, x_{-i})\right] - \sum_i \mathbb{E}_{g_i}\left[\log g_i(x_i)\right] \\
&= \mathbb{E}_{g_i}\left[\mathbb{E}_{g_{-i}}\left[\log\ f(x_i, x_{-i}) | x_i\right]\right] - \mathbb{E}_{g_i}\left[\log g_i(x_i)\right] + C,
\end{aligned}$$

and the first term in the right-hand side can be simplified using that

$$\begin{aligned}\mathbb{E}_{g_{-i}}\left[\log f(x_i, x_{-i})|x_i\right] &= \int_{\mathbb{R}^{d-1}} \log f(x_i, x_{-i}) \cdot g(x_{-i}|x_i)\, dx_{-i} \\ &= \int_{\mathbb{R}^{d-1}} \log f(x_i, x_{-i}) \cdot g(x_{-i})\, dx_{-i} \\ &= \mathbb{E}_{g_{-i}}\left[\log f(x_i, x_{-i})\right].\end{aligned}$$

Therefore,

$$\begin{aligned}\text{ELBO}(g) &= \mathbb{E}_{g_i}\left[\mathbb{E}_{g_{-i}}\left[\log\ f(x_i, x_{-i})\right] - \log g_i(x_i)\right] + C \\ &= \mathbb{E}_{g_i}\left[\log\Big(\exp\big(\mathbb{E}_{g_{-i}}\left[\log\ f(x_i, x_{-i})\right]\big)\Big) - \log g_i(x_i)\right] + C \\ &= -\mathbb{E}_{g_i}\left[\log\left(\frac{g_i(x_i)}{\exp\big(\mathbb{E}_{g_{-i}}\left[\log\ f(x_i, x_{-i})\right]\big)}\right)\right] + C \\ &= -d_{\text{KL}}(g_i \| g_i^*) + C,\end{aligned}$$

and the proof is complete. □

Theorem 10.7 gives us an expression for the optimizer of the ELBO with respect to g_i holding the other coordinates fixed. This procedure can be repeated iteratively, optimizing each coordinate and holding the others fixed. The resulting algorithm is known as Coordinate Ascent Variational Inference (CAVI):

Algorithm 10.1 Coordinate Ascent Variational Inference (CAVI)

1: **Input**: Unnormalized target density $\tilde{f}$.
2: **Initialization**: Choose initial variational densities g_i for each $i = 1, \ldots, d$.
3: While the ELBO/KL-divergence has not converged, do:
4: Update, for each $i = 1, \ldots, d$,

$$g_i(x_i) \propto \exp\Big(\mathbb{E}_{g_{-i}}\left[\log \tilde{f}(x_i, x_{-i})\right]\Big).$$

5: **Output**: $g(x) = \prod_{i=1}^{d} g_i(x_i) \approx f(x)$.

In each update, CAVI lowers the KL-divergence between q^* and the target f, and hence one can expect convergence to a local minimizer under suitable assumptions. The ELBO is not necessarily concave and therefore there is in general no guarantee to reach a maximizer of the ELBO, or, equivalently, a minimizer of the KL-divergence.

Pros and Cons: Each CAVI update lowers the KL-divergence, which facilitates introducing stopping criteria for the algorithm. If the factors g_i are assumed to be in the exponential family, computing each factor update is simplified significantly. However, since CAVI relies on the mean-field assumption, it inherits all its disadvantages. In addition, for some target distributions it may be computationally expensive to compute the updates of the variational factors. In such a case, other optimization methods (e.g. stochastic optimization and natural gradient methods) rather than CAVI can be used to maximize the ELBO.

Remark 10.8. *CAVI shares a similar structure with the Gibbs sampler. To see why, notice that*

$$g_i^*(x_i) \propto \exp\Big(\mathbb{E}_{g_{-i}}\left[\log f(x_i, x_{-i})\right]\Big) \propto \exp\Big(\mathbb{E}_{g_{-i}}\left[\log f(x_i|x_{-i})\right]\Big).$$

In Gibbs sampling, we update each coordinate by sampling the corresponding full conditional. CAVI uses full conditionals as well, with each density g_i set to be proportional to the exponential of the expectation of the log of the full conditional.

10.2 ▪ Expectation Maximization Algorithm

In its most basic form, the Expectation Maximization (EM) algorithm is a deterministic optimization method. EM is widely used to compute maximum likelihood estimators in problems where introducing a latent variable makes the likelihood function easier to optimize.

Suppose for concreteness that we want to find the maximum likelihood estimator

$$\widehat{\theta} = \arg\max_{\theta} f(y|\theta)$$

of a parameter θ given observed data y. Suppose that $f(y|\theta)$ is the marginal of a complete likelihood $f(y, z|\theta)$, which includes the data y and an auxiliary variable z. Then, for any p.d.f. g with compatible support, it holds by Jensen's inequality that

$$\begin{aligned}
\log f(y|\theta) &= \log \int f(y, z|\theta)\, dz \\
&= \log \int \frac{f(y, z|\theta)}{g(z)} g(z)\, dz \\
&\geq \int \log\left(\frac{f(y, z|\theta)}{g(z)}\right) g(z)\, dz \\
&= \mathbb{E}_g[\log f(y, z|\theta)] - \mathbb{E}_g[\log(g)] = \text{ELBO}(g, \theta).
\end{aligned}$$

Here we view the ELBO as a function of the auxiliary distribution g and the parameter θ in the density $f(y, z|\theta)$. Now, notice that we have

$$\begin{aligned}
\text{ELBO}(g, \theta) &= \int \log\left(\frac{f(y, z|\theta)}{g(z)}\right) g(z)\, dz \\
&= \int \log\left(\frac{f(z|y, \theta)}{g(z)}\right) g(z)\, dz + \int \log\big(f(y|\theta)\big) g(z)\, dz \\
&= -d_{\text{KL}}\big(g(z) \| f(z|y, \theta)\big) + \log f(y|\theta).
\end{aligned}$$

In other words,

$$\log f(y|\theta) = \text{ELBO}(g, \theta) + d_{\text{KL}}\big(g(z) \| f(z|y, \theta)\big). \tag{10.3}$$

The above derivations are direct consequences of Theorem 10.2 and Corollary 10.3 with target $f(z) \equiv f(z|y, \theta)$, unnormalized density $\tilde{f}(z) \equiv f(y, z|\theta)$, and normalizing constant $c \equiv f(y|\theta)$.

Equation (10.3) motivates an iterative approach to maximize the log-likelihood $\log f(y|\theta)$, which is equivalent to maximizing the likelihood $f(y|\theta)$ since the logarithm is an increasing function. Given the current iterate θ_ℓ, we obtain the next iterate $\theta_{\ell+1}$ in two steps, maximizing in turn the two components of the ELBO; the approach is summarized in Figure 10.2.

1. First, we find $g_\ell(z)$ by maximizing $\text{ELBO}(g, \theta_\ell)$ over p.d.f. g. From (10.3),

$$\log f(y|\theta_\ell) = \text{ELBO}(g, \theta_\ell) + d_{\text{KL}}\big(g \| f(z|y, \theta_\ell)\big),$$

and it follows that maximizing $\text{ELBO}(g, \theta_\ell)$ or minimizing $d_{\text{KL}}\big(g \| f(z|y, \theta_\ell)\big)$ over g are equivalent. Since the KL-divergence is minimized when both arguments agree, we obtain that $g_\ell(z) = f(z|y, \theta_\ell)$.

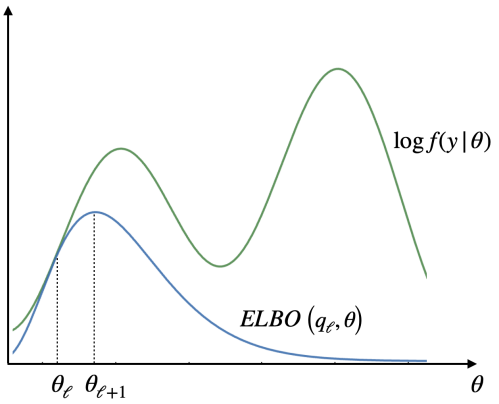


Figure 10.2. *One iteration of the EM algorithm.*

2. Second, we find $\theta_{\ell+1}$ by maximizing $\text{ELBO}(g_\ell, \theta)$ over θ. Note that

$$\text{ELBO}(g_\ell, \theta) = \int \log f(y, z|\theta) f(z|y, \theta_\ell)\, dz + \text{const}, \tag{10.4}$$

where const is independent of θ. Hence, the quantity to maximize is the expected value of the joint log-density $\log f(y, z|\theta)$ with respect to $g_\ell(z) = f(z|y, \theta_\ell)$.

Combining these two steps gives the EM algorithm, summarized below.

Algorithm 10.2 Expectation Maximization (EM)

1: **Input**: Initialization θ_0, number of iterations L.
2: For $\ell = 0, 1, \ldots, L-1$ do the following expectation and maximization steps:
3: **E-Step**: Compute

$$\mathbb{E}_{Z\sim f(z|y,\theta_\ell)}\Big[\log f(y, Z|\theta)\Big] = \int \log f(y, z|\theta) f(z|y, \theta_\ell)\, dz.$$

4: **M-Step**: Compute

$$\theta_{\ell+1} = \arg\max_\theta \mathbb{E}_{Z\sim f(z|y,\theta_\ell)}\Big[\log f(y, Z|\theta)\Big].$$

5: **Output**: Parameter θ^L.

The following result shows that the EM algorithm has the desirable property that the likelihood function increases monotonically along iterates θ_ℓ.

Theorem 10.9 (Monotonic Increase of Likelihood along EM Iterates). *Let $\{\theta_\ell\}_{\ell=0}^{L-1}$ be the iterates of Algorithm 10.2. Then, for $0 \le \ell \le L-1$, it holds that*

$$\log f(y|\theta_\ell) \le \log f(y|\theta_{\ell+1}). \tag{10.5}$$

Proof. Let $g_\ell(z) = f(z|y, \theta_\ell)$. Using the log-likelihood characterization in (10.3), it holds that

$$\begin{aligned}\log f(y|\theta_{\ell+1}) &= \text{ELBO}(g_\ell, \theta_{\ell+1}) + d_{\text{KL}}\big(g_\ell \| f(z|y, \theta_{\ell+1})\big) \\ &\geq \text{ELBO}(g_\ell, \theta_\ell) + d_{\text{KL}}\big(g_\ell \| f(z|y, \theta_{\ell+1})\big) \\ &\geq \text{ELBO}(g_\ell, \theta_\ell) + d_{\text{KL}}\big(g_\ell \| f(z|y, \theta_\ell)\big) \\ &= \log f(y|\theta_\ell).\end{aligned}$$

The first inequality follows because $\theta_{\ell+1} = \arg\max_\theta \text{ELBO}(g_\ell, \theta)$; the second inequality follows because $d_{\text{KL}}\big(g_\ell \| f(z|y, \theta_\ell)\big) = 0$ since $g_\ell = f(z|y, \theta_\ell)$, while $d_{\text{KL}}\big(g_\ell \| f(z|y, \theta_{\ell+1})\big) \geq 0$. □

Remark 10.10. *Under mild additional assumptions, it follows from Theorem 10.9 that the iterates θ_ℓ of the EM algorithm converge, as $\ell \to \infty$, to a local maximizer of the likelihood function. It is important to note, however, that the expectation in the E-step and the optimization in the M-step are often intractable. Monte Carlo algorithms may be employed to approximate the E-step, and optimization algorithms to approximate the M-step. Such approximations can cause loss of monotonicity and convergence guarantees.*

Pros and Cons: The EM algorithm is a workhorse in computational mathematics and statistics. It is widely used to compute maximum likelihood estimators in mixture models, hierarchical models, hidden Markov models, etc. For some problems, the M step can be solved in closed form, in which case the EM algorithm can be an efficient derivative-free optimization method. A caveat of the EM algorithm is that the choice of initialization is important, as the algorithm converges to local maxima. In addition, for some problems computing the E and M steps can be computationally expensive, and the convergence can be slow.

