M4A44: Computational Stochastic Processes

A. Duncan

March 22, 2016

- Lectures:
  - Mondays 13:00-14:00, Huxley 342 ;
  - Tuesdays 12:00-13:00 Huxley 139 ;
  - Fridays 9:00-10:00 Huxley 658;
- Office Hours: Monday 9:00 - 10:00, Tuesday 9:00 - 10:00 or by appointment.
- Course webpage:
  M4A44 Blackboard Page
- Assessment: 25% assignment (to be handed out 1^{st} week of March, and returned 18^{th} March) and 75% final Examination (May/June 2016)

Contents

Chapter 1

Introduction

Some motivating examples and questions.

Applied mathematicians have made use of ordinary and partial differential equations to model dynamic phenomena such as fluid flow, molecular motion, climate dynamics, it has become clear that introducing randomness into mathematical models of real-world phenonema is an extremely powerful and useful idea. Noise is introduced into models for a number of reasons. Firstly, it can be used to *reflect uncertainty* within the model: for example, parameters within models, for example material properties, boundary conditions, etc, will never be known exactly, and a more robust model should be able to reflect this uncertainty. There are numerous other forms of uncertainty one might wish to incorporate beyond the variability of the parameters. Structural uncertainty, for example, reflects our ignorance about part of the model, for example, lack of knowledge of the underlying true physics.

Secondly, noise can be introduced as a means to *reduce complexity* of a model: if an existing model is too complex to be studied or simulated, then it is not very useful. In many cases however, we can replace part of the model with a random noise term which (at least in a statistical sense) exhibits the same behaviour. Such methods permit us to approximate (in an appropriate sense) a deterministic, but extremely complex high-dimensional model (which would be impossible to simulate on a computer) with a tractible low-dimensional model. One particular approach to model reduction is the *Mori-Zwanzig* formalism, was originally used to approximate complex (deterministic) molecular dynamical models by a much simpler Langevin equation (which we will discuss in Chapter 4).

Certainly, stochastic models have been applied with success to model phenomena arising in almost every field of science, from the traditional origin of statistical physics, to cell biology, epidemiology, climate science and medicine, not to mention economics and finance. The objective of these lecture notes are to understand some of the common computational tasks involved in the construction, simulation and prediction using these stochastic models. The particular focus of this course will be on computational problems related to *stochastic processes*. Stochastic processes describe dynamical systems whose time evolution is of a probabilistic nature. The precise definition is given below:

Definition 1.1 (**Stochastic process**). *Let T be a totally-ordered set, $(\Omega, \mathcal{F}, \mathbb{P})$ a probability space, and $(E, \mathcal{G})$ a measurable space. A stochastic process is a collection of random variables*

$X = \{X_t\,;\, t \in T\}$ *such that for each fixed* $t \in T$*,* X_t *is a random variable from* $(\Omega, \mathcal{F}, \mathbb{P})$ *to* $(E, \mathcal{G})$*. The set* Ω *is known as the sample space, and* E *is said to be the state space of the stochastic process* X_t*.*

The index set T will typically either be $T = \mathbb{Z}$ or $T = \mathbb{N}$ in which case X_t is said to be a *discrete time* stochastic process; or $T = \mathbb{R}$ or $T = \mathbb{R}_+ = \{x \in \mathbb{R}\,;\, x \geqslant 0\}$ in which case X_t is said to be a *continuous time* stochastic process. During this course, the state space E will be either $\mathbb{R}^n$ or $\mathbb{Z}$ equipped with the Borel σ-algebra, unless otherwise stated.

A stochastic process X_t may be viewed as a function of both $t \in T$ and $\omega \in \Omega$. We sometimes write $X(t), X(t, \omega)$ or $X_t(\omega)$. There are two ways of viewing the stochastic process: If we fix ω, we can consider the (non-random) map:

$$t \to X(t, \omega) \in E, \qquad \text{for fixed } \omega \in \Omega,$$

i.e. we are looking at the path $X_t(\omega) =: \omega(t)$, i.e. we identify the sample space Ω with the set of paths from 0 to T. Alternatively, we can fix t and consider the map

$$\omega \to X(t, \omega) \in E, \qquad \text{for fixed } t \in T,$$

then this is a random variable, which gives us a snapshot of what is happening (non-deterministically) to all sample points $\omega \in \Omega$ at a fixed time t. Heuristically, this view corresponds X_t being obtained by performing an experiment at each time $t \in T$, which determines the evolution of the stochastic process. Although both viewpoints are equivalent, both can useful in different contexts, as we shall see in the coming chapters.

In this course, we shall primarily be interested in *Markov processes*. For discrete time processes this means that

$$\mathbb{P}\left[X_n = x_n \,|X_{n-1} = x_{n-1}, \ldots, X_0 = x_0\right] = \mathbb{P}\left[X_n = x_n \,|X_{n-1} = x_{n-1}\right].$$

We shall call such a process a *Markov chain*. The analogous concept generalised to continuous time Markov chains naturally. Within this course, we shall only consider very particular type of continous time process, namely diffusion processes, which are characterised as being the solution of a stochastic differential equation (SDE) of the form

$$dX_t = b(X_t)\,dt + \sqrt{2\sigma^2(X_t)}\,dW_t,$$

where W_t is a standard Brownian motion. We shall discuss the properties of this process in detail in Chapter 4.

First, let us introduce some motivating examples and identify some of the computational challenges we wish to address.

1.1 Motivation 1: Monte Carlo Methods

Typically, we interested in computing expectations of some function f with respect to the distribution π, i.e.

$$I = \mathbb{E}_{X\sim\pi}[f(X)] = \int f(x)\pi(x)\,dx.$$

Integrals of these forms arise frequently when computing properties of statistical models. For example, computing probabilities of the form $\mathbb{P}[a < Y < b]$ can clearly be expressed as the expectation of an appropriate observable f. There might be several reasons which such an integral is impossible to compute directly. For example, suppose the state space is restricted to the unit square $\Omega = [0,1]^d$. In this case, one could resort to numerical quadrature to approximate I to compute the integral directly. For example, using a regular mesh of $K \times K \ldots K$ points, using the standard midpoint rule one can show that the error will be $O(K^{-2})$ (provided f is smooth). However, the number of evaluations of f and π will be $M = N^d$. Thus, in terms of the computational effort, the error will be $O\left(N^{-\frac{2}{d}}\right)$. While this is fine for moderately high dimensions, as d increases, the number of evaluations must grow exponentially to maintain the same error. This problem is known as the *curse of dimensionality*. There are also other issues may also make direct approximation of I impossible, for example if we don't have an explicit or computable formula for π.

However, if we are able to generate a sequence of iid samples $x_1, x_2, \ldots$ of π, we know from the law of large numbers that

$$\frac{1}{N}\sum_{n=1}^{N} f(x_n) \to \mathbb{E}[f(X)] = \int f(x)\pi(x)\,dx, \qquad \text{as } N \to \infty,$$

where $X \sim \pi$. Thus we can approximate the integral I using the approximation

$$I_N = \frac{1}{N}\sum_{n=1}^{N} f(x_n),$$

knowing that, as $N \to \infty$, I_N converges to I. This is the general idea of Monte-Carlo methods. In 1996, Alan Sokal wrote:

> "Monte Carlo is an extremely bad method; it should be used only when all alternative methods are worse".

Why is this so? As we shall see, the rate of convergence is $O(N^{-1/2})$, which basically means that

$$\text{error} \sim \frac{1}{\sqrt{\text{computational cost}}}.$$

When compared to other methods for computing integrals this is horrendous! Indeed, there exist quadrature schemes which can compute I with error scaling like $O(\text{cost}^{-3})$ or even $O(e^{-\text{cost}})$. However, as in the previously described example, these methods will all suffer from the curse of dimensionality, making them infeasible to use in high–dimensions. Monte Carlo methods however, do not suffer from the curse, and it is in this scenario where a Monte–Carlo method would have a strong advantage. Molecular dynamics problems, where the number of dimensions is typically $10^3 - 10^6$ certainly falls into this category, as do many others. In this course we shall address the following questions:

Generating Random Objects: How do we generate iid samples from π?

Stopping criteria: How many samples do we need to generate to obtain a sufficiently good approximation of I?

Performance of MC methods: How do we measure the performance of MC methods, and can we develop techniques to speed up the convergence of I_N to I?

MCMC methods: If we cannot sample from π directly, but have a stochastic process which has distribution π at equilbrium, can we develop an MC method using this process to approximate I?

1.2 Motivation 2: Computational Statistical Physics

Consider a microscopic system composed of M particles (typically atoms). The state of the system is described by the positions of the particles $q = (q_1, \ldots, q_M) \in \mathcal{R}^{3M}$ and the associated momenta $p = (p_1, \ldots, p_M) \in \mathbb{R}^{3M}$. The interactions between the particles are taken into account through a potential function V which depend only on position. The evolution of an isolated system is governed by the Hamiltonian dynamics

$$\ddot{q}(t) = -\nabla V(q(t)). \tag{1.1}$$

with initial conditions $(q(0), \dot{q}(0))$ specified. The *Langevin process* is a model of a Hamiltonian system coupled to an infinite reservoir of energy coupled via a thermostat. This model arises through model reduction of a more complex system, and indeed, can be derived via the Mori-Zwanzig formalism. The corresponding stochatic equations are given by

$$\ddot{q}_t = -\nabla V(q_t) - \gamma \dot{q}_t + \sqrt{2\gamma\beta^{-1}}\dot{W}_t, \tag{1.2}$$

where W_t is a $3M$–dimensional Brownian motion and β^{-1} and γ are parameters which characterise the temperature and friction in the model. We will defer the precise interpretation of equation (1.2) until Chapter 4, and for now, will merely consider (1.2) to describe an ODE subject to random, Gaussian distributed "kicks" in the momenta. To make sense of this equation, we can express the Langevin process as a pair of coupled first order SDEs:

$$\begin{aligned} dq(t) &= p(t)\,dt \\ dp(t) &= -\nabla_q V(q(t))\,dt - \gamma p(t)\,dt + \sqrt{2\beta^{-1}\gamma}\,dW(t). \end{aligned} \tag{1.3}$$

Another model which is frequently used is the *overdamped Langevin equation* given by the solution of the following SDE:

$$dX_t = -\nabla V(X_t)\,dt + \sqrt{2\beta^{-1}}\,dW_t,$$

this arises from the Langevin equation (1.2) in the $\gamma \to \infty$ limit. To study the dynamics of these models, clearly we will require some means of simulation of the processes. While it is possible to obtain explicit expressions for the solution of simple SDEs, in general there will no closed-form solution. Thus, as we do for ODEs, we typically must resort to numerical approximations to simulate the process. This motivates the following question:

Discretisation: Given a step size $\Delta t \ll 1$, can we derive numerical discretisations $X^{(n)})$ and $(q^{(n)}, p^{(n)})$ which provide good approximations to $X_{n\Delta t}$ and $(q_{n\Delta t}, p_{\Delta t})$ respectively? What is the natural sense in which this approximation should hold?

Stability: If we are guaranteed that the process X_t remains finite for all time $t > 0$, what conditions must we assume to ensure that the corresponding discretisation is also stable?

Under appropriate conditions, as $t \to \infty$ the distribution of the position process X_t will approach the *Boltzmann distribution*

$$\pi(x) = \frac{1}{Z} e^{-\beta V(x)},$$

where β is the inverse temperature, and Z is the normalisation constant (known as the partition function), i.e.

$$Z = \int_{\mathbb{R}^{3M}} e^{-\beta V(x)}\, dx.$$

The distribution $\pi(x)$ thus characterises the fluctuations of the process X_t *at equilibrium*. Very frequently, we are only interested in the equilibrium behaviour of the molecular system, and not the transient behaviour.

Steady-State simulation: How can we generate samples from $\pi(x)$? If we use the approximation $X^{(n)}$ as $n \to \infty$, will the resulting distribution approximate π, or will the discretisation errors accumulate? If so, can we modify the process $X^{(n)}$ to ensure that we sample exactly from π?

1.3 Motivation 3: Inference Problems and Model Fitting

A central problem in statistics is to infer unobserved parameters from a sample of observed data. In parametric inference, we suppose that θ is the parameter we wish to infer, based on a a random vector $\mathbf{y}$ of observed data, which is assumed to have distribution $l\,(\mathbf{y}\,|\,\theta)$. This distribution is known as the *likelihood* and is assumed to be completely known from the model. Our objective is to find a value of θ which is most compatible with the observed data $\mathbf{y}$. This process is some times known as *model fitting*.

One method of inference is known as *maximum likelihood estimation* (MLE), where the parameter θ is estimated by $\hat{\theta}$ which is the solution of the following optimisation problem:

$$\hat{\theta} = \text{argmax}\ _{\theta\in\Theta} l(\mathbf{y}\,|\,\theta),$$

where Θ is the set of admissable values of the unknown parameter. Another approach is based on Bayes' rule. In this approach we view θ and $\mathbf{y}$ as a coupled pair of random variables. The likelihood $l(\mathbf{y}\,|\,\theta)$ specifies the conditional density of $\mathbf{y}$ conditioned on a particular value θ. Suppose we have an initial *prior* distribution $\pi_0(\theta)$ on the unobserved parameter, in which we encode any prior information we might have about the parameter θ. By Bayes' rule we know that

$$\pi(\theta\,|\,\mathbf{y}) = \frac{l\,(\mathbf{y}\,|\theta)\ \pi_0(\theta)}{\mathbb{P}[\mathbf{y}]}. \tag{1.4}$$

The probability density $\pi(\theta\,|\mathbf{y})$ is known as the *posterior* distribution. As opposed to the maximum likelihood estimator, this distribution characterises our certainty and uncertainty about the value of the parameter θ given what we've already observed. Indeed, the posterior variance can be viewed as a measure of certainty on the value of θ, and if the variance is small, this suggests that the posterior mean might be a good point estimate for θ. A complication arises due to the denominator

$$\mathbb{P}[\mathbf{y}] = \int_{\Theta} l(\mathbf{y}\,|\,\theta)\pi_0(\theta)\,d\theta.$$

This is a high-dimensional integral and is in general difficult to compute. One strategy would be to choose a prior in such a way that $\mathbb{P}[\mathbf{y}]$ is known. Another alternative is to compromise between MLE and Bayesian inference, and use a *Maximum a posteriori estimator*, namely use the mode of the posterior distribution as a best guess for θ. Since the mode doesn't depend on the normalisation constant, then we need not compute $\mathbb{P}[\mathbf{y}]$. Another possibility is to resort to Markov-Chain Monte Carlo methods (MCMC), i.e. we will construct a stochastic process $(\theta_n)_{n\in\mathbb{N}}$, whose stationary distribution is equal to $\pi(\theta\,|\,\mathbf{y})$. For n large enough, the distribution of θ_n will be very close to the desired distribution. In Section 5 we shall make this more precise.

Chapter 2

Monte Carlo Simulation

Generating uniformly–distributed random numbers, sampling from non-uniform probability distributions, Estimators and Monte-Carlo mehods, Variance Reduction Methods for MC simulation.

2.1 Generating Uniform Random Numbers

All the methods that we shall describe within this course inherently assume the availability of a stream $u_1, u_2, u_3 \ldots$ of random numbers which are the realisation of a sequence of independent, identically random variables which are $U(0,1)$–distributed, i.e. having uniform distribution on $(0,1)$. In general, there are two main methods to produce such a stream. The first approach relies on some physical phenomenon that is expected to be random which can be measured to obtain a stream of random numbers with a given distribution. One such approach (used by `HotBits` service at Fourmilab in Switzerland) involves measuring radioactive decay: the emission times of particles from a radioative source, measured using a Gieger-Muller tube. The times between successive decay events are known to be iid exponentially distributed random variables, which are then transformed to a $U(0,1)$–distributed iid sequence. Other approaches which use a physical source of noise involve measuring atmospheric noise (as is done at `Random.org`), etc.

The second method involves using a deterministic algorithm which can produce sequences of numbers which in a statistical sense is very close to being random (i.e. the sequence passes a stringent number of statistical tests). These *pseudo-random number generators* (PRNGs) are the standard way of generating uniform random numbers of computers nowadays. Although it was not uncommon in the past that software packages would make use of PRNGs with poor properties, most modern software libraries provide high-quality PRNG routines.

Virtually all pseudo-random number generators can be viewed as a recursive algorithm, for which, given an initial *seed* x_0, produces a sequence $u_0, u_1, u_2, \ldots \in [0,1]$, constructed by

$$u_i = g(x_i), \text{ where } x_i = f(x_{i-1}), \quad i \geqslant 1.$$

We note that if we use the same value of x_0 in two separate simulations then the PRNG will produce the same sequence of pseudorandom numbers. In practice the set of possible values of the $(x_i)_{i\in\mathbb{N}}$ is finite, and thus the PRNG will eventually repeat, i.e. $x_{l+d} = x_d$ for some l. The smallest value of d for which this occurs is known as the *period* of the PRNG. Clearly we want the period to be as long as possible.

Example 2.1. *A simple example of a PRNG is the* linear congruential generator *(LCG), defined by the recurrence relation:*

$$u_{n+1} = \frac{x_{n+1}}{M}, \qquad \textit{where } x_{n+1} = (ax_n + c) \textit{ mod } M.$$

Clearly, the period of the linear congruential generator is less or equal to M, but will be less in general, depending on the choice of a and c. The Hull-Dobell theorem provides necessary and sufficient conditions which a, c and M must satisfy so that the LCG has full period for all seed values. For example glibc's `rand()` implementation uses:

$$M = 2^{32} \quad a = 22695477, \textit{ and } c = 1.$$

In practice, the LCG should not be used in Monte–Carlo simulations, as it does not produce sufficiently independent samples. Most modern software packages make use of high quality PRNGs such as the *Mersenne Twister*, which is used by `MATLAB`, `Julia`, and `GNU-R`. For the remainder of the module we shall *assume* that we are provided with a sequence of iid uniformly distributed random numbers, without worrying about their provenance.