Example 10.11 (Dempster et al. 1977). Observations $y = (y_1, y_2, y_3, y_4)$ are gathered from the multinomial distribution

$$\text{Multinomial}\left(K; \frac{1}{2} + \frac{\theta}{4}, \frac{1}{4}(1-\theta), \frac{1}{4}(1-\theta), \frac{\theta}{4}\right).$$

Estimation is simplified if y_1 is split into two cells $z = (z_1, z_2)$ so we create the augmented model

$$(z_1, z_2, y_2, y_3, y_4) \sim \text{Multinomial}\left(K; \frac{1}{2}, \frac{\theta}{4}, \frac{1}{4}(1-\theta), \frac{1}{4}(1-\theta), \frac{\theta}{4}\right),$$

with $y_1 = z_1 + z_2$. Here

$$\begin{aligned}f(y, z|\theta) &\propto \theta^{z_2+y_4}(1-\theta)^{y_2+y_3}, \\ f(y|\theta) &\propto (2+\theta)^{y_1}\theta^{y_4}(1-\theta)^{y_2+y_3}.\end{aligned}$$

We have (up to constants)

$$\begin{aligned}\mathbb{E}_{Z\sim f(z|y,\theta_\ell)}\left[\log f(y, Z|\theta)\right] &= \mathbb{E}\left[(Z_2 + y_4)\log\theta + (y_2 + y_3)\log(1-\theta)\right] \\ &= \left(\frac{\theta_\ell}{2+\theta_\ell}y_1 + y_4\right)\log\theta + (y_2 + y_3)\log(1-\theta),\end{aligned}$$

where we used that

$$Z_2|y, \theta_\ell \sim \text{Binomial}\left(y_1, \frac{\frac{\theta_\ell}{4}}{\frac{1}{2} + \frac{\theta_\ell}{4}}\right) = \text{Binomial}\left(y_1, \frac{\theta_\ell}{2+\theta_\ell}\right).$$

The expectation $\mathbb{E}_{Z\sim f(z|y,\theta_\ell)}\left[\log f(y,Z|\theta)\right]$ can then be maximized over θ to give

$$\theta_{\ell+1} = \frac{\frac{\theta_\ell y_1}{2+\theta_\ell} + y_4}{\frac{\theta_\ell y_1}{2+\theta_\ell} + y_2 + y_3 + y_4}. \qquad \square$$

Monte Carlo EM In practice, computing the E-step may be computationally difficult, and Monte Carlo can be used to approximate this expectation:

$$\mathbb{E}_{Z\sim f(z|y,\theta_\ell)}\left[\log f(y,Z|\theta)\right] \approx \frac{1}{N}\sum_{n=1}^{N} \log f(y, Z^{(n)}|\theta),$$

where

$$Z^{(1)}, \ldots, Z^{(N)} \overset{\text{i.i.d.}}{\sim} f(z|y,\theta_\ell).$$

Example 10.12 (Monte Carlo EM for Dempster et al. 1977). In the setting of Example 10.11 there is no need to use Monte Carlo EM, since the expectation can be computed analytically using that $Z_2|y,\theta_\ell \sim \text{Binomial}\left(y_1, \frac{\theta_\ell}{2+\theta_\ell}\right)$. However, merely for illustration of the technique, we note that the Monte Carlo version would define

$$\theta_{\ell+1} = \frac{\bar{z}_N + y_4}{\bar{z}_N + y_2 + y_3 + y_4},$$

where

$$\bar{z}_N = \frac{1}{N}\sum_{n=1}^{N} Z^{(n)}, \quad Z^{(1)}, \ldots, Z^{(N)} \overset{\text{i.i.d.}}{\sim} \text{Binomial}\left(y_1, \frac{\theta_\ell}{2+\theta_\ell}\right). \qquad \square$$

10.3 ▪ Discussion and Bibliography

We refer to [28, 29] for accessible introductions to VI that have inspired the presentation in this chapter. For a more in-depth exploration of VI, see [119, 232]. The paper [112] contains a modern approach to VI with applications to massive text data-sets. Applications to graphical models, and a generalization of the mean-field family to allow for interactions, are discussed in [119]. While CAVI provides a simple framework for ELBO maximization under the mean-field family, computing the CAVI updates can be challenging for highly complex targets, in which case alternative optimization methods based on stochastic VI or autodifferentiation may be required [112, 131]. Beyond the mean-field family, other simple but practical choices of variational family include Gaussians [207] and Gaussian mixtures [133].

Our presentation has solely focused on VI techniques that seek to minimize the KL-divergence [132]. The works [141, 110] consider VI using the broad family of Renyi-alpha divergences, which includes for instance the KL-divergence, the χ^2-divergence, and the Hellinger distance [86]. However, as emphasized in this chapter, the KL-divergence holds a special place in VI, since among a wide class of divergences it is the unique choice for which minimization of the objective does not require knowledge of the normalizing constant of the target [47]. For sampling

problems, the paper [79] explores gradient flows arising from minimizing various objectives, and shows that the choice of KL-divergence also leads to fundamental practical advantages.

An overarching idea behind VI, the EM algorithm, and other popular computational methods such as variational auto-encoders [123], is to minimize KL-divergence by maximizing the ELBO. The ELBO functional is typically non-convex, even for simple potentials; see for instance the discussion in [4]. Due to this lack of convexity, it is often challenging to develop theory for VI outside restricted settings on the target distribution. For this reason, the study of statistical and optimization convergence guarantees for VI is still an active area of research, see e.g. [241] for statistical convergence rates and [133] for a theoretical framework that relies on gradient flows.

The EM algorithm was introduced in [55], although particular instances had been previously considered on numerous occasions, see e.g. [43, 106]. Convergence of the EM algorithm was established in [238]. We refer to [158] for a textbook on the EM algorithm and to [27, 107, 152, 28] for gentle introductions. EM is an example of a minimization-maximization method [115], a broad family of algorithms with rich theoretical underpinnings. Several modifications to the original EM algorithm have been proposed to speed up its convergence. We refer to [159, 143, 117, 172] for some important methodological developments and to [160] for a survey written on the 20th anniversary of the original paper [55].

Appendix A
Markov Chain Theory

This appendix contains an informal review of Markov chains. The aim is to describe briefly and intuitively some of the main concepts and ideas that are important to understand Markov chain Monte Carlo algorithms.

A.1 ▪ Basic Definitions

Definition A.1 (Markov Chain). *A collection* $\mathbb{X} = \{X_n,\, n = 0, 1, 2, \ldots\}$ *of random variables taking values in a state space* E *is called a* Markov chain *if*

$$\mathbb{P}(X_{n+1} \in A_{n+1} \,|\, X_n \in A_n, \ldots,\, X_0 \in A_0) = \mathbb{P}(X_{n+1} \in A_{n+1} \,|\, X_n \in A_n)$$

for all $n \geq 0$, *and all measurable* $A_{n+1}, A_n, \ldots, A_0 \subset E$.

The index n is often interpreted as time, and the Markov property then states that the future (time $n + 1$) is independent of the past (times $\leq n - 1$) given the present (time n).

Definition A.2 (Time Homogeneous Markov Chain; Markov Kernel). *A Markov chain* $\mathbb{X}$ *is called* time homogeneous *if its transition probabilities*

$$P(x, A) = \mathbb{P}(X_{n+1} \in A \,|\, X_n = x)$$

do not depend on n. $P(x, A)$ *is called the transition kernel or* Markov kernel *of* $\mathbb{X}$.

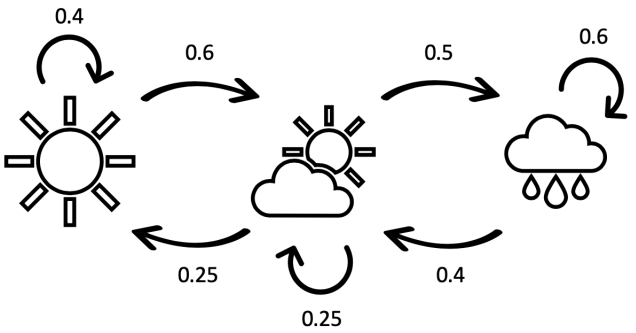
Here and below, the expression $\mathbb{P}(X_{n+1} \in A \,|\, X_n = x)$ is only formal if $\mathbb{P}(X_n = x) = 0$, but it can be made rigorous using regular conditional probability. From now on, all Markov chains that we consider will be time homogeneous. We will work mainly with transition kernels that are absolutely continuous with respect to the Lebesgue measure or with respect to the counting measure (an exception will be the Metropolis Hastings kernel studied in Chapter 4). This means that, for every x, there is an associated p.d.f. (or p.m.f. for the discrete case) $p(x, z)$ such that

$$P(x, A) = \int_A p(x, z)\, dz.$$

With slight abuse of terminology, we will often call $p(x, z)$ a Markov kernel.

Example A.3 (Weather Markov Chain). Suppose $X_n \equiv$ weather at day $n = \begin{cases} 0 & \text{if sunny,} \\ 1 & \text{if cloudy,} \\ 2 & \text{if rainy.} \end{cases}$

The state space is $E = \{0, 1, 2\}$. We may specify (time homogeneous) transition probabilities graphically as follows:



.

□

Definition A.4 (n-th Step Transition Densities). *The* n-th step transition probability densities $p^n(x, z)$ *are defined by*

$$\mathbb{P}(X_n \in A \,|\, X_0 = x) = P^n(x, A) = \int_A p^n(x, z)\, dz. \qquad \square$$

If the state space is finite, say $E = \{0, 1, \ldots, s\}$ for some integer s, then we can collect the transition probabilities in the *transition matrix*

$$P = [p_{ij}]_{i,j \in E},$$

where $p_{ij} = \mathbb{P}(X_{n+1} = j \,|\, X_n = i)$ for $(i, j) \in E \times E$. It holds that

$$p^n(i, j) = (P^n)_{ij}.$$

Lemma A.5. *If* $X_0 \sim \pi_0$ *and* $\mathbb{X}$ *has* n*-th step transition probability densities* $p^n(x, z)$*, then the p.d.f. of* X_n *is given by*

$$\pi_n(x) = \int_E \pi_0(z) p^n(z, x)\, dz.$$

If the state space E *is finite, then* π_n *is the probability vector given by* $\pi_n = \pi_0 P^n$.