2.2 Generating Non–Uniform Random Numbers

Code examples for this section can be found in:
`http://nbviewer.jupyter.org/url/dl.dropboxusercontent.com/u/65686487/workbook1.ipynb`

2.2.1 Inverse Transform Method

Suppose now that we wish to produce samples of a random variable X with non-uniform distribution. For some one–dimensional random variables we can use the *inverse transform method*. Indeed, suppose that X has cumulative distribution function $F(x)$, i.e.

$$F(x) = \mathbb{P}\left[X \leqslant x\right].$$

If the cumulative distribution function is strictly increasing and continuous, then define $G(u) = F^{-1}(u)$, i.e. $x = G(u)$ is the unique solution of $F(x) = u$. In the more general scenario (for example if the distribution has jumps), then we define G to be the following function:

$$G(u) := \inf\left\{x \,:\, F(x) \geqslant u\right\}, \quad 0 < u < 1.$$

Then, once again, for $0 < u < 1$, it follows that $F(G(u)) = u$. The inverse transform method for sampling producing samples of X is then as follows:

The Inverse Transform Method

1. Generate a random number u from $U(0, 1)$.
2. Compute $x = G(u)$.
3. Take x to be a sample of the random variable X with cdf F.

The reason this approach works follows from the subsequent lemma.

Lemma 2.1. *If U is a random variable on uniformly distributed on $(0,1)$, then the random variable $G(U)$ has cdf F.*

Proof. One can check that,

$$G(u) \leqslant x \iff u \leqslant F(x).$$

Let $x \in \mathbb{R}$ and consider the cdf of $G(U)$:

$$\mathbb{P}\left[G(U) \leqslant x\right] = \mathbb{P}\left[U \leqslant F(x)\right] = F(x),$$

since U is uniformly distributed. □

Example 2.2. *Suppose we wish to sample from an expontential distribution $Exp(\lambda)$ with rate λ. The cdf on $[0,\infty)$ is given by $F(x) = 1 - e^{-\lambda x}$. Applying the inverse transform method it follows that*

$$X := -\frac{1}{\lambda}\log(1-U), \quad U \sim U(0,1)$$

is $Exp(\lambda)$–distributed. Note that, since $1 - U$ is also $U(0,1)$–distributed, we can simply use $X = -\frac{1}{\lambda}\log(U)$.

Exercise 2.1 (Generalized Bernoulli distribution)**.** *Suppose X is a discrete valued-random variable taking values i with probability p_i for $i = 1,\ldots,k$ where $\sum_{i=1}^{k} p_i = 1$. Use the inverse transform method to derive an algorithm to sample from this distribution.*

Exercise 2.2 (Cauchy Distribution)**.** *The standard Cauchy distribution is a continuous probability distribution having probability density function*

$$f(x) = \frac{1}{\pi(1+x^2)}.$$

It is the distribution of a random variable given by the ratio of two independent standard Gaussian variables. Use the inverse transform method to derive an algorithm to sample from this distribution.

Exercise 2.3 (Logistic Distribution)**.** *The logistic distribution is a continuous probability distribution whose cumulative distribution function is the logistic function, i.e.*

$$F(x) = \frac{e^x}{1+e^x}$$

Use the inverse transform method to derive an algorithm to sample from this distribution.

2.2.2 Rejection Sampling

Suppose that we wish to generate samples of a random variable X with a known and computable density $f(x)$ (which still might have a very complicated form). If there is a density $g(x)$ on the same state space from which we can easily generate samples, and the condition

$$f(x) \leqslant Mg(x), \quad \forall x, \tag{2.1}$$

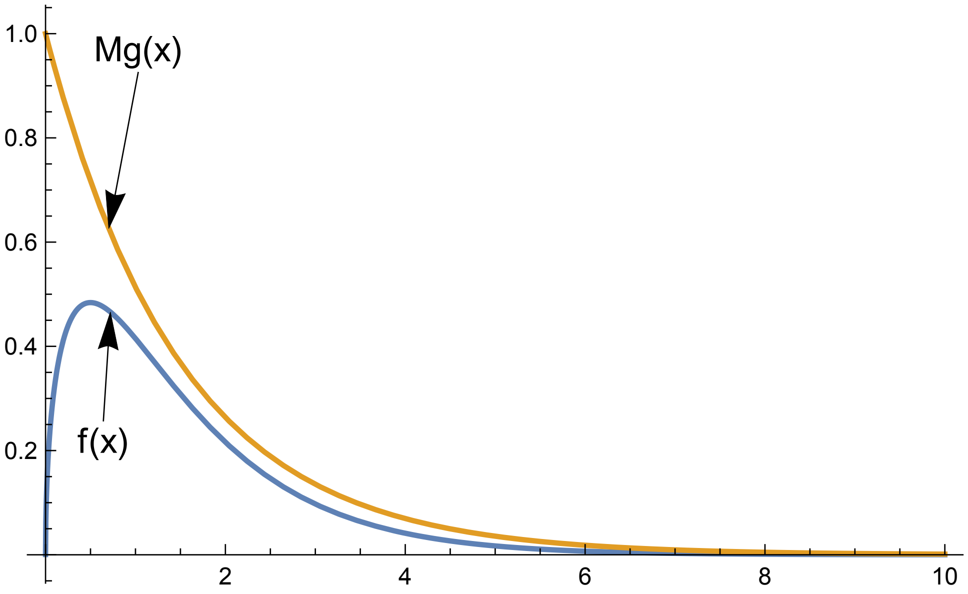


Figure 2.1: Target distribution $f(x)$ with proposal distribution $g(x)$

holds for some constant $M < \infty$, (see Figure 2.1) then we can use rejection sampling to generate samples of the density $f(x)$ from a stream of samples of $g(x)$. The idea is that we generate a sample x from the proposal density $g(x)$. Then one accepts $X = x$ with probability $f(x)/Mg(x)$, otherwise we reject it and repeat until a sample is finally accepted. The algorithm is given as follows:

Rejection Sampling Method

1. Generate sample x from density $g(\cdot)$ and let $u \sim U(0,1)$.
2. If $u < \frac{f(x)}{Mg(x)}$, then accept the sample $X = x$ and stop.
3. Otherwise, reject the sample and return to step 1.

Remark 2.2. *The Rejection Sampling Method can be used to sample from both discrete and continuous distributions: in the discrete case we replace the pdfs f and g by pmfs.*

For the continuous case, we first prove that the samples generated by the rejection algorithm are distributed according to $f(x)$.

Lemma 2.3. *Let Z be the random variable given by the output of rejection sampling algorithm. Then Z is distributed according to the density f.*

Proof. We shall assume that the state space E is given by $(-\infty, +\infty)$, noting that the result holds for other domains in a similar manner. Let Y be a random variable having density g.

Denote by $A = 1$ the event that the sample with distribution Y was accepted, i.e. the event $U < f(Y)/Mg(Y)$ where $U \sim U(0,1)$, $Y \sim g(\cdot)$. Then the probability of an acceptance event occurring is given by

$$\mathbb{P}\,[A=1] = \int_{-\infty}^{+\infty} \mathbb{P}\,[A=1\,|Y=y]\,g(y)\,dy = \int_{-\infty}^{+\infty} g(y)\frac{f(y)}{Mg(y)}\,dy = \frac{1}{M}\int_{-\infty}^{+\infty} f(y)\,dy = \frac{1}{M},$$

and moreover, for fixed $r \in \mathbb{R}$:

$$\mathbb{P}\,[Y < r, A=1] = \int_{-\infty}^{r} \left[\int_{0}^{f(x)/Mg(x)} du\right] g(x)\,dx = \frac{1}{M}\int_{-\infty}^{r} f(x)\,dx.$$

Therefore,

$$\mathbb{P}\,[Z \leqslant r] = \mathbb{P}\,[Y \leqslant r\,|\,A=1] = \frac{\mathbb{P}\,[Y \leqslant r\,,\,A=1]}{\mathbb{P}[A=1]} = \int_{-\infty}^{r} f(x)\,dx,$$

so that the random variable Z has the desired density f. □

Since f and g are both densities (i.e. integrate to 1), it follows that $M \geqslant 1$. For the sake of efficiency we want the rejection algorithm to reject as few samples as possible. From the previous proof, the acceptance probability is given by $\mathbb{P}[A=1] = \frac{1}{M}$. Thus on average the rejection algorithm must generate M proposals to produce a single sample. Thus, for the best performance, the bound M must be chosen as close to 1 as possible, in particular g should as close to f as possible.

Example 2.3 (Sampling from the Gamma distribution)**.** *The gamma distribution of order $k \in \mathbb{N}$, $k > 0$, is the waiting time of the k^{th} event in a Poisson random process of unit mean. When $k = 1$ it is the waiting time for the first event, i.e. just an exponential distribution. The gamma distribution has probability density*

$$f(x) = \frac{x^{a-1}e^{-x}}{\Gamma(a)}, \quad x > 0.$$

To sample from this distribution we can use rejection sampling using a Cauchy distribution as proposal. We first need to establish condition (2.1). Computing the ratio of $f(x)/g(x)$ for $a > 1$:

$$\begin{aligned} f(x)/g(x) &= \frac{\pi}{\Gamma(a)}x^{a-1}(1+x^2)e^{-x} \\ &= \frac{\pi}{\Gamma(a)}\left(e^{-x+(a-1)\log(x)} + e^{-x+(a+1)\log(x)}\right) \end{aligned}$$

The term $(a-1)\log(x) - x$ attains its maximum when $x = a-1$ and similarly the term $(a+1)\log(x) - x$ attains it maximum when $x = a+1$, so we can bound both terms above by:

$$f(x)/g(x) \leqslant \frac{\pi}{\Gamma(a)}\left((a-1)^{(a-1)}e^{-(a-1)}\pi + \pi(a+1)^{a+1}e^{-(a+1)}\right) =: M.$$

As $a \to \infty$, the value of M behaves like $a^{3/2}$ (exercise), so that the performance of the rejection sampler becomes poor for large a. Indeed, this is an extremely poor choice of proposal density! In the exercise sheets we shall carefully investigate a much more reasonable choice for g.

However, a rejection algorithm based on the above example would have a problem: if $a \in \mathbb{N}$ then $\Gamma(a) = (a-1)!$ which we can compute exactly, but if $a \notin \mathbb{N}$, then we must resort to computing integrals to approximate this special function. If possible we should avoid computing the normalisation constant entirely.

Indeed, in many cases arising in applications, one does not know the normalising constant of the target density f, i.e. $\int_{-\infty}^{\infty} f(x)\,dx = Z \neq 1$. Similarly, even though one might be able to generate samples with distribution g, the normalisation constant needn't be known. In this case, we can still apply rejection sampling. Indeed, suppose

$$\int_{-\infty}^{\infty} f(x)\,dx = Z \quad \text{and} \int_{-\infty}^{\infty} g(x)\,dx = Z',$$

and

$$f(x) \leqslant Mg(x), \quad x \in \mathbb{R}. \tag{2.2}$$

If $\tilde{f}(x)$ and $\tilde{g}(x)$ are the corresponding normalisation distributions, then (2.2) is equivalent to

$$\tilde{f}(x) = \frac{f(x)}{Z} \leqslant M'\frac{g(x)}{Z'} = M'\tilde{g}(x), \quad x \in \mathbb{R},$$

where $M' = \frac{Z'M}{Z}$. In particular, if we implement a rejection sampling algorithm for $\tilde{f}$ using $\tilde{g}$ as proposal, then the accept/reject condition for accepting a sample $x \sim g$ is given by

$$u \leqslant \frac{\tilde{f}(x)}{M'\tilde{g}(x)}, \text{ where } u \sim U(0,1).$$

From the definition of M', $\tilde{f}$ and $\tilde{g}$ this is equivalent to

$$u \leqslant \frac{f(x)}{Mg(x)}, \text{ where } u \sim U(0,1).$$

The implication of this is that we can safely ignore the normalising constant from the start: accepting a sample with probability $f(x)/Mg(x)$ is equivalent to $\tilde{f}(x)/M'\tilde{g}(x)$. In this case, the probability of accepting a proposal is Z/MZ'.

Exercise 2.4. *Repeat the above exercise using unnormalised densities.*

Exercise 2.5 (Generating Gaussians using Rejection Sampling)**.** *Using the Cauchy distribution as proposal, use the rejection algorithm to generate samples from the standard Gaussian distribution $f(x) = e^{-x^2/2}/\sqrt{2\pi}$. Would it be possible to work the other way round, i.e., use rejection sampling to produce Cauchy distributed draws from using a Gaussian proposal distribution?*

2.2.3 Sampling from Gaussian distributions

Ideally we would like to sample from the Gaussian distribution using the inverse tranform method, however there is no closed form formula for the cdf $\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-y^2/2}\,dy$. One possible approach outlined in Exercise 2.5 suggests using a rejection sampler with a Cauchy proposal distribution. However, it turns out that by applying a clever transformation, one can directly obtain a *pair* of iid standard Gaussian random variables from a pair of independent $U[0, 1]$–distributed RVs.

One such method is the *Box-Muller algorithm*, which is based on the observation that a pair (X, Y) of independent standard normals will have pdf:

$$f(x, y) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} = \frac{1}{2\pi} e^{-(x^2+y^2)/2}.$$

Consider the random variables (R, Θ) where $R > 0$ and $0 \leqslant \Theta < 2\pi$ such that

$$(X, Y) = R(\cos(\Theta), \sin(\Theta)),$$

which correspond to a polar coordinate representation of (X, Y). Clearly, since the (X, Y) is rotationally symmetric around the origin, the variable Θ is uniformly distributed on $[0, 2\pi]$. So we can write $\Theta = 2\pi U_1$, where $U_1 \sim U[0, 1]$. Moreover, the cdf of R can be computed explicitly as:

$$\mathbb{P}[R \leqslant r] = \frac{1}{2\pi} \int_0^r \int_0^{2\pi} e^{-r'^2/2} r' \, dr' \, d\theta = \int_0^r e^{-r'^2/2} r' \, dr' = 1 - e^{-r^2/2}.$$

The function $F(r) = 1 - e^{-r^2/2}$, $r > 0$ can be easily inverted. Indeed, we can use the inverse transform method to generate samples of R as follows

$$R = \sqrt{-2\log(U_2)}, \quad U_2 \sim U[0, 1].$$

Thus we can produce two independent standard normals X and Y as follows:

$$X = \sqrt{-2\log(U_2)} \cos(2\pi U_1), \quad Y = \sqrt{-2\log(U_2)} \sin(2\pi U_1), \tag{2.3}$$

where U_1 and U_2 are iid $U[0, 1]$ rvs. The actual verification of this fact is left as an easy exercise.

Exercise 2.6. *Show that the coupled random variables (X, Y) as defined in (2.3) have the correct distribution.*

Proof. First we note that if

$$x = \sqrt{-2\log(u_2)} \cos(2\pi u_1), \quad y = \sqrt{-2\log(u_2)} \sin(2\pi u_1),$$

then

$$u_1 = e^{-(x^2+y^2)/2} \quad \text{and } u_2 = \frac{1}{2\pi} \tan^{-1}(y/x).$$

Denote by $f_{X,Y}(x, y)$ the joint density of the random vector (X, Y),

$$f_{X,Y}(x, y) = \frac{1}{2\pi} \left| \frac{\partial(u_1, u_2)}{\partial(x, y)} \right|,$$

so that

$$f_{X,Y}(x, y) = \frac{1}{2\pi} e^{-(x^2+y^2)/2}$$

□

Once we can generate $\mathcal{N}(0,1)$–randomly distributed rvs, it is straightfoward to generate $\mathcal{N}(\mu,\sigma^2)$–distributed numbers by applying the transformation

$$\mu + \sigma Z, \quad Z \sim \mathcal{N}(0,1).$$

(Exercise: Prove this!) While the Box-Muller algorithm is a perfectly good method to sample from a Gaussian distribution, in practice it is quite slow, due to the fact that we have to compute $\sin$, $\cos$ and $\log$ functions Most software libraries provide access to highly-optimized routines for generating Gaussian distributed random numbers. Perhaps surprisingly, a common method for generating Gaussians is using the inversion method! Even though, the CDF of a Gaussian doesn't have a closed form inverse, one can approximate it using a high-order polynomial (typically quintic) and inverse that instead. Another method used nowadays is the *Ziggurat algorithm*, which is a class of rejection sampling. This algorithm is beyond the scope of the course, and the curious reader is invited to consult the paper *The Ziggurat Method for Generating Random Variables* by Marsaglia and Tsang.

2.2.4 Multivariate Gaussian Distributions

Let us recall the definition of a Multivariate Gaussian distribution. Let $m \in \mathbb{R}^n$ and $\Sigma \in \mathbb{R}^{n\times n}$ be symmetric and positive definite. The random variable $X : \Omega \mapsto \mathbb{R}^n$ with pdf

$$\gamma_{\Sigma,m}(x) := \left((2\pi)^n \mathrm{d}et\Sigma\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left\langle \Sigma^{-1}(x-m),(x-m)\right\rangle\right) \tag{2.4}$$

is termed a **multivariate Gaussian** or **normal** random variable. The mean is

$$\mathbb{E}(X) = m,$$

and the covariance matrix is

$$\mathbb{E}\Big((X-m)\otimes(X-m)\Big) = \Sigma,$$

where $\langle\cdot,\cdot\rangle$ denotes the standard Euclidean inner product, and $(A\otimes B)_{i,j} = A_iB_j$, for $i,j \in \{1,\dots,n\}$. Since the mean and variance specify completely a Gaussian random variable on $\mathbb{R}^n$, the Gaussian is commonly denoted by $\mathcal{N}(m,\Sigma)$.

As in the univariate case, we can obtain a random variable $\mathbf{X}$ with distribution $\mathcal{N}(m,\Sigma)$ from a $\mathcal{N}(0,I)$–distributed rv $\mathbf{Y}$ via a transformation.

Lemma 2.4. *Let* $\mathbf{Y} \sim \mathcal{N}(0, I_{n\times n})$*, i.e.*

$$Y_1,\dots,Y_n \sim \mathcal{N}(0,1) \quad iid.$$

Let C *be a real matrix such that* $CC^\top = \Sigma$ *and define*

$$\mathbf{X} = m + C\mathbf{Y},$$

them $\mathbf{X} \sim \mathcal{N}(m,\Sigma)$.