Example A.6 (Weather Markov Chain: 2-Step Probabilities). The transition matrix of the weather chain is given by

$$P = \begin{bmatrix} 0.4 & 0.6 & 0 \\ 0.25 & 0.25 & 0.5 \\ 0 & 0.4 & 0.6 \end{bmatrix}.$$

Suppose day 0 is sunny. Then,

$$\pi_0 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.$$

The distribution of the weather on day 2 is

$$\pi_2 = \pi_0 P^2 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} P^2 = \begin{bmatrix} 0.31 & 0.39 & 0.3 \end{bmatrix}.$$

So if day 0 is sunny, then day 2 has a 0.31 probability of being sunny. □

A.2 ▪ Invariant Distributions: General Balance and Detailed Balance

In this section, we discuss the notion of statistical equilibrium, which is central to the theory of Markov chains.

Definition A.7 (General Balance Equation; Invariant Distribution). *A Markov kernel $p(x, z)$ satisfies the* general balance equation *with respect to π if*

$$\pi(x) = \int_E \pi(z) p(z, x)\, dz.$$

We then say that π is an invariant distribution *of the Markov kernel $p(x, z)$.*

Note that if E is discrete, the general balance equation is simply $\pi = \pi P$.

Definition A.8 (Ergodic Markov Chain). *A Markov chain $\mathbb{X}$ is called* ergodic *if there exists a distribution π such that, for all $A \subset E$ and initial distributions π_0, it holds that*

$$\lim_{n\to\infty} \mathbb{P}(X_n \in A) = \pi(A).$$

We say that π is the limit distribution of $\mathbb{X}$.

Lemma A.9. *Let $\mathbb{X}$ be an ergodic Markov chain with Markov kernel $p(x, z)$ and limit distribution π. Then, π is an invariant distribution for $p(x, z)$. In particular, if $\mathbb{X}$ is initialized at statistical equilibrium ($X_0 \sim \pi$), then $X_n \sim \pi$ for all $n \geq 0$.*

Proof. [Sketch Proof] By the dominated convergence theorem

$$\begin{aligned} \lim_{n\to\infty} \pi_{n+1}(x) &= \lim_{n\to\infty} \int_E \pi_n(z) p(z, x)\, dz \\ &= \int_E \lim_{n\to\infty} \pi_n(z) p(z, x)\, dz, \end{aligned}$$

which shows that $\pi(x) = \int_E \pi(z) p(z, x)\, dz$. The final claim of the lemma is proved by induction.

The general balance equation implies that the transition probabilities of $\mathbb{X}$ preserve equilibrium. For this reason, we call a distribution π that satisfies the general balance equation (A.7) an invariant distribution of the chain $\mathbb{X}$. If $\mathbb{X}$ is ergodic, we can use the general balance equation to show that π is the equilibrium distribution of the chain. However, given a distribution π, it

is often difficult to find a Markov kernel for which π satisfies the general balance equation. It is actually easier to find a Markov kernel that satisfies a *stronger* condition, known as detailed balance.

Definition A.10 (Detailed Balance). *We say that a transition density $p(x, z)$ satisfies* detailed balance *with respect to π if, for all $x, z \in E$, it holds that $\pi(x)p(x, z) = \pi(z)p(z, x)$.*

The Metropolis Hastings algorithm in Chapter 4 provides a general recipe to define a family of kernels that satisfy detailed balance with respect to a given distribution. The following result shows that detailed balance implies general balance.

Theorem A.11 (Detailed Balance Implies General Balance). *Let $p(x, z)$ be a Markov kernel that satisfies detailed balance with respect to a distribution π. Then, π is an invariant distribution for $p(x, z)$.*

Proof. Using the assumption of detailed balance and that $p(x, z)$ is a Markov kernel (and so it integrates to one over its second argument) gives

$$\int_E \pi(z)p(z, x)\, dz = \int_E \pi(x)p(x, z)\, dz = \pi(x) \int_E p(x, z)\, dz = \pi(x). \qquad \square$$

Remark A.12. *Detailed balance is* not *necessary for general balance. Detailed balance implies that the chain is time reversible: in equilibrium the chain behaves the same whether it is run forward or backward in time. However, there are many ergodic Markov chains that are not time reversible.*

A.3 ▪ Reducibility, Periodicity, and Transience in a Countable State Space

A Markov chain taking values on a countable state space E is ergodic if it is irreducible, aperiodic, and positive recurrent. We now review these concepts in the context of countable state space and then generalize them to general state space.

Definition A.13 (Irreducible Markov Chain). *A Markov chain is* irreducible *if all states intercommunicate. That is, if for all $i, j \in E$ there is $n \geq 0$ such that*

$$\mathbb{P}(X_n = i \,|\, X_0 = j) > 0. \qquad \square$$

Definition A.14 (Recurrent Markov Chain). *A Markov chain is* recurrent *if, for all $i \in E$,*

$$\mathbb{P}(\mathbb{X} \textit{ visits } i \textit{ eventually} \,|\, X_0 = i) = 1. \qquad \square$$

Definition A.15 (Transient Markov Chain). *A Markov chain is* transient *if, for all $i \in E$,*

$$\mathbb{P}(\mathbb{X} \textit{ visits } i \textit{ eventually} \,|\, X_0 = i) = 0. \qquad \square$$

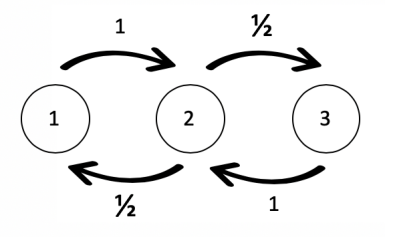
Definition A.16 (Positive Recurrent Markov Chain). *A Markov chain is positive recurrent if $\mathbb{E}[T_{ii}] < \infty$ for all $i \in E$, where T_{ii} is the time of the first return to state i.*

Definition A.17 (Aperiodic Markov Chain). *A Markov chain is* aperiodic *if all states have period 1. The period of state $i \in E$ is defined as*

$$d(i) := gcd\{n > 0 : p^n(i,i) > 0\},$$

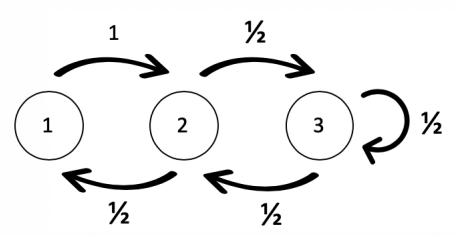
where gcd stands for greatest common divisor.

Example A.18 (Periodicity). For this chain



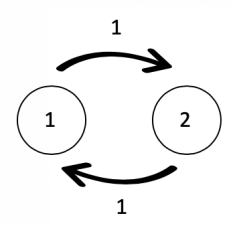
we can return to state 1 after 2, 4, 6, ... steps. Hence, state 1 has period 2. □

Example A.19 (Aperiodic State). For this chain



we can return to state 1 after 2, 4, 6, 7, ... steps. Hence, state 1 has period 1. □

Example A.20 (Periodic Chain). This is a periodic chain:



Note that

$$P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, P^2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, P^3 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \ldots$$

and so the transition probabilities do not converge as $n \to \infty$. Also P^n has some zero entries for all $n \geq 1$. □

Remark A.21. *Recurrence and aperiodicity are chain properties. If a Markov chain is irreducible, the whole space is a communicating chain. Therefore, if one state is recurrent, then the whole space is recurrent (for irreducible chains). Irreducibility ensures that the state space doesn't split into subsets such that the chain cannot move from one subset to the others. (Positive) recurrence ensures that the chain eventually visits every subset of the state space of positive mass (sufficiently often). Periodicity causes the state space to split into subsets (cyclically moving chains) which are visited by the chain in sequential order.*

A.4 ▪ Ergodic Theorem in Finite State Space

In this subsection, we study ergodicity of Markov chains under the assumption that the state space E is finite. In this setting, irreducibility and aperiodicity are enough to guarantee ergodicity. We will use the following lemma.

Lemma A.22. *Let P be the transition probability matrix of an irreducible, aperiodic, and finite state Markov chain. Then, there is an integer m such that, for all $n \geq m$, the matrix P^n has strictly positive entries.*

In order to establish ergodicity we will use a coupling argument which generalizes beyond the finite state space setting considered here. The coupling argument will be used repeatedly in these notes, and we will see that central to the argument is bounding the Markov kernel from below, which in the case of finite state space is achieved using the previous lemma. The coupling argument uses the total variation distance between probability measures.

Definition A.23 (Total Variation Distance). *The* total variation distance *between two probability measures ν_1, ν_2 on E is defined by*

$$d_{\text{TV}}(\nu_1, \nu_2) = \sup_{A \subseteq E} |\nu_1(A) - \nu_2(A)|. \tag{A.1}$$

While (A.1) is the standard definition of the total variation distance, we will use the following characterization.

Lemma A.24. *It holds that*

$$d_{\text{TV}}(\nu_1, \nu_2) = \frac{1}{2} \sup_{|h|_\infty \leq 1} \Big| \mathbb{E}_{X \sim \nu_1}[h(X)] - \mathbb{E}_{X \sim \nu_2}[h(X)] \Big|.$$

Theorem A.25 (Ergodicity of Markov Chains in Finite State Space). *Let $\mathbb{X} = \{X_n\}_{n=0}^{\infty}$ be a Markov chain that is irreducible and aperiodic with finite state space E. Then, there are $\varepsilon > 0$ and a unique invariant distribution π such that*

$$d_{\text{TV}}(\pi_n, \pi) \leq (1 - \varepsilon)^n, \tag{A.2}$$

where π_n denotes the law of X_n.

Proof. We will assume (without loss of generality by Lemma A.22) that $p_{ij} > \varepsilon$ for all $i, j \in E$. Consider the following map from $\mathcal{P}(E)$ (the set of probability row vectors on E) into itself:

$$\begin{aligned} \mathcal{P}(E) &\to \mathcal{P}(E) \\ \nu &\mapsto \nu P. \end{aligned}$$

Since this map is continuous and $\mathcal{P}(E)$ is compact and convex, Brouwer's fixed point theorem guarantees the existence of a fixed point. That is, there is $\pi \in \mathcal{P}(E)$ such that $\pi P = \pi$, showing the existence of an invariant distribution. In the rest of the proof we will prove that if π is an invariant distribution, then inequality (A.2) holds. This in particular will automatically imply the uniqueness of the invariant distribution.

We will use a *coupling* argument. Let $\{B_n\}_{n=0}^{\infty}$ be a sequence of independent Bernoulli(ε) random variables, independent of all other randomness, and define for given random variable W_0 and $n \geq 0$

$$W_{n+1} \sim \begin{cases} s(W_n, \cdot) & \text{if } B_n = 0, \\ r(W_n, \cdot) & \text{if } B_n = 1, \end{cases} \tag{A.3}$$

where

$$s(i,j) = \frac{p_{ij} - \varepsilon r(i,j)}{1-\varepsilon}$$

and r is the uniform transition kernel, so that $r(i,\cdot)$ is the discrete uniform distribution in E. Note that

$$s(i,E) = \frac{p(i,E) - \varepsilon r(i,E)}{1-\varepsilon} = \frac{1-\varepsilon}{1-\varepsilon} = 1$$

and

$$s(i,j) \geq \frac{\varepsilon - \varepsilon r(i,j)}{1-\varepsilon} \geq 0,$$

so s is a Markov kernel. Moreover,

$$\begin{aligned} \mathbb{P}(W_{n+1} = j \,|\, W_n = i) &= \varepsilon \mathbb{P}(W_{n+1} = j \,|\, B_n = 1, W_n = i) + (1-\varepsilon)\,\mathbb{P}(W_{n+1} = j \,|\, B_n = 0, W_n = i) \\ &= \varepsilon r(i,j) + p(i,j) - \varepsilon r(i,j) \\ &= p(i,j). \end{aligned}$$

Thus, $\{W_n\}$ has transition kernel P.

Let $h : E \to \mathbb{R}$ with $|h|_\infty \leq 1$, and let $\tau = \min(n \in \mathbb{N} : B_n = 1)$. Regardless of the distribution of W_0,

$$\begin{aligned} \mathbb{E}[h(W_n)] &= \mathbb{E}[h(W_n) \,|\, \tau \geq n]\,\mathbb{P}(\tau \geq n) + \sum_{l=0}^{n-1} \mathbb{E}[h(W_n) \,|\, \tau = l]\mathbb{P}(\tau = l) \\ &= \underbrace{\mathbb{E}[h(W_n) \,|\, \tau \geq n]}_{\leq 1}\underbrace{\mathbb{P}(\tau \geq n)}_{\leq (1-\epsilon)^n}| + \underbrace{\sum_{l=0}^{n-1} \mathbb{E}_{W_0 \sim \mathrm{Unif}(E)}[h(W_{n-l})]\mathbb{P}(\tau = l)}_{\text{independent of the initial distribution}}, \end{aligned}$$

where $\mathrm{Unif}(E)$ denotes the uniform distribution on E.

Now define two sequences of random variables as in equation (A.3): let $\{Y_n\}_{n=0}^{\infty}$ be initialized with π_0 (the initial distribution of $\mathbb{X}$), and let $\{Z_n\}_{n=0}^{\infty}$ be initialized with an invariant distribution π (which we have shown that exists) so that $Z_n \sim \pi$ for all n. Then,

$$d_{\mathrm{TV}}(\pi_n, \pi) = \frac{1}{2} \sup_{|h|_\infty \leq 1} \Big| \mathbb{E}_{Y_n \sim \pi_n}[h(Y_n)] - \mathbb{E}_{Z_n \sim \pi}[h(Z_n)] \Big| \leq (1-\varepsilon)^n. \qquad \square$$

A.5 ▪ Ergodic Theory in General State Space

This section shows how to generalize the definitions of irreducibility, aperiodicity and recurrence to general state spaces. We will also summarize some ergodicity results in general state space.

Definition A.26 (φ-Irreducible Markov Chain). *A Markov chain is called* φ-irreducible *if there exists a non-zero measure φ on E such that, for all $A \subseteq E$ with $\varphi(A) > 0$ and for all $x \in E$, there exist a positive integer $n = n(x)$ such that $P^n(x, A) > 0$.*

Definition A.27 (Aperiodic Markov Chain). *A chain is* aperiodic *if there do not exist $d \geq 2$ and disjoint subsets $E_1, \ldots, E_d \subseteq E$, with*

$$\begin{aligned} P(x, E_{i+1}) &= 1, \qquad \forall x \in E_i, \qquad i \in \{1, \ldots, d-1\}, \\ P(x, E_1) &= 1, \qquad \forall x \in E_d. \end{aligned}$$

We remark that if E is discrete, this definition agrees with the one previously given. The following result ensures that a Markov chain that is φ-irreducible and aperiodic has a limit distribution.

Theorem A.28 (Limit Distribution for Almost-All Initial States). *The distribution of an aperiodic, φ-irreducible Markov chain converges to a limit distribution π for almost-all initial states. More precisely, there is π such that, for π-almost all $x \in E$,*

$$\lim_{n\to\infty} d_{\text{TV}}(P^n(x, \cdot), \pi) = 0.$$

The above limit holds for almost all starting values $x \in E$. To make it hold for *all* starting values $x \in E$ and thus for all initial distributions π_0 we need an additional property: *Harris recurrence*.

Definition A.29 (Harris Recurrence). *A chain $\mathbb{X}$ is* Harris recurrent *if there is a probability ν such that for all $A \subseteq E$ with $\nu(A) > 0$ and all $x \in E$ we have*

$$\mathbb{P}(X_n \in A \text{ for some } n > 0 | X_0 = x) = 1. \qquad \square$$

Theorem A.30 (Limit Distribution for All Initial States). *The distribution of an aperiodic, Harris recurrent Markov chain converges to a limit distribution π, regardless of the initial state. More precisely, there is a limit distribution π such that, for all $x \in E$,*

$$\lim_{n\to\infty} d_{\text{TV}}(P^n(x, \cdot), \pi) = 0.$$

Corollary A.31 (Limit Distribution and Ergodicity). *Under the conditions of the previous theorem, for all $A \subseteq E$ and all initial conditions π_0, it holds that*

$$\lim_{n\to\infty} \mathbb{P}(X_n \in A) = \pi(A).$$

Therefore, the chain is ergodic.

Proof. The result follows by the dominated convergence theorem, noting that $\pi_n(A) = \mathbb{P}(X_n \in A) = \int_E \pi_0(x) P^n(x, A)\, dx$.

A.6 ▪ Sample Path Ergodicity

Why do we care about ergodicity of Markov chains in a Monte Carlo simulation course? Markov chain Monte Carlo (MCMC) algorithms draw samples from a Markov chain which has as invariant a prescribed target distribution $\pi = f$. Ultimately, we are interested in understanding the *sample path ergodicity* of MCMC chains, that is, we want to determine if ergodic averages

$$\frac{1}{N}\sum_{n=1}^{N} h(X^{(n)})$$

computed from MCMC samples accurately approximate expectations $\mathcal{I}_f[h]$ with respect to the target. These samples are *not* independent, and consequently the efficiency of the ergodic average estimator will depend on the correlations between them.

First, we have the following law of large numbers.

Theorem A.32 (Law of Large Numbers). *Let $h : E \to \mathbb{R}$ and let $\mathbb{X} = \{X_n\}_{n=0}^{\infty}$ be an ergodic Markov chain with stationary distribution π. Then, π-almost surely,*

$$\frac{1}{N}\sum_{n=1}^{N} h(X_n) \overset{N\to\infty}{\longrightarrow} \int_E h(x)\pi(x)\,dx \equiv \mathcal{I}_\pi[h].$$

We also have a central limit theorem, which requires geometric ergodicity of the chain.

Definition A.33 (Geometric and Uniform Ergodicity). *An ergodic Markov chain with invariant distribution π is* geometrically ergodic *if there exist a non-negative function M with $\mathbb{E}_\pi[M(X)] < \infty$ and $0 < r < 1$ such that*

$$d_{\text{TV}}(P^n(x,\cdot),\pi) \le M(x)r^n$$

for all x and all n. If the function M is bounded above, then the chain is called uniformly ergodic. □

Note, as an example, that we have proved that any irreducible and aperiodic Markov chain on a finite state space is uniformly ergodic. The following result shows that sample path ergodicity can be deduced from geometric ergodicity. In Chapter 4, we discuss the uniform and geometric ergodicity of several MCMC algorithms, thus guaranteeing in particular their sample path ergodicity.

Theorem A.34 (Central Limit Theorem). *With the same notation as in the previous theorem, suppose further that $\mathbb{X}$ is geometrically ergodic and that, for some $\epsilon > 0$,*

$$\mathbb{E}_{Y\sim\pi}[h(Y)^{2+\epsilon}] < \infty.$$

Then,

$$\sqrt{N}\Big(\frac{1}{N}\sum_{n=1}^{N} h(X_n) - \mathcal{I}_\pi[h]\Big) \xrightarrow{\mathcal{D}} \mathcal{N}(0,\tau^2),$$

where τ^2 is the integrated autocorrelation time, defined below in Lemma A.36.

The rest of this appendix is devoted to defining and understanding the integrated autocorrelated time, which may be used to assess the relative efficiency of different MCMC algorithms.

Unsurprisingly, the asymptotic variance τ^2 in the above central limit theorem arises as the limiting variance of ergodic averages. We define

$$\frac{\tau_N^2}{N} := \mathbb{V}\left[\frac{1}{N}\sum_{n=1}^{N} h(X_n)\right]$$

and note that τ_N^2 (and similarly σ^2 and τ below) depends on h but we omit said dependence from our notation. The value τ_N^2 quantifies the amount of correlation between the variables X_n's.

Definition A.35 (Autocovariance and Autocorrelation). *At stationarity, the* autocovariance *of lag k of the time series $\{h(X_n)\}_{n=0}^{\infty}$ is defined as*

$$\gamma_k = \mathrm{Cov}\big(h(X_0), h(X_k)\big), \qquad k \geq 1.$$

We also define $\sigma^2 = \mathrm{Cov}\big(h(X_0), h(X_0)\big) = \mathbb{V}_{X_0\sim\pi}[h(X_0)]$. The autocorrelation *of lag k is*

$$\rho_k = \frac{\gamma_k}{\sigma^2}, \qquad k \geq 0. \qquad \square$$

Lemma A.36. *At stationarity,*

$$\tau_N^2 = \sigma^2\left[1 + 2\sum_{k=1}^{N-1}\frac{N-k}{N}\rho_k\right].$$

As $N \to \infty$,

$$\tau_N^2 \to \tau^2 := \sigma^2\left[1 + 2\sum_{k=1}^{\infty}\rho_k\right],$$

where τ^2 is called the integrated autocorrelation time.