Proof. The density of $\mathbf{Y}$ is given by

$$f_Y(y) = \frac{1}{(2\pi)^{n/2}} \exp\left(-\frac{1}{2}\langle y, y\rangle\right).$$

The joint density of the new variables is

$$f_X(x) = f_Y(y(x))\,|J(x)|\,,$$

where $J(x)$ denotes the Jacobian matrix of partial derivatives $(\partial y_i/\partial x_j)_{i,j}$ and where, in coordinate form

$$x_i = m_i + \sum_{j=1}^{n} C_{ij} y_j, \quad i = 1, \ldots, n.$$

Then $J(x) = C^{-1}$, and

$$|CC^\top| = \text{det}(CC^\top) = \text{det}(C)\text{det}(C^\top) = \text{det}(\Sigma),$$

thus $|C^{-1}| = |\Sigma|^{-1/2}$. Moreover,

$$\langle y, y\rangle = \langle (x-m), \Sigma^{-1}(x-m)\rangle.$$

The result then follows from the definition of $\mathcal{N}(m, \Sigma)$ given by (2.4). □

As a particular example, consider the two dimensional case. Suppose we wish to generate a sample of a pair of standard Gaussian random variables with correlation ρ, i.e. we want

$$(X, Y) \sim \mathcal{N}(0, \Sigma),$$

where

$$\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix},$$

It is straightforward to check that for

$$C = \begin{pmatrix} 1 & 0 \\ \rho & \sqrt{1-\rho^2} \end{pmatrix},$$

we have $CC^\top = \Sigma$, so that we can simulate $(X, Y) \sim \mathcal{N}(0, \Sigma)$ by generating $(X_1, X_2) \sim \mathcal{N}(0, I)$ and then setting $(X, Y)^\top = C(X_1, X_2)^\top$, i.e.

$$X = X_1, \quad Y = \rho X_1 + \sqrt{1-\rho^2} X_2.$$

Exercise 2.7. *Suppose we wish to sample a pair of Gaussian random variables X_1, X_2 having means μ_i, variances σ_i^2 and correlation ρ. By assuming that the Cholesky decomposition of the covariance matrix is of the form*

$$C = \begin{pmatrix} a_{11} & 0 \\ a_{21} & a_{22} \end{pmatrix},$$

find expressions for a_{11}, a_{21} and a_{22} and solve them to generate samples from X_1, X_2.

More generally, a natural choice for C is the matrix square root of Σ, i.e. the nonnegative (symmetric) square root of Σ. One possible way of computing this is via diagonalisation, i.e. we diagonalise Σ

$$\Sigma = BDB^\top,$$

for an orthogonal matrix B and where $D = \text{diag}\,(\lambda_1, \ldots, \lambda_n)$. Note that since Σ is symmetric and positive definite, then $\lambda_i \geqslant 0$, for $i = 1, \ldots, n$. Then we can write the square root of C as

$$C = B\sqrt{D}B^\top,$$

where $D = \text{diag}\left(\lambda_1^{1/2}, \ldots, \lambda_n^{1/2}\right)$. An alternative computationally simpler approach is to use the *Cholesky decomposition*. Given a symmetric, positive definite matrix Σ, this algorithm produces a lower triangular matrix L such that

$$LL^\top = \Sigma.$$

While the Cholesky decomposition is convenient and many efficient implementations exist, the computational cost of the factorisation is $O(n^3)$, which is prohibitive when n is large. On the other hand, when computing large stream of iid samples from $\mathcal{N}(m, \Sigma)$ it is only necessary to compute the Cholesky decomposition once at the beginning.

2.3 Monte Carlo Simulation

As we described in the introduction, given a random variable X with density p, our objective is to estimate expectations of the form

$$I = \mathbb{E}\left[f(X)\right] = \int f(x)p(x)\,dx.$$

The Monte Carlo approach assumes we can produce a sequence $x_1, x_2, \ldots,$ of independent samples with distribution p, we then approximate I using

$$\hat{I}_n := \frac{1}{n}\sum_{i=1}^{n} f(x_i).$$

Example 2.4. *As a toy example, we shall approximate π using Monte Carlo Methods. Consider a 2×2 square $B \subset \mathbb{R}^2$ with an inscribed circle C of radius 1, see Figure 2.2. Clearly*

$$\frac{\pi}{4} = \frac{\int\int_C dx_1 dx_2}{\int\int_B dx_1 dx_2} = \mathbb{E}\left[\mathbf{1}_C(X)\right],$$

where X is uniformly distributed on B, and

$$\mathbf{1}_C(x) = \begin{cases} 1 & \textit{if } x \in C, \\ 0 & \textit{otherwise} \end{cases}.$$

To sample uniformly on B, we generate $u_1, u_2 \sim U[0,1]$ and then use $X = (x_1, x_2)$ where

$$x_1 = 2u_1 - 1 \quad \textit{and} \quad x_2 = 2u_2 - 1. \tag{2.5}$$

This results in a sequence $\{X_i\}_{i \in \mathbb{N}}$ of samples. The Monte Carlo estimator to compute $\mathbb{E}\left[\mathbf{1}_C(X)\right]$ is then given by

$$\hat{I}_n = \frac{1}{n} \sum_{k=1}^{n} \mathbf{1}_C(X_k).$$

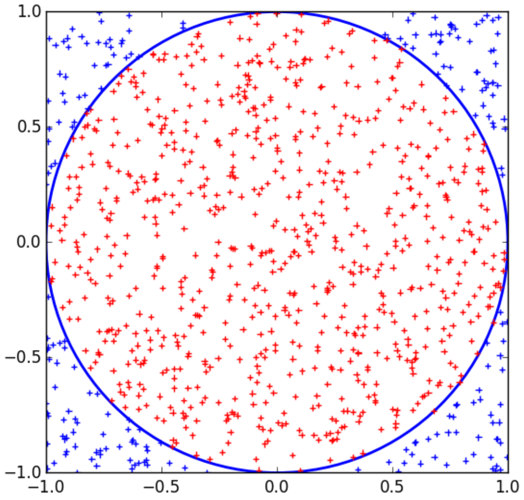


Figure 2.2: Monte Carlo Method to approximate π

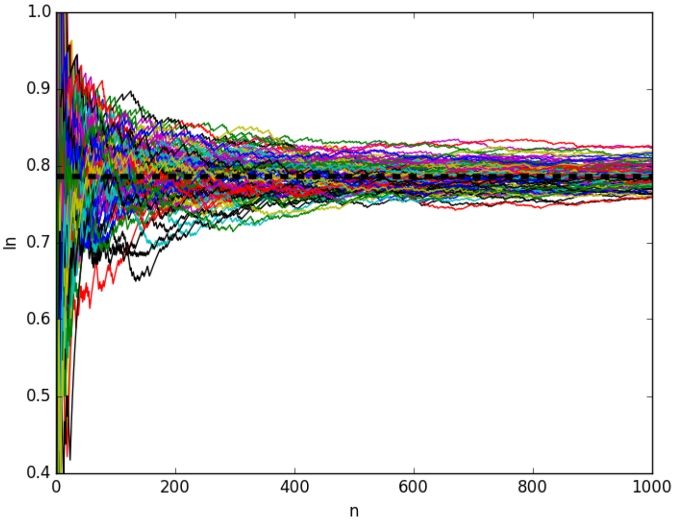


Figure 2.3: Plot of the family of estimators $\hat{I}_n$ over $n = 1, \ldots N$.

Exercise 2.8. *Show that one can instead use $\mathbb{E}\left[f(U_1, U_2)\right]$, where $U_1, U_2 \sim U(0,1)$ and*

$$f(u_1, u_2) = \mathbf{1}\{u_1^2 + u_2^2 \leqslant 1\},$$

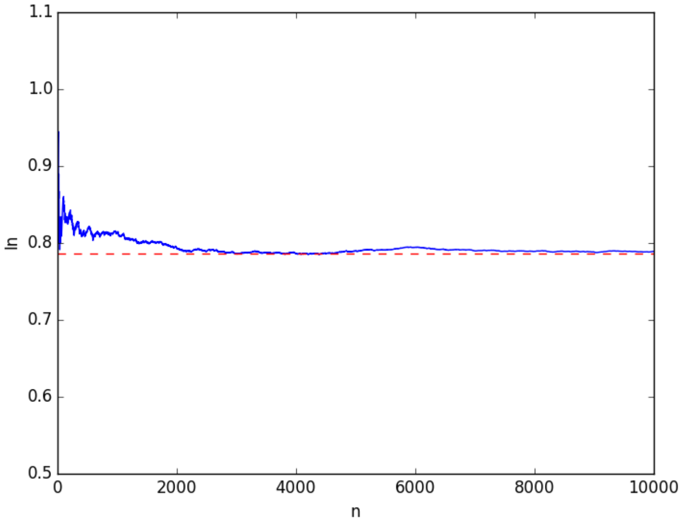


Figure 2.4: Plot of a single Monte Carlo estimator $\hat{I}_n$ over larger numbers of steps.

to approximate $\frac{\pi}{4}$.

The quantity $\hat{I}_n$ is known as an *estimator*, i.e. a quantity obtained from a sequence of observed data used to approximate a quantity of interest. In our case, the observed data is the set of samples $x_1, \ldots, x_n$, while the quantity of interest is $\mathbb{E}\left[f(X)\right]$. Before we study the properties of $\hat{I}_n$ let us recall the following important limit theorems for iid sequences of random variables.

Theorem 2.5 (Strong Law of Large Numbers). *Let $\{Z_i\}_{i\in\mathbb{N}}$ be a sequence of iid integrable random variables with $\mathbb{E}(Z_i) = \mu$ and consider*

$$S_n := \frac{1}{n}\sum_{i=1}^{n} Z_i.$$

Then S_n converges to μ almost surely, that is,

$$\mathbb{P}\left(\lim_{n\to\infty} S_n = \mu\right) = 1.$$

The strong law of large numbers provides us with information about the behavior of a sum of random variables (or a large number or repetitions of the same experiment) on average. While S_n converges almost surely to the expected value, this estimate will possess fluctuations around the average value. The *central limit theorem* allows us to quantify the fluctuations of this finite average around the mean. For the purposes of this module, we only state it in the one–dimensional case.