Proof. A calculation shows that

$$\begin{aligned}
\frac{\tau_N^2}{N} &= \mathbb{V}\left[\frac{1}{N}\sum_{n=1}^{N} h(X_n)\right] \\
&= \frac{1}{N^2}\left[\sum_{n=1}^{N}\mathbb{V}[h(X_n)] + 2\sum_{n=1}^{N-1}\sum_{k>n}\mathrm{Cov}\left(h(X_n), h(X_k)\right)\right] \\
&= \frac{1}{N^2}\left[N\sigma^2 + 2\sum_{n=1}^{N-1}\sum_{k=1}^{N-n}\mathrm{Cov}\left(h(X_n), h(X_{n+k})\right)\right] \\
&= \frac{\sigma^2}{N}\left[1 + \frac{2}{N\sigma^2}\sum_{n=1}^{N-1}\sum_{k=1}^{N-n}\gamma_k\right] \\
&= \frac{\sigma^2}{N}\left[1 + \frac{2}{N}\sum_{n=1}^{N-1}\sum_{k=1}^{N-n}\rho_k\right] \\
&= \frac{\sigma^2}{N}\left[1 + \frac{2}{N}\sum_{k=1}^{N-1}\sum_{n=1}^{N-k}\rho_k\right] \\
&= \frac{\sigma^2}{N}\left[1 + 2\sum_{k=1}^{N-1}\frac{N-k}{N}\rho_k\right].
\end{aligned}$$

$\square$

Note that if $X_n \overset{\text{i.i.d.}}{\sim} \pi$, then

$$\mathbb{V}\left[\frac{1}{N}\sum_{n=1}^{N} h(X_n)\right] = \frac{\sigma^2}{N}.$$

Therefore, computing an ergodic average with positively autocorrelated samples leads to higher variance than computing it with an i.i.d. sample. An intuitive explanation is that positively correlated random variables have redundant information so are less informative than i.i.d. random variables. On the other hand, if the correlations are negative, ergodic averages may be more accurate than a direct Monte Carlo estimator with i.i.d. samples.

A.7 ▪ Discussion and Bibliography

The material in this appendix can be found in any standard reference on Markov chains. For gentle introductions, we refer to [174, 35] and [135, Chapter 1]. For a discussion of more advanced topics, we refer to [164, 57, 175].

Bibliography

[1] E. H. L. AARTS AND J. KORST, *Simulated Annealing and Boltzmann Machines: a Stochastic Approach to Combinatorial Optimization and Neural Computing*, John Wiley & Sons, Inc., 1989.

[2] E. H. L. AARTS AND P. J. M. VAN LAARHOVEN, *Statistical cooling: A general approach to combinatorial optimization problems*, Philips Journal of Research, 40 (1985), pp. 193–226.

[3] S. AGAPIOU, O. PAPASPILIOPOULOS, D. SANZ-ALONSO, AND A. M. STUART, *Importance sampling: Intrinsic dimension and computational cost*, Statistical Science, 32 (2017), pp. 405–431.

[4] S. AGRAWAL, H. KIM, D. SANZ-ALONSO, AND A. STRANG, *A variational inference approach to inverse problems with gamma hyperpriors*, SIAM/ASA Journal on Uncertainty Quantification, 10 (2022), pp. 1533–1559.

[5] O. AL-GHATTAS, J. CHEN, D. SANZ-ALONSO, AND N. WANIOREK, *Covariance operator estimation: sparsity, lengthscale, and ensemble Kalman filters*, arXiv preprint arXiv:2310.16933, (2023).

[6] O. AL-GHATTAS AND D. SANZ-ALONSO, *Non-asymptotic analysis of ensemble Kalman updates: effective dimension and localization*, Information and Inference: A Journal of the IMA, 13 (2024), p. iaad043.

[7] L. AMBROSIO, N. GIGLI, AND G. SAVARÉ, *Gradient Flows: in Metric Spaces and in the Space of Probability Measures*, Springer Science & Business Media, 2008.

[8] Y. AMIT AND U. GRENANDER, *Comparing sweep strategies for stochastic relaxation*, Journal of Multivariate Analysis, 37 (1991), pp. 197–222.

[9] C. ANDERSON, *Monte Carlo methods and importance sampling*, Lecture Notes for Statistical Genetics, (2014).

[10] C. ANDRIEU, A. DOUCET, AND R. HOLENSTEIN, *Particle Markov chain Monte Carlo methods*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72 (2010), pp. 269–342.

[11] C. ANDRIEU, A. LEE, AND S. LIVINGSTONE, *A general perspective on the Metropolis-Hastings kernel*, arXiv preprint arXiv:2012.14881, (2020).

[12] C. ANDRIEU AND G. ROBERTS, *The pseudo-marginal approach for efficient Monte Carlo computations*, Annals of Statistics, 37 (2009), pp. 697–725.

[13] C. ANDRIEU AND J. THOMS, *A tutorial on adaptive MCMC*, Statistics and Computing, 18 (2008), pp. 343–373.

[14] S. ASMUSSEN AND P. W. GLYNN, *Stochastic Simulation: Algorithms and Analysis*, vol. 57, Springer, 2007.

[15] A. BARBU AND S.-C. ZHU, *Monte Carlo Methods*, vol. 35, Springer, 2020.

[16] R. BARDENET, A. DOUCET, AND C. HOLMES, *On Markov chain Monte Carlo methods for tall data*, Journal of Machine Learning Research, 18 (2017), pp. 1–43.

[17] M. A. BEAUMONT, *Estimation of population growth or decline in genetically monitored populations*, Genetics, 164 (2003), pp. 1139–1160.

[18] M. A. BEAUMONT, W. ZHANG, AND D. J. BALDING, *Approximate Bayesian computation in population genetics*, Genetics, 162 (2002), pp. 2025–2035.

[19] C. BECQUET AND M. PRZEWORSKI, *A new approach to estimate parameters of speciation models with application to apes*, Genome research, 17 (2007), pp. 1505–1519.

[20] J. BESAG, *Comments on "Representations of knowledge in complex systems" by U. Grenander and M.I. Miller*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 56 (1994), pp. 591–592.

[21] A. BESKOS, N. PILLAI, G. ROBERTS, J. M. SANZ-SERNA, AND A. M. STUART, *Optimal tuning of the hybrid Monte Carlo algorithm*, Bernoulli, 19 (2013), pp. 1501–1534.

[22] A. BESKOS, F. J. PINSKI, J. M. SANZ-SERNA, AND A. M. STUART, *Hybrid Monte Carlo on Hilbert spaces*, Stochastic Processes and their Applications, 121 (2011), pp. 2201–2230.

[23] M. BETANCOURT, *A conceptual introduction to Hamiltonian Monte Carlo*, arXiv preprint arXiv:1701.02434, (2017).

[24] P. BICKEL, B. LI, AND T. BENGTSSON, *Pushing the limits of contemporary statistics: Contributions in honor of Jayanta K. Ghosh: Sharp failure rates for the bootstrap particle filter in high dimensions*, Institute of Mathematical Statistics, (1994), pp. 318–329.

[25] G. BIERKENS, P. FEARNHEAD, AND G. ROBERTS, *The Zig-Zag process and super-efficient sampling for Bayesian analysis of big data*, Annals of Statistics, 47 (2019).

[26] J. BIERKENS, *Non-reversible Metropolis-Hastings*, Statistics and Computing, 26 (2016), pp. 1213–1228.

[27] J. A. BILMES ET AL., *A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models*, International Computer Science Institute, 4 (1998), p. 126.

[28] C. M. BISHOP, *Pattern Recognition and Machine Learning*, Springer New York, 2006.

[29] D. M. BLEI, A. KUCUKELBIR, AND J. D. MCAULIFFE, *Variational inference: A review for statisticians*, Journal of the American Statistical Association, 112 (2017), p. 859–877.

[30] N. BOU-RABEE, A. EBERLE, AND R. ZIMMER, *Coupling and convergence for Hamiltonian Monte Carlo*, The Annals of Applied Probability, 30 (2018), pp. 1209–1250.

[31] N. BOU-RABEE AND J. M. SANZ-SERNA, *Geometric integrators and the Hamiltonian Monte Carlo method*, Acta Numerica, 27 (2018), pp. 113–206.

[32] A. BOUCHARD-CÔTÉ, S. J. VOLLMER, AND A. DOUCET, *The bouncy particle sampler: A nonreversible rejection-free Markov chain Monte Carlo method*, Journal of the American Statistical Association, 113 (2018), pp. 855–867.

[33] G. E. BOX AND M. E. MULLER, *A note on the generation of random normal deviates*, The Annals of Mathematical Statistics, 29 (1958), pp. 610–611.

[34] S. BOYD, S. P. BOYD, AND L. VANDENBERGHE, *Convex Optimization*, Cambridge University Press, 2004.

[35] P. BRÉMAUD, *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*, vol. 31, Springer Science & Business Media, 2013.

[36] S. BROOKS, *Markov chain Monte Carlo method and its application*, Journal of the Royal Statistical Society: Series D (the Statistician), 47 (1998), pp. 69–100.

[37] S. BROOKS, A. GELMAN, G. JONES, AND X. MENG, *Handbook of Markov chain Monte Carlo*, CRC Press, (2011).

[38] S. P. BROOKS AND G. O. ROBERTS, *Convergence assessment techniques for Markov chain Monte Carlo*, Statistics and Computing, 8 (1998), pp. 319–335.

[39] M. F. BUGALLO, V. ELVIRA, L. MARTINO, D. LUENGO, J. MIGUEZ, AND P. M. DJURIC, *Adaptive importance sampling: The past, the present, and the future*, IEEE Signal Processing Magazine, 34 (2017), pp. 60–79.

[40] R. E. CAFLISCH, *Monte Carlo and quasi-Monte Carlo methods*, Acta Numerica, 7 (1998), pp. 1–49.

[41] B. CALDERHEAD, *A general construction for parallelizing Metropolis-Hastings algorithms*, Proceedings of the National Academy of Sciences, 111 (2014), pp. 17408–17413.

[42] M. P. CALVO, D. SANZ-ALONSO, AND J. M. SANZ-SERNA, *HMC: reducing the number of rejections by not using leapfrog and some results on the acceptance rate*, Journal of Computational Physics, 437 (2021), p. 110333.

[43] R. CEPPELLINI, M. SINISCALCO, AND S. CA, *The estimation of gene frequencies in a random-mating population*, Annals of Human Genetics, 20 (1955), pp. 97–115.

[44] N. K. CHADA, Y. CHEN, AND D. SANZ-ALONSO, *Iterative ensemble Kalman methods: A unified perspective with some new variants*, Foundations of Data Science, 3 (2021), pp. 331–369.

[45] S. CHATTERJEE AND P. DIACONIS, *The sample size required in importance sampling*, The Annals of Applied Probability, 28 (2018), pp. 1099–1135.

[46] Y. CHEN, *Another look at rejection sampling through importance sampling*, Statistics & Probability Letters, 72 (2005), pp. 277–283.

[47] Y. CHEN, D. Z. HUANG, J. HUANG, S. REICH, AND A. M. STUART, *Gradient flows for sampling: mean-field models, Gaussian approximations and affine invariance*, arXiv preprint arXiv:2302.11024, (2023).

[48] C. CHIMISOV, K. LATUSZYNSKI, AND G. O. ROBERTS, *Adapting the Gibbs sampler*, arXiv preprint arXiv:1801.09299, (2018).

[49] N. CHOPIN AND O. PAPASPILIOPOULOS, *An Introduction to Sequential Monte Carlo*, Springer, 2020.

[50] J. A. CHRISTEN AND C. FOX, *Markov chain Monte Carlo using an approximation*, Journal of Computational and Graphical Statistics, 14 (2005), pp. 795–810.

[51] G. L. L. COMTE DE BUFFON, *Histoire naturelle: générale et particulière*, vol. 16, De l'Imprimerie Royale, 1770.

[52] S. L. COTTER, G. O. ROBERTS, A. M. STUART, AND D. WHITE, *MCMC methods for functions: modifying old algorithms to make them faster*, Statistical Science, 28 (2013), pp. 424–446.

[53] A. S. DALALYAN AND A. KARAGULYAN, *User-friendly guarantees for the Langevin Monte Carlo with inaccurate gradient*, Stochastic Processes and their Applications, 129 (2019), pp. 5278–5311.

[54] P. DEL MORAL, *Feynman-Kac Formulae*, Springer, 2004.

[55] A. P. DEMPSTER, N. M. LAIRD, AND D. B. RUBIN, *Maximum likelihood from incomplete data via the EM algorithm*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 39 (1977), pp. 1–22.

[56] L. DEVROYE, *Non-Uniform Random Variate Generation*, Springer Science & Business Media, 2013.

[57] R. DOUC, E. MOULINES, P. PRIOURET, AND P. SOULIER, *Markov Chains*, vol. 1, Springer, 2018.

[58] A. DOUCET, N. DE FREITAS, AND N. GORDON, *An Introduction to Sequential Monte Carlo Methods*, in Sequential Monte Carlo Methods in Practice, Springer, 2001, pp. 3–14.

[59] A. DOUCET, S. GODSILL, AND C. ANDRIEU, *On sequential Monte Carlo sampling methods for Bayesian filtering*, Statistics and Computing, 10 (2000).

[60] S. DUANE, A. D. KENNEDY, B. J. PENDLETON, AND D. ROWETH, *Hybrid Monte Carlo*, Physics letters B, 195 (1987), pp. 216–222.

[61] D. B. DUNSON AND J. E. JOHNDROW, *The Hastings algorithm at fifty*, Biometrika, 107 (2020), pp. 1–23.

[62] P. DUPUIS, K. SPILIOPOULOS, AND H. WANG, *Importance sampling for multiscale diffusions*, Multiscale Modeling and Simulation, 10 (2012), pp. 1–27.

[63] P. DUPUIS AND G.-J. WU, *Analysis and optimization of certain parallel Monte Carlo methods in the low temperature limit*, Multiscale Modeling and Simulation, 20 (2022), pp. 220–249.

[64] A. DURMUS, S. MAJEWSKI, AND B. MIASOJEDOW, *Analysis of Langevin Monte Carlo via convex optimization*, The Journal of Machine Learning Research, 20 (2019), pp. 2666–2711.

[65] A. DURMUS, E. MOULINES, AND E. SAKSMAN, *On the convergence of Hamiltonian Monte Carlo*, arXiv preprint arXiv:1705.00166, (2017).

[66] D. J. EARL AND M. W. DEEM, *Parallel tempering: Theory, applications, and new perspectives*, Physical Chemistry Chemical Physics, 7 (2005), pp. 3910–3916.

[67] R. ECKHARDT, *San Ulam, John von Neumann*, Los Alamos Science, (1987), p. 131.

[68] Y. EFENDIEV, A. DATTA-GUPTA, V. GINTING, X. MA, AND B. MALLICK, *An efficient two-stage Markov chain Monte Carlo method for dynamic data integration*, Water Resources Research, 41 (2005).

[69] Y. EFENDIEV, T. HOU, AND W. LUO, *Preconditioning Markov chain Monte Carlo simulations using coarse-scale models*, SIAM Journal on Scientific Computing, 28 (2006), pp. 776–803.

[70] B. EFRON, *Bootstrap method: another look at the Jackknife method*, Annals of Statistics, 7 (1979), pp. 1–26.

[71] V. ELVIRA, L. MARTINO, AND C. P. ROBERT, *Rethinking the effective sample size*, International Statistical Review, 90 (2022), pp. 525–550.

[72] L. C. EVANS, *An Introduction to Stochastic Differential Equations*, vol. 82, American Mathematical Society, 2012.

[73] A. FARCHI AND M. BOCQUET, *Comparison of local particle filters and new implementations*, Nonlinear Processes in Geophysics, 25 (2018), pp. 765–807.

[74] P. FEARNHEAD, J. BIERKENS, M. POLLOCK, AND G. O. ROBERTS, *Piecewise deterministic markov processes for continuous-time monte carlo*, Statistical Science, 33 (2018), pp. 386–412.

[75] M. FOLL, M. A. BEAUMONT, AND O. GAGGIOTTI, *An approximate Bayesian computation approach to overcome biases that arise when using amplified fragment length polymorphism markers to study population structure*, Genetics, 179 (2008), pp. 927–939.

[76] M. FREI AND H. R. KÜNSCH, *Bridging the ensemble kalman and particle filters*, Biometrika, 100 (2013), pp. 781–800.

[77] A. GARBUNO-INIGO, F. HOFFMANN, W. LI, AND A. M. STUART, *Interacting Langevin diffusions: Gradient structure and ensemble Kalman sampler*, SIAM Journal on Applied Dynamical Systems, 19 (2020), pp. 412–441.

[78] N. GARCIA TRILLOS, B. HOSSEINI, AND D. SANZ-ALONSO, *From optimization to sampling through gradient flows*, Notices of the American Mathematical Society, 70 (2023).

[79] N. GARCIA TRILLOS AND D. SANZ-ALONSO, *The Bayesian update: variational formulations and gradient flows*, Bayesian Analysis, 15 (2020), pp. 29–56.

[80] A. E. GELFAND AND A. F. M. SMITH, *Sampling-based approaches to calculating marginal densities*, Journal of the American Statistical Association, 85 (1990), pp. 398–409.

[81] A. GELMAN AND D. B. RUBIN, *Inference from iterative simulation using multiple sequences*, Statistical Science, 7 (1992), pp. 457–472.

[82] S. GEMAN AND D. GEMAN, *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*, IEEE Transactions on Pattern Recognition, (1984), pp. 721–741.

[83] J. E. GENTLE, *Random Number Generation and Monte Carlo Methods*, vol. 381, Springer, 2003.

[84] J. GEWEKE, *Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments*, vol. 196, Federal Reserve Bank of Minneapolis, Research Department Minneapolis, MN, 1991.

[85] C. J. GEYER, *Computing science and statistics: Proceedings of the 23rd Symposium on the Interface*, American Statistical Association, New York, 156 (1991).

[86] A. GIBBS AND F. SU, *On choosing and bounding probability metrics*, International Statistical Review, 70 (2002), pp. 419–435.

[87] B. GIDAS, *Nonstationary Markov chains and convergence of the annealing algorithm*, Journal of Statistical Physics, 39 (1985), pp. 73–131.

[88] W. R. GILKS, S. RICHARDSON, AND D. SPIEGELHALTER, *Markov Chain Monte Carlo in Practice*, Chapman and Hall/CRC, 1995.

[89] M. GIROLAMI AND B. CALDERHEAD, *Riemann manifold Langevin and Hamiltonian Monte Carlo methods*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 73 (2011), pp. 123–214.

[90] P. GLASSERMAN, *Monte Carlo Methods in Financial Engineering*, vol. 53, Springer, 2004.

[91] N. E. GLATT-HOLTZ, A. J. HOLBROOK, J. A. KROMETIS, AND C. F. MONDAINI, *Parallel MCMC algorithms: Theoretical foundations, algorithm design, case studies*, arXiv preprint arXiv:2209.04750, (2022).

[92] N. E. GLATT-HOLTZ AND C. F. MONDAINI, *Mixing rates for Hamiltonian Monte Carlo algorithms in finite and infinite dimensions*, Stochastics and Partial Differential Equations: Analysis and Computations, (2021), pp. 1–74.

[93] A. GOLIGHTLY AND D. J. WILKINSON, *Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo*, Interface Focus, 1 (2011), pp. 807–820.

[94] N. J. Gordon, D. J. Salmond, and A. F. Smith, *Novel approach to nonlinear/non-Gaussian Bayesian state estimation*, in IEE Proceedings F (Radar and Signal Processing), vol. 140, IET, 1993, pp. 107–113.

[95] C. Graham and D. Talay, *Stochastic Simulation and Monte Carlo Methods: Mathematical Foundations of Stochastic Simulation*, vol. 68, Springer Science & Business Media, 2013.

[96] P. J. Green, *Reversible jump Markov chain Monte Carlo computation and Bayesian model determination*, Biometrika, 82 (1995), pp. 711–732.

[97] P. J. Green and A. Mira, *Delayed rejection in reversible jump Metropolis–Hastings*, Biometrika, 88 (2001), pp. 1035–1053.

[98] U. Grenander and M. I. Miller, *Representations of knowledge in complex systems*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 56 (1994), pp. 549–581.

[99] J. Gubernatis, N. Kawashima, and P. Werner, *Quantum Monte Carlo Methods*, Cambridge University Press, 2016.

[100] H. Haario, E. Saksman, and J. Tamminen, *An adaptive Metropolis algorithm*, Bernoulli, (2001), pp. 223–242.

[101] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration: Structure-preserving Algorithms for Ordinary Differential Equations*, vol. 31, Springer Science & Business Media, 2006.

[102] M. Hairer, A. M. Stuart, and S. J. Vollmer, *Spectral gaps for a Metropolis–Hastings algorithm in infinite dimensions*, The Annals of Applied Probability, 24 (2014), pp. 2455–2490.

[103] B. Hajek, *Cooling schedules for optimal annealing*, Mathematics of Operations Research, 13 (1988), pp. 311–329.

[104] G. Hamilton, M. Stoneking, and L. Excoffier, *Molecular analysis reveals tighter social regulation of immigration in patrilocal populations than in matrilocal populations*, Proceedings of the National Academy of Sciences, 102 (2005), pp. 7476–7480.

[105] J. Hammersley and D. Handscomb, *Percolation processes*, Monte Carlo Methods, (1964), pp. 134–141.

[106] H. O. Hartley, *Maximum likelihood estimation from incomplete data*, Biometrics, 14 (1958), pp. 174–194.

[107] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2, Springer, 2009.

[108] W. K. Hastings, *Monte Carlo sampling methods using Markov chains and their applications*, Biometrika, 57 (1970), pp. 97–109.

[109] D. Henderson, S. H. Jacobson, and A. W. Johnson, *The theory and practice of simulated annealing*, Handbook of Metaheuristics, (2003), pp. 287–319.

[110] J. Hernandez-Lobato, Y. Li, M. Rowland, T. Bui, D. Hernández-Lobato, and R. Turner, *Black-box alpha divergence minimization*, in International Conference on Machine Learning, PMLR, 2016, pp. 1511–1520.

[111] D. Higham and P. Kloeden, *An Introduction to the Numerical Simulation of Stochastic Differential Equations*, SIAM, 2021.

[112] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, *Stochastic variational inference*, Journal of Machine Learning Research, (2013).

[113] M. D. HOFFMAN AND A. GELMAN, *The No-U-Turn Sampler: adaptively setting path lengths in Hamiltonian Monte Carlo*, Journal of Machine Learning Research, 15 (2014), pp. 1351–1381.