Theorem 2.6 (Central Limit Theorem). *Let $\{Z_i\}_{i\in\mathbb{N}}$ be a sequence of iid, square integrable*[1] *random variables with $\mathbb{E}(Z_i) = \mu$ and $Var(Z_i) = \sigma^2$. Then*

$$\sqrt{n}(S_n - \mu) \xrightarrow{d} \mathcal{N}(0, \sigma^2).$$

[1] a rv Z is square integrable if $\mathbb{E}|Z|^2 < \infty$

i.e.

$$\lim_{n\to\infty} \mathbb{P}\left(a < \frac{\sum_{i=1}^n Z_i - n\mu}{\sigma\sqrt{n}} < b\right) = \frac{1}{\sqrt{2\pi}} \int_a^b e^{-\frac{x^2}{2}}\, dx, \quad \forall a \leqslant b \in \mathbb{R}.$$

Both the law of large numbers and the central limit theorem have been studied in great generality, and the assumptions of the random variables being iid can be relaxed considerably.

Given the above two results we can now study the properties of the estimator $\hat{I}_n$ in more detail. First we make the following definitions:

Definition 2.1. *A family of estimators* $(\hat{\theta}_n)_{n\in\mathbb{N}}$ *for* θ *is said to be*

1. unbiased *if*
$$\mathbb{E}[\hat{\theta}_n] = \theta, \quad \forall n \in \mathbb{N};$$
2. asymptotically unbiased *if*
$$\mathbb{E}[\hat{\theta}_n] \to \theta, \quad \textit{as } n \to \infty;$$
3. weakly consistent *if*
$$\lim_{n\to\infty} \hat{\theta}_n = \theta, \quad \textit{in probability},$$
i.e. for all $a > 0$
$$\mathbb{P}\left[\left|\hat{\theta}_n - \theta\right| > a\right] \to 0, \ \textit{as } n \to \infty.$$
4. *is* strongly consistent *if*
$$\lim_{n\to\infty} \hat{\theta}_n = \theta, \quad \textit{almost surely};$$
5. *and is* asymptotically normal *if*
$$\sqrt{n}\left(\hat{\theta}_n - \theta\right) \xrightarrow{d} \mathcal{N}(0, \sigma^2), \quad \textit{as } n \to \infty.$$

Clearly, if $\hat{\theta}$ is unbiased then it is asymptotically unbiased, and moreover if $\hat{\theta}$ is strongly consistency then it is also weak consistency. We now demonstrate that all these properties hold for the estimator Monte Carlo estimator $\hat{I}_n$ for I

Proposition 2.7. *Assume that* $I = \mathbb{E}[f(X)]$ *exists. Then* $\hat{I}_n$ *is an unbiased, strongly consistent estimator for* I.

Proof. Clearly

$$\mathbb{E}\left[\hat{I}_n\right] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n f(X_i)\right] = \frac{1}{n}\sum_{i=1}^n \mathbb{E}\left[f(X_i)\right] = \mathbb{E}\left[f(X)\right] = I,$$

so that $\hat{I}_n$ is unbiased. The strong consistency is a consequence of the law of large numbers applied to $Z_i = f(X_i)$, which is applicable as $\mathbb{E}[f(X)]$ is assumed to exist. □

Although it already follows from the previous result, it is quite straightforward to directly prove weak consistency of $\hat{I}_n$, as this is left as an exercise. While consistency guarantees that $\hat{I}_n$ will converge to the correct value as $n \to \infty$, in practice we will only compute $\hat{I}_n$ for some finite (but large) value of n. It is therefore important to have some quantitative measurements of the fluctuations of $\hat{I}_n$ around I for large n. This is one of the benefits of knowing that the estimator is asymptotically normal. We shall prove that $\hat{I}_n$ is asymptotically normal using central limit theorem.

Proposition 2.8. *Assume that $\mathbb{E}\left[f(X)\right]$ and $\sigma^2 = Var\left[f(X)\right]$ exist, then*

$$Var\left[f(X)\right] = \frac{\sigma^2}{n},$$

and

$$\frac{\hat{I}_n - I}{\sigma/\sqrt{n}} \xrightarrow{d} \mathcal{N}(0,1), \qquad \text{as } n \to \infty, \tag{2.6}$$

where $\xrightarrow{d}$ denotes convergence in distribution. In particular, $\hat{I}_n$ is asymptotically normal.

Proof. Using the fact that the X_i are iid:

$$\text{Var}\left[I_n\right] = \sum_{i=1}^{n} \text{Var}\left[\frac{f(X_i)}{n}\right] = \sum_{i=1}^{n} \frac{\text{Var}\left[\hat{I}_n\right]}{n^2} = \frac{\sigma^2}{n}.$$

We now apply the CLT, with $Z_i = f(X_i)$, and $\mu = I$ to obtain (2.6). □

Proposition 2.8 provides us with a means to quantify how good an estimate $\hat{I}_n$ is of I. Applying Chebychev's inequality directly we obtain the bound

$$\mathbb{P}\left[\left|\hat{I}_n - I\right| > a\frac{\sigma}{\sqrt{n}}\right] \leqslant \frac{\text{Var}\left[\hat{I}_n\right]}{a^2\sigma^2/n} = \frac{1}{a^2}.$$

While this bound is rigorous and holds uniformly over n, it is not useful in practice as it is very course. Of course, if one knows some special property about the distribution of X, then one can derive tighter inequalities than those obtained via Chebychev's inequality.

Fortunately, in the large n limit we can obtain a much tighter bound on the error, which holds only in an asymptotic sense (i.e. for n sufficiently large):

$$\frac{\hat{I}_n - I}{\sigma/\sqrt{n}} \approx \mathcal{N}(0,1),$$

which implies that

$$\mathbb{P}\left(\left|\hat{I}_n - I\right| > a\frac{\sigma}{\sqrt{n}}\right) \approx 2\left(1 - \Phi(a)\right),$$

where $\Phi(\cdot)$ denotes the cdf of a standard Gaussian distribution. Thus, for a $(1-\alpha)100\%$ confidence interval for I, we choose $c = c_\alpha$ such that $2(1-\Phi(c_\alpha)) = \alpha$, so that

$$\left(\hat{I}_n - c_\alpha\frac{\sigma}{\sqrt{n}}, \hat{I}_n + c_\alpha\frac{\sigma}{\sqrt{n}}\right), \tag{2.7}$$

is a $(1-\alpha)100\%$ confidence interval for I.

To use (2.7) in practice, we would need to know the value of σ which is the standard deviation of $f(X)$. In general, we will not have a closed form expression for this. Instead, we would make use of an estimator $\hat{\sigma}_n(f)$, given by

$$\hat{\sigma}_n^2(f) = \frac{1}{n-1}\sum_{i=1}^{n}\left(f(X_i) - \hat{I}_n\right)^2,$$

and then use

$$\left(\hat{I}_n - c_\alpha \frac{\hat{\sigma}_n}{\sqrt{n}}, \hat{I}_n + c_\alpha \frac{\hat{\sigma}_n}{\sqrt{n}}\right), \tag{2.8}$$

as approximate $(1-\alpha)100\%$ confidence intervals for I.

Exercise 2.9. *Show that $\hat{\sigma}_n$ is an unbiased estimator for σ^2.*

These confidence intervals give us a means of estimating how many samples x_i we need to generate before the estimator $\hat{I}_n$ is within a specific tolerance of I. The salient point is the following: regardless of the dimension of the state space, i.e. if $\Omega = \mathbb{R}$ or if the state space is $\Omega = \mathbb{R}^{1000}$, the rate of convergence is still $\sigma/\sqrt{n}$, which roughly speaking, means that

$$\text{error} \sim \frac{1}{\sqrt{\text{cost}}}.$$

Based on this apparent independence of the error on dimension, many would claim that Monte Carlo methods beat the curse of dimensionality. There is slightly more to the story however, since σ can depend on dimension, sometimes very badly.

The interval estimator (2.8) provides us with a practical *stopping criterion* to stop an MC simulation (this will be further explored in the code examples). Note that, however, the confidence intervals make use of the central limit theorem, and thus, (2.8) should only considered valid in the large n limit.

2.4 Variance Reduction techniques MC Simulation

From (2.7) is it clear that, for standard MC simulations, the performance (i.e. the number of samples required to approximate I within a given tolerance) depends strongly on σ^2. More generally, a natural way to measure the accuracy of an estimator is via the *mean squared error*, MSE:

$$\text{MSE}(\hat{\theta}_n) = \mathbb{E}\left[(\hat{\theta}_n - \theta)^2\right].$$

In general, the MSE can be decomposed as follows:

$$\begin{aligned}
\mathbb{E}\left[(\hat{\theta}_n - \theta)^2\right] &= \mathbb{E}\left[(\hat{\theta}_n - \mathbb{E}\hat{\theta}_n + \mathbb{E}\hat{\theta}_n - \theta)^2\right] \\
&= \left(\mathbb{E}\hat{\theta}_n - \theta\right)^2 + \mathbb{E}\left(\theta_n - \mathbb{E}\hat{\theta}_n\right)^2 \\
&= B_n^2 + V_n,
\end{aligned}$$

where $B_n = \mathbb{E}\hat{\theta}_n - \theta$ is the bias of the estimator $\hat{\theta}_n$ and $V_n =$ Var$[\hat{\theta}_n]$ is the variance of the estimator. To compute the MSE for the standard MC estimator

$$\hat{I}_n = \frac{1}{n}\sum_{i=1}^{n} f(x_i),$$

for $I = \mathbb{E}[f(X)$, we first note that since $\hat{I}_n$ is unbiased, $B_n = 0$ for all n. Moreover the variance satisfies

$$V_n = \frac{\text{Var}[f(X)]}{n}.$$

Therefore

$$\text{MSE}(\hat{\theta}_n) = \frac{\text{Var}[f(X)]}{n} = \frac{\sigma^2}{n},$$

which is consistent with the error estimator obtained via the CLT in (2.7).