[114] K. HUKUSHIMA AND K. NEMOTO, *Exchange Monte Carlo method and application to spin glass simulations*, Journal of the Physical Society of Japan, 65 (1996), pp. 1604–1608.

[115] D. R. HUNTER AND K. LANGE, *A tutorial on MM algorithms*, The American Statistician, 58 (2004), pp. 30–37.

[116] P. JÄCKEL, *Monte Carlo Methods in Finance*, vol. 5, John Wiley & Sons, 2002.

[117] M. JAMSHIDIAN AND R. I. JENNRICH, *Acceleration of the EM algorithm by using quasi-Newton methods*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 59 (1997), pp. 569–587.

[118] A. JOHANSEN AND A. DOUCET, *A note on auxiliary particle filters*, Statistics and Probability Letters, 78 (2008), pp. 1498–1504.

[119] M. I. JORDAN, Z. GHAHRAMANI, T. S. JAAKKOLA, AND L. K. SAUL, *An introduction to variational methods for graphical models*, Machine Learning, 37 (1999), pp. 183–233.

[120] H. KAHN, *Use of different Monte Carlo sampling techniques*, Rand Corporation, 1955.

[121] H. KAHN AND A. W. MARSHALL, *Methods of reducing sample size in Monte Carlo computations*, Journal of the Operations Research Society of America, 1 (1953), pp. 263–278.

[122] R. KAWAI, *Adaptive importance sampling Monte Carlo simulation for general multivariate probability laws*, Journal of Computational and Applied Mathematics, 319 (2017), pp. 440–459.

[123] D. P. KINGMA AND M. WELLING, *Auto-Encoding Variational Bayes*, in 2nd International Conference on Learning Representations, 2014.

[124] S. KIRKPATRICK, C. D. GELATT, AND M. P. VECCHI, *Optimization by simulated annealing*, Science, 220 (1983), pp. 671–680.

[125] P. E. KLOEDEN AND E. PLATEN, *Numerical Solution of Stochastic Differential Equations*, vol. 23, Springer Science & Business Media, 2013.

[126] H. KNOTHE, *Contributions to the theory of convex bodies*, Michigan Mathematical Journal, 4 (1957), pp. 39–52.

[127] D. E. KNUTH, *The Art of Computer Programming*, vol. 2, Pearson Education, 1997.

[128] A. KONG, *A note on importance sampling using standardized weights*, University of Chicago, Dept. of Statistics, Tech. Rep, 348 (1992).

[129] A. KONG, J. S. LIU, AND W. H. WONG, *Sequential imputations and Bayesian missing data problems*, Journal of the American Statistical Association, 89 (1994), pp. 278–288.

[130] D. P. KROESE, T. TAIMRE, AND Z. I. BOTEV, *Handbook of Monte Carlo Methods*, John Wiley & Sons, 2013.

[131] A. KUCUKELBIR, D. TRAN, R. RANGANATH, A. GELMAN, AND D. M. BLEI, *Automatic differentiation variational inference*, The Journal of Machine Learning Research, 18 (2017), pp. 430–474.

[132] S. KULLBACK AND R. A. LEIBLER, *On information and sufficiency*, The Annals of Mathematical Statistics, 22 (1951), pp. 79–86.

[133] M. LAMBERT, S. CHEWI, F. BACH, S. BONNABEL, AND P. RIGOLLET, *Variational inference via Wasserstein gradient flows*, Advances in Neural Information Processing Systems, 35 (2022), pp. 14434–14447.

[134] P.-S. LAPLACE, *Theorie Analytique des Probabilités*, Paris: V. Courcier, 1812.

[135] G. F. LAWLER, *Introduction to Stochastic Processes*, Chapman and Hall/CRC, 2018.

[136] P. L'ECUYER, *Random Number Generation*, Springer, 2012.

[137] B. LEIMKUHLER, C. MATTHEWS, AND J. WEARE, *Ensemble preconditioning for Markov chain Monte Carlo simulation*, Statistics and Computing, 28 (2018), pp. 277–290.

[138] T. LELIEVRE, F. NIER, AND G. A. PAVLIOTIS, *Optimal non-reversible linear drift for the convergence to equilibrium of a diffusion*, Journal of Statistical Physics, 152 (2013), pp. 237–274.

[139] R. LI, H. ZHA, AND M. TAO, *Sqrt (d) dimension dependence of Langevin Monte Carlo*, in International Conference on Learning Representations, 2021.

[140] W. LI, Z. TAN, AND R. CHEN, *Two-stage importance sampling with mixture proposals*, Journal of the American Statistical Association, 108 (2013), pp. 1350–1365.

[141] Y. LI AND R. E. TURNER, *Rényi divergence variational inference*, Advances in Neural Information Processing Systems, 29 (2016).

[142] F. LIANG, C. LIU, AND R. J. CARROLL, *Stochastic approximation in Monte Carlo computation*, Journal of the American Statistical Association, 102 (2007), pp. 305–320.

[143] C. LIU AND D. B. RUBIN, *The ECME algorithm: a simple extension of EM and ECM with faster monotone convergence*, Biometrika, 81 (1994), pp. 633–648.

[144] J. S. LIU, *Metropolized independent sampling with comparisons to rejection sampling and importance sampling*, Statistics and Computing, 6 (1996), pp. 113–119.

[145] J. S. LIU, *Monte Carlo Strategies in Scientific Computing*, Springer Science & Business Media, 2008.

[146] J. S. LIU AND R. CHEN, *Sequential Monte Carlo methods for dynamic systems*, Journal of the American Statistical Association, 93 (1998), pp. 1032–1044.

[147] J. S. LIU, F. LIANG, AND W. H. WONG, *The multiple-try method and local optimization in Metropolis sampling*, Journal of the American Statistical Association, 95 (2000), pp. 121–134.

[148] S. LIVINGSTONE, M. BETANCOURT, S. BYRNE, AND M. GIROLAMI, *On the geometric ergodicity of Hamiltonian Monte Carlo*, Bernoulli, 25 (2019), pp. 3109–3138.

[149] Z. LOU, *A massively scalable parallel simulated annealing algorithm*, PhD thesis, The University of Chicago, 2016.

[150] Y.-A. MA, T. CHEN, AND E. FOX, *A complete recipe for stochastic gradient mcmc*, Advances in Neural Information Processing Systems, 28 (2015).

[151] Y.-A. MA, N. J. FOTI, AND E. B. FOX, *Stochastic gradient MCMC methods for hidden Markov models*, in International Conference on Machine Learning, PMLR, 2017, pp. 2265–2274.

[152] D. J. MACKAY, *Information Theory, Inference and Learning Algorithms*, Cambridge University Press, 2003.

[153] X. MAO, *Stochastic Differential Equations and Applications*, Elsevier, 2007.

[154] E. MARINARI AND G. PARISI, *Simulated tempering: a new Monte Carlo scheme*, Europhysics Letters, 19 (1992), p. 451.

[155] P. MARJORAM, J. MOLITOR, V. PLAGNOL, AND S. TAVARE., *Markov chain Monte Carlo without likelihoods*, Proceedings of the National Academy of Sciences of the United States of America, 100 (2003), pp. 15324–15328.

[156] L. MARTINO, V. ELVIRA, AND F. LOUZADA, *Effective sample size for importance sampling based on discrepancy measures*, Signal Processing, 131 (2017), pp. 386–401.

[157] Y. MARZOUK, T. MOSELHY, M. PARNO, AND A. SPANTINI, *Sampling via measure transport: an introduction*, Handbook of Uncertainty Quantification, (2016), pp. 1–41.

[158] G. MCLACHLAN AND T. KRISHNAN, *The EM Algorithm and Extensions*, vol. 382, John Wiley & Sons, 2007.

[159] X.-L. MENG AND D. B. RUBIN, *Maximum likelihood estimation via the ECM algorithm: A general framework*, Biometrika, 80 (1993), pp. 267–278.

[160] X.-L. MENG AND D. VAN DYK, *The EM algorithm—an old folk-song sung to a fast new tune*, Journal of the Royal Statistical Society Series B: Statistical Methodology, 59 (1997), pp. 511–567.

[161] K. L. MENGERSEN AND R. L. TWEEDIE, *Rates of convergence of the Hastings and Metropolis algorithms*, The Annals of Statistics, 24 (1996), pp. 101–121.

[162] N. METROPOLIS, A. W. ROSENBLUTH, M. N. ROSENBLUTH, A. H. TELLER, AND E. TELLER, *Equation of state calculations by fast computing machines*, The Journal of Chemical Physics, 21 (1953), pp. 1087–1092.

[163] N. METROPOLIS AND S. ULAM, *The Monte Carlo Method*, Journal of the American Statistical Association, 44 (1949), pp. 335–341.

[164] S. P. MEYN AND R. L. TWEEDIE, *Markov Chains and Stochastic Stability*, Springer Science & Business Media, 2012.

[165] M. MORZFELD, D. HODYSS, AND C. SNYDER, *What the collapse of the ensemble Kalman filter tells us about particle filters*, Tellus A: Dynamic Meteorology and Oceanography, 69 (2017), p. 1283809.

[166] I. MURRAY, R. P. ADAMS, AND D. J. MACKAY, *Elliptical slice sampling*, in Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS), Journal of Machine Learning Research: Workshop and Conference Proceedings, 2010, pp. 541–548.

[167] P. MYKLAND, L. TIERNEY, AND B. YU, *Regeneration in Markov chain samplers*, Journal of the American Statistical Association, 90 (1995), pp. 233–241.

[168] R. M. NEAL, *Sampling from multimodal distributions using tempered transitions*, Statistics and computing, 6 (1996), pp. 353–366.

[169] ———, *Slice sampling*, The Annals of Statistics, 31 (2003), pp. 705–767.

[170] R. M. NEAL, *MCMC using Hamiltonian dynamics*, Handbook of Markov chain Monte Carlo, 2 (2011), p. 2.

[171] ———, *Bayesian Learning for Neural Networks*, vol. 118, Springer Science & Business Media, 2012.

[172] R. M. NEAL AND G. E. HINTON, *A view of the EM algorithm that justifies incremental, sparse, and other variants*, in Learning in Graphical Models, Springer, 1998, pp. 355–368.

[173] M. E. NEWMAN AND G. T. BARKEMA, *Monte Carlo Methods in Statistical Physics*, Clarendon Press, 1999.

[174] J. R. NORRIS, *Markov Chains*, Cambridge University Press, 1998.

[175] E. NUMMELIN, *General Irreducible Markov Chains and Non-negative Operators*, vol. 83, Cambridge University Press, 2004.

[176] B. OKSENDAL, *Stochastic Differential Equations: an Introduction with Applications*, Springer Science & Business Media, 2013.

[177] A. OWEN AND Y. ZHOU, *Safe and effective importance sampling*, Journal of the American Statistical Association, 95 (2000), pp. 135–143.

[178] A. B. OWEN, *Monte Carlo Theory, Methods and Examples*, 2013.

[179] G. PAPAMAKARIOS AND I. MURRAY, *Fast ε-free inference of simulation models with Bayesian conditional density estimation*, Advances in Neural Information Processing Systems, 29 (2016).