The performance of MC simulation is strongly dependent on the variance of the estimator. In some situations, this can be huge (this will be explored in worksheets), which means that prohibitively many samples would be required to approximate I within a given tolerance. This has motivated the study of *variance reduction methods* which modify the standard estimator $\hat{I}_n$ to reduce the variance (and thus the MSE). We shall discuss a number of standard variance reduction approaches here.

2.4.1 Control Variates

Suppose we wish to compute $I = \mathbb{E}[Z]$. In our case, $Z = f(X)$. suppose we can find a random variable W (known as a *control variate*) with *known* expectation $\mathbb{E}[W]$. Then, for some constant α, let

$$Y = Z + \alpha(W - \mathbb{E}[W]).$$

Clearly, $\mathbb{E}[Y] = \mathbb{E}[Z] + \alpha\mathbb{E}\left(W - \mathbb{E}[W]\right) = \mathbb{E}[Z]$. The variance however, is given by

$$\text{Var}\left[Y\right] = \text{Var}[Z] + \alpha^2\text{Var}[W] + 2\alpha\,\text{Cov}\left[Z, W\right].$$

We want to choose α so that Var$[Y]$ is as small as possible. Clearly, this happens when

$$\alpha = -\frac{\text{Cov[Z, W]}}{\text{Var}[W]},$$

and the minimum variance is then given by

$$\text{Var}[Z] - \frac{\left(\text{Cov}[Z, W]\right)^2}{\text{Var}[W]},$$

which is always less that Var$[Z]$. Thus no matter how we choose W, there will always be a reduction in variance provided we choose the correct value of α.

Suppose we wish to use Monte-Carlo simulation to approximate $\mathbb{E}[f(X)]$ for some rv X. A suitable control variate for Z is then $W = g(X)$ where $\mathbb{E}[W]$ is known, and ideally, f is close to g. When then apply standard MC simulation to estimate

$$I = \mathbb{E}[h(X)], \quad \text{where } h(x) = f(x) + \alpha\left(g(x) - \mathbb{E}[g]\right),$$

where α is chosen as above. The MSE of the modified estimator $\hat{I}_n$ will then be σ_h^2/n, where

$$\sigma_h^2 = \sigma_f^2 - \frac{\text{Cov}[f(X), g(X)]^2}{\text{Var}[g(X)]}.$$

In practice, the optimal constant α is not computable. Instead, we approximate it via an estimator based on empirical values:

$$\hat{\alpha} = \frac{\hat{C}_{f,g}}{\hat{C}_{g,g}},$$

where

$$\hat{C}_{f,g} := \frac{1}{n-1}\sum_{n=1}^{N}\left(f(x_i) - \hat{I}_f\right)\left(g(x_i) - \hat{I}_g\right),$$

and

$$\hat{C}_{g,g} := \frac{1}{n-1}\sum_{i=1}^{n}\left(g(x_i) - \hat{I}_g\right)^2,$$

and where $\hat{I}_f$ and $\hat{I}_g$ are the sample averages:

$$\hat{I}_f := \frac{1}{n}\sum_{i=1}^{n} f(x_i) \quad \text{and } \hat{I}_g := \frac{1}{n}\sum_{i=1}^{n} g(x_i).$$

We can extend the approach to multiple control variates. In this case, we would consider a modified random variable for the form

$$Y = Z + \sum_{i=1}^{M}\alpha_i(W_i - \mathbb{E}[W_i]).$$

One obtains an expression for the optimal α_i in a similar manner.

Example 2.5. *Consider a variance reduction scheme using multiple control variates, namely we construct an unbiased estimator of I by computing the expectation of*

$$Y = Z + \alpha_1(W_1 - \mathbb{E}[W_1]) + \ldots + \alpha_m(W_m - \mathbb{E}[W_m]),$$

where each $\alpha_i \in \mathbb{R}$. Clearly, $\mathbb{E}[Y] = \mathbb{E}[Z] = I$. Our objective is to choose $\alpha_1, \ldots, \alpha_m$ so that the variance of the estimator based on Y is smaller than the variance of that using Z. The variance of Y is given by

$$Var[Y] = Var[Z] + 2\sum_{i=1}^{m}\alpha_i Cov\left[Z, W_i\right] + \sum_{i=1}^{m}\sum_{i=1}^{m}\alpha_i\alpha_j Cov\left(W_i, W_j\right).$$

By taking the derivatives with respect to $\alpha = (\alpha_1, \ldots, \alpha_m)$, it is straightforward that the minimum variance is attained by the solution to the linear equation

$$M\alpha = F,$$

where $M_{i,j} = Cov[W_i, W_j]$ and $F_i = Cov[Z, W_i]$, for $i = 1, \ldots, m$ and $j = 1, \ldots, m$. As before, we will not know the coefficients of the matrix M or the vector F analytically, and one must perform simulation runs to approximate these values. However, a convenient way of estimating the optimal coefficients α, is to observe that $\alpha_{opt} = -\beta_{opt}$, where β_{opt} is the (least-squares) solution to the following linear regression:

$$Z = a + \beta_1 W_1 + \beta_2 W_2 + \ldots + \beta_m W_m + \epsilon,$$

where ϵ is an error term. Many software packages provide commands to automatically output the values of b. For example in MATLAB there is the `regress` command, and `lm` in GNU-R and `glm` in Julia.

2.4.2 Variance Reduction by Conditioning

Once again, suppose we wish to estimate $\mathbb{E}[Z]$ for some random variable Z. Clearly, if $Z_c = \mathbb{E}[Z \mid W]$ for some random variable W, then

$$\mathbb{E}[Z_c] = \mathbb{E}[Z].$$

To compute the variance of the random variable Z_c we use the law of total variance

Lemma 2.9. *Let X and Y be random variables such that variance of Y is finite, then*

$$Var\,[Y] = \mathbb{E}_X\,[Var[Y \mid X]] + Var_X\,(\mathbb{E}\,[Y \mid X])\,.$$

Proof.

$$\begin{aligned}\text{Var}[Y] &= \mathbb{E}[Y^2] - (\mathbb{E}[Y])^2 \\ &= \mathbb{E}\left[\mathbb{E}[Y^2 \mid X]\right] - (\mathbb{E}\,[\mathbb{E}[Y \mid X]])^2\,,\end{aligned}$$

using the law of total expectation. Now

$$\mathbb{E}\left[\mathbb{E}[Y^2 \mid X]\right] = \mathbb{E}\left[\text{Var}[Y \mid X] + (\mathbb{E}\,[Y \mid X])^2\right].$$

and using the fact that

$$\text{Var}\,[\mathbb{E}[Y \mid X]] = \mathbb{E}\,[\mathbb{E}[Y \mid X]^2] - (\mathbb{E}[\mathbb{E}\,[Y \mid X]])^2\,,$$

then the result follows. □

Applying the above lemma with $Y = Z$ and $X = W$ we obtain:

$$\begin{aligned}\text{Var}[Z] &= \text{Var}(\mathbb{E}(Z \mid W)) + \mathbb{E}\,(\text{Var}\,[Z \mid W]) \\ &= \text{Var}(Z_c) + \mathbb{E}\,(\text{Var}\,[Z \mid W]) \geqslant \text{Var}(Z_c),\end{aligned}$$

Thus by conditioning Z with respect to any random variable W we always get a reduction in variance. This motivates the idea of carefully choosing W so that the conditional expectation is a) Computable b) gives a signficant variance reduction.

Example 2.6. *Consider the problem of approximating π, via monte carlo integration, using $\mathbb{E}[Z]$, where $Z = f(U_1, U_2)$ with*

$$f(u_1, u_2) = 4\mathbf{1}\{(u_1^2 + u_2^2 < 1\}),$$

and where $U_1, U_2 \sim U(0,1)$. Take

$$Z_c = \mathbb{E}[Z \,|\, U_1] = 4\mathbb{P}\left(U_2^2 < 1 - U_1^2 \,|\, U_1\right) = 4\sqrt{1 - U_1^2}.$$

We'll approximate the reduction of variance using this estimator in the worksheets.