[180] G. PAPAMAKARIOS, E. NALISNICK, D. J. REZENDE, S. MOHAMED, AND B. LAKSHMINARAYANAN, *Normalizing flows for probabilistic modeling and inference*, The Journal of Machine Learning Research, 22 (2021), pp. 2617–2680.

[181] G. PAPAMAKARIOS, T. PAVLAKOU, AND I. MURRAY, *Masked autoregressive flow for density estimation*, Advances in Neural Information Processing Systems, 30 (2017).

[182] O. PAPASPILIOPOULOS AND G. O. ROBERTS, *Stability of the Gibbs sampler for Bayesian hierarchical models*, The Annals of Statistics, 36 (2008), pp. 95–117.

[183] S. PATTERSON AND Y. W. TEH, *Stochastic gradient Riemannian Langevin dynamics on the probability simplex*, Advances in Neural Information Processing Systems, 26 (2013).

[184] G. A. PAVLIOTIS, *Stochastic Processes and Applications: Diffusion Processes, the Fokker-Planck and Langevin Equations*, vol. 60, Springer, 2014.

[185] M. PITT AND N. SHEPHARD, *Filtering via simulation: Auxiliary particle filters*, Journal of the American Statistical Association, 94 (1999), pp. 590–599.

[186] H. POINCARÉ, *Les Méthodes Nouvelles de la Mécanique Céleste*, vol. 3, Gauthier-Villars et fils, 1899.

[187] J. K. PRITCHARD, M. T. SEIELSTAD, A. PEREZ-LEZAUN, AND M. W. FELDMAN, *Population growth of human Y chromosomes: a study of Y chromosome microsatellites*, Molecular Biology and Evolution, 16 (1999), pp. 1791–1798.

[188] A. E. RAFTERY AND S. LEWIS, *How many iterations in the Gibbs sampler?*, tech. rep., Washington University Seattle Department of statistics, 1991.

[189] P. REBESCHINI AND R. V. HANDEL, *Can local particle filters beat the curse of dimensionality?*, The Annals of Applied Probability, 25 (2015), pp. 2809–2866.

[190] L. REY-BELLET AND K. SPILIOPOULOS, *Irreversible Langevin samplers and variance reduction: a large deviations approach*, Nonlinearity, 28 (2015), p. 2081.

[191] C. ROBERT AND G. CASELLA, *A short history of Markov chain Monte Carlo: Subjective recollections from incomplete data*, Statistical Science, (2011), pp. 102–115.

[192] ———, *Monte Carlo Statistical Methods*, Springer Science & Business Media, 2013.

[193] C. P. ROBERT, G. CASELLA, AND G. CASELLA, *Introducing Monte Carlo Methods with R*, vol. 18, Springer, 2010.

[194] C. P. ROBERT, J.-M. CORNUET, J.-M. MARIN, AND N. S. PILLAI, *Lack of confidence in approximate Bayesian computation model choice*, Proceedings of the National Academy of Sciences, 108 (2011), pp. 15112–15117.

[195] G. O. Roberts, A. Gelman, and W. R. Gilks, *Weak convergence and optimal scaling of random walk Metropolis algorithms*, The Annals of Applied Probability, 7 (1997), pp. 110–120.

[196] G. O. Roberts and N. G. Polson, *On the geometric convergence of the Gibbs sampler*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 56 (1994), pp. 377–384.

[197] G. O. Roberts and J. S. Rosenthal, *Optimal scaling for various Metropolis-Hastings algorithms*, Statistical Science, 16 (2001), pp. 351–367.

[198] G. O. Roberts and J. S. Rosenthal, *General state space Markov chains and MCMC algorithms*, Probability Surveys, 1 (2004), pp. 20–71.

[199] ———, *Examples of adaptive MCMC*, Journal of Computational and Graphical Statistics, 18 (2009), pp. 349–367.

[200] G. O. Roberts and S. K. Sahu, *Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 59 (1997), pp. 291–317.

[201] G. O. Roberts and A. M. Smith, *Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms*, Stochastic Processes and their Applications, 49 (1994), pp. 207–216.

[202] G. O. Roberts and R. L. Tweedie, *Exponential convergence of Langevin distributions and their discrete approximations*, Bernoulli, 2 (1996), pp. 341–363.

[203] M. Rosenblatt, *Remarks on a multivariate transformation*, The Annals of Mathematical Statistics, 23 (1952), pp. 470–472.

[204] D. B. Rubin, *Bayesianly justifiable and relevant frequency calculations for the applied statistician*, The Annals of Statistics, 12 (1984), pp. 1151–1172.

[205] H. Rue and L. Held, *Gaussian Markov random fields: theory and applications*, CRC Press, 2005.

[206] D. Sanz-Alonso, *Importance sampling and necessary sample size: An information theory approach*, SIAM/ASA Journal on Uncertainty Quantification, 6 (2018), pp. 867–879.

[207] D. Sanz-Alonso, A. Stuart, and A. Taeb, *Inverse Problems and Data Assimilation*, vol. 107, Cambridge University Press, 2023.

[208] D. Sanz-Alonso and Z. Wang, *Bayesian update with importance sampling: Required sample size*, Entropy, 23 (2021), p. 22.

[209] J. M. Sanz-Serna and M. P. Calvo, *Numerical Hamiltonian Problems*, Courier Dover Publications, 2018.

[210] T. Schwedes and B. Calderhead, *Rao-Blackwellised parallel MCMC*, in International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 3448–3456.

[211] C. Sherlock, A. Golightly, and D. A. Henderson, *Adaptive, delayed-acceptance MCMC for targets with expensive likelihoods*, Journal of Computational and Graphical Statistics, 26 (2017), pp. 434–444.

[212] K. D. Siegmund, P. Marjoram, and D. Shibata, *Modeling DNA methylation in a population of cancer cells*, Statistical Applications in Genetics and Molecular Biology, 7 (2008).

[213] S. A. Sisson, Y. Fan, and M. M. Tanaka, *Sequential Monte Carlo without likelihoods*, Proceedings of the National Academy of Sciences, 104 (2007), pp. 1760–1765.

[214] C. SNYDER, *Particle filters, the optimal proposal and high-dimensional systems*, Proceedings of the ECMWF Seminar on Data Assimilation for Atmosphere and Ocean, (2011), pp. 1–10.

[215] C. SNYDER, T. BENGTSSON, P. BICKEL, AND J. ANDERSON, *Obstacles to high-dimensional particle filtering*, Monthly Weather Review, 136 (2016), pp. 4629–4640.

[216] C. SNYDER, T. BENGTSSON, AND M. MORZFELD, *Performance bounds for particle filters using the optimal proposal*, Monthly Weather Review, 143 (2015), pp. 4750–4761.

[217] A. S. STORDAL, H. A. KARLSEN, G. NÆVDAL, H. J. SKAUG, AND B. VALLÈS, *Bridging the ensemble Kalman filter and particle filters: the adaptive Gaussian mixture filter*, Computational Geosciences, 15 (2011), pp. 293–305.

[218] A. M. STUART, *Inverse problems: a Bayesian perspective*, Acta Numerica, 19 (2010), pp. 451–559.

[219] R. H. SWENDSEN AND J.-S. WANG, *Replica Monte Carlo simulation of spin-glasses*, Physical Review Letters, 57 (1986), p. 2607.

[220] H. SZU AND R. HARTLEY, *Fast simulated annealing*, Physics letters A, 122 (1987), pp. 157–162.

[221] Z. TAN, *On a likelihood approach for Monte Carlo integration*, Journal of the American Statistical Association, 99 (2004), pp. 1027–1036.

[222] M. M. TANAKA, A. R. FRANCIS, F. LUCIANI, AND S. A. SISSON, *Using approximate Bayesian computation to estimate tuberculosis transmission parameters from genotype data*, Genetics, 173 (2006), pp. 1511–1520.

[223] S. TAVARÉ, D. J. BALDING, R. C. GRIFFITHS, AND P. DONNELLY, *Inferring coalescence times from DNA sequence data*, Genetics, 145 (1997), pp. 505–518.

[224] L. TIERNEY, *Markov chains for exploring posterior distributions*, The Annals of Statistics, (1994), pp. 1701–1728.

[225] L. TIERNEY, *A note on Metropolis-Hastings kernels for general state spaces*, The Annals of Applied Probability, 8 (1998), pp. 1–9.

[226] B. M. TURNER AND T. VAN ZANDT, *A tutorial on approximate Bayesian computation*, Journal of Mathematical Psychology, 56 (2012), pp. 69–85.

[227] P. J. M. VAN LAARHOVEN AND E. H. L. AARTS, *Simulated Annealing: Theory and Applications*, Springer, 1987.

[228] D. VATS AND C. KNUDSON, *Revisiting the Gelman–Rubin diagnostic*, Statistical Science, 36 (2021), pp. 518–529.

[229] A. VEHTARI, A. GELMAN, D. SIMPSON, B. CARPENTER, AND P.-C. BÜRKNER, *Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC (with discussion)*, Bayesian Analysis, 16 (2021), pp. 667–718.

[230] R. VERSHYNIN, *High-dimensional Probability: An Introduction with Applications in Data Science*, vol. 47, Cambridge University Press, 2018.

[231] J. VON NEUMANN, *Various techniques used in connection with random digits*, Appl. Math Ser, 12 (1951), p. 5.

[232] M. J. WAINWRIGHT AND M. I. JORDAN, *Graphical Models, Exponential Families, and Variational Inference*, vol. 1, Now Publishers, Inc., 2008.

[233] H. WANG, *Monte Carlo Simulation with Applications to Finance*, CRC Press, 2012.

[234] G. Weiss and A. von Haeseler, *Inference of population history using a likelihood approach*, Genetics, 149 (1998), pp. 1539–1546.

[235] M. Welling and Y. W. Teh, *Bayesian learning via stochastic gradient Langevin dynamics*, in Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011, pp. 681–688.

[236] A. Wibisono, *Sampling as optimization in the space of measures: The Langevin dynamics as a composite optimization problem*, in Conference on Learning Theory, PMLR, 2018, pp. 2093–3027.

[237] R. D. Wilkinson, *Approximate Bayesian computation (ABC) gives exact results under the assumption of model error*, Statistical Applications in Genetics and Molecular Biology, 12 (2013), pp. 129–141.

[238] C. J. Wu, *On the convergence properties of the EM algorithm*, The Annals of Statistics, (1983), pp. 95–103.

[239] J. Yang, G. O. Roberts, and J. S. Rosenthal, *Optimal scaling of random-walk Metropolis algorithms on general target distributions*, Stochastic Processes and their Applications, 130 (2020), pp. 6094–6132.

[240] B. Yu and P. Mykland, *Looking at Markov samplers through cusum path plots: a simple diagnostic idea*, Statistics and Computing, 8 (1998), pp. 275–286.

[241] F. Zhang and C. Gao, *Convergence rates of variational posterior distributions*, The Annals of Statistics, 48 (2020), pp. 2180–2207.