Example 2.7. *Conditioning is particularly useful when we the underlying random model has some natural hierarchical structure. Take this very simple case where*

$$Y \sim Exp(1) \text{ and given } Y = y, X = \mathcal{N}(y, 4).$$

Our objective is to compute the probality of the event $X > 1$, i.e. $p = \mathbb{P}[X > 1]$. Using standard MC simulation, we'd generate an iid sequence $\{u_i\}_{i \geqslant 1}$ with distribution $U(0,1)$ and $\{z_i\}_{igeq1}$ with distribution $\mathcal{N}(0,1)$ and calculate the proportion of times that

$$x_i = 2z_i - \log(u_i) > 1.$$

To apply conditioning we write down p as an integral:

$$\begin{aligned} p &= \int_0^\infty \left(\int_1^\infty \frac{e^{-(x-y)^2/(2\cdot4)}}{2\sqrt{2\pi}} \, dx \right) e^{-y} \, dy \\ &= \int_0^\infty \left(\int_{(1-y)/2}^\infty \frac{e^{-z^2/2}}{\sqrt{2\pi}} \, dz \right) e^{-y} \, dy \\ &= \int_0^\infty (1 - \Phi((1-y)/2)) e^{-y} \, dy, \end{aligned}$$

where $\Phi(\cdot)$ is the cdf of $\mathcal{N}(0,1)$. Therefore

$$\mathbb{P}[X > 1 \,|\, Y = y] = \mathbb{P}\left[Z > \frac{1-y}{2} \right] = 1 - \Phi((1-y)/2).$$

The conditioned Monte Carlo scheme would then be as follows:

1. *Let $\{u_i\}$ be iid $U(0,1)$ sequence, and set $y_i = -\log(u_i)$.*
2. *Compute $\hat{I}_n^c = \frac{1}{n} \sum_{i=1}^n w_i$, where*

$$w_i = 1 - \Phi\left(\frac{1 - y_i}{2} \right), \quad i = 1, \ldots, n.$$

As we shall see in the worksheets, by using conditioning we obtain a reduction in variance by a factor of 10, and by introducing a further trick, we can obtain a variance reduction to the order of 100.

2.4.3 Importance Sampling

Suppose we want to compute $I = \mathbb{E}\left[f(X)\right]$ where f is an observable which is nearly zero outside a region A, such that $\mathbb{P}(X \in A)$ is small. It can be that the region A has small volume, or it may be that A lies in the tail of the distribution of X. Either way, using a "vanilla" Monte Carlo simulation to generate samples of X will only rarely produce samples in the set A, and an exhorbitant number of samples must be generated before the estimator $\hat{I}_n$ starts to approximate I by a reasonable tolerance.

Intuitively if we can somehow reweight the samples we generate in such a way that samples are generated more frequently within A, while still computing the corrected expected value I, then we would be able to drastically improve performance. This is the main idea behind *importance sampling*: we modify our distribution to oversample from the "important" region A, and then we somehow adjust their relative contribution in the sample average $n^{-1}\sum_{i=1}^{n} f(X_i)$ so as to obtain the correct expectation. Importance sampling can bring enourmous gains, making an otherwise infeasible problem amenable to standard Monte Carlo. On the other hand, when used poorly it can yield estimators which have infinite variance, whereas a vanilla Monte Carlo scheme would have had finite variance.

We should note that importance sampling is more than just a variance reduction method, it is a new sampling scheme in its own right, as it allows us to generate samples of one distribution given that we can sample from another.

Basic Importance Sampling

We shall present the idea on $\mathbb{R}$, nothing that it holds similarly for more general domains. Suppose that X has positive density $p(x)$ on $\mathbb{R}^d$, so that $I = \int f(x)p(x)\,dx$. Suppose that q is another probability density function on $\mathbb{R}$, then we can write

$$I = \int f(x)p(x)\,dx = \int \frac{f(x)p(x)}{q(x)} q(x)\,dx = \mathbb{E}\left[f(Y)w(Y)\right],$$

where $w(x) = p(x)/q(x)$ and $Y \sim q(\cdot)$. Our original goal was to compute the expectation of f with respect to X. Instead we compute the expectation of f with respect to Y, and made an adjustment $w(x)$, known as a *likelihood ratio* to compensate for sampling from q instead of p. The distribution q is the *importance distribution* and p is the *nominal distribution.* The importance distribution q doesn't need to be positive everywhere, for the above expectation to be finite, we merely require that $q(x) > 0$ whenever $f(x)p(x) \neq 0$.

The *importance sampling estimator* for $I = \mathbb{E}\left[f(X)\right]$ is

$$\hat{I}_n^{is} = \frac{1}{n}\sum_{i=1}^{n} \frac{f(X_i)p(X_i)}{q(X_i)}, \quad X_i \sim q. \tag{2.9}$$

Note that the estimator needn't have lower variance than the original MC estimator. Indeed, it is possible that $\hat{I}_n$ has finite variance, while $\hat{I}_n^{is}$ has infinite variance! The following result characterises the variance of the IS estimator:

Proposition 2.10. *Suppose that $q(x) > 0$ whenever $f(x)p(x) \neq 0$ and let $\hat{I}_n^{is}$ be given by (2.9). Then $\mathbb{E}[\hat{I}_n^{is}] = I$ and Var$(\hat{I}_n^{is}) = \sigma_q^2/n$ where*

$$\sigma_q^2 = \int \frac{(f(x)p(x))^2}{q(x)}\,dx - I^2 = \int \frac{(f(x)p(x) - Iq(x))^2}{q(x)}\,dx. \tag{2.10}$$

Proof. That $\mathbb{E}[\hat{I}_n^{is}] = I$ follows from the discussion above. The expression for σ_q^2 in (2.10) is left as an easy exercise. □

We note that it is not apriori true that σ_q^2 is finite! Indeed, it is possible that $\sigma^2 < \infty$ while σ_q^2 is infinite, which suggests a poor choice of $q(x)$. Either way, this should be checked for every case. Equation (2.10) illustrates how importance sampling can succeed or fail. The numerator of the second integrand is small when $f(x)p(x) - Iq(x)$ is close to zero, i.e. when $q(x)$ is nearly proportional to $f(x)p(x)$. From the denominator, we see that in regions where $q(x)$ is small, this lack of proportionality is greatly magnified.

So what would be the optimal choice for q?

Proposition 2.11. *The density q^* that minimises the variance σ_q is*

$$q^*(x) = \frac{|f(x)|p(x)}{\int |f(y)|p(y)\,dy},$$

In particular, if $f \geqslant 0$, then $\sigma_q = 0$, so that $q^*(x)$ is a zero variance estimator.

Proof. By the previous lemma, the variance is minimised if and only if

$$\int \frac{f(x)^2 p(x)^2}{q(x)}\,dx,$$

is minimised. We have for any density q:

$$\begin{aligned}\int \frac{f(x)^2 p(x)^2}{q(x)}\,dx &= \int \frac{f(x)^2 p(x)^2}{q(x)^2} q(x)\,dx \\ &= \mathbb{E}_{Y\sim q}\left[\frac{f(x)^2 p(x)^2}{q(x)^2}\right] \\ &\geqslant \left(\mathbb{E}_{Y\sim q}\left[\frac{f(x)^2 p(x)}{q(x)}\right]\right)^2,\end{aligned}$$

by Jensens inequality. Now since

$$\left(\mathbb{E}_{Y\sim q}\left[\frac{f(x)^2 p(x)}{q(x)}\right]\right)^2 = \left(\int |f(x)|p(x)\,dx\right)^2,$$

this implies that for any density q we have

$$\text{Var}[I_n^{is}(q)] \geqslant \frac{1}{n}\left(\left(\int |f(x)|p(x)\,dx\right)^2 - I^2\right).$$

Plugging in $q = q^*$ we see that this inequality is attained for $q = q^*$, so that the result follows.

For $f \geqslant 0$, it follows that $|f(x)| = f(x)$ so that the variance reduces to zero. □

The previous result gives us a zero–variance estimator, at least for non-negative f. Of course, this is useless in practice, since we would need to compute I to be able to compute $q^*(x)$, however, it provides insight as to what a good choice for q shuold be. Intuitively, it is best for q to have mass (i.e. peaks in the density) wherever, fp does. In general, choosing such a q requires experience and/or numerical experiment.

Expontential Tilting

A common way of generating an importance distribution q from the original density p is to use the moment generating function (MGF) of p (assuming it is finite). Denote by $M_p(t)$ the MGF of p given by

$$M_p(t) = \mathbb{E}\left[e^{tX}\right], \quad X \sim p.$$

We consider the *tilted* density of p given by

$$q(x) = \frac{p(x)e^{tx}}{M_p(t)},$$

for $-\infty < t < \infty$. Here we assume a priori that it exists. If we want to sample more often from a region where X is typically large, we might want to use a tilted density with $t > 0$ as a candidate for q. Similarly if we want to sample more often from the region where X tends to be small, then we can use a tilted density with $t < 0$.

Example 2.8. *Suppose that X is an exponential random variable with mean $1/\lambda$. Then $p(x) = \lambda e^{-\lambda x}$ for $x \geqslant 0$ and it is easy to see that the corresponding tilted distribution is given by $p_t(x) = Ce^{-(\lambda - t)x}$, where C is the normalising constant.*

As an example, let $X \sim \mathcal{N}(\mu, \sigma^2)$ be a normal random variable, and suppose we wish to estimate $I = \mathbb{P}(X > x_0)$ for some x_0 which is large. We apply exponentially tilting, to tilt the pdf of X towards larger values so that we are able to obtain some samples within the desired region, and then use the importance weights to correct for our tilting. If X has pdf $p(x)$, then let

$$q(x) = \frac{p(x)e^{tx}}{M_p(t)},$$

If $p(x)$ is the normal density, then by completing the square:

$$q(x) \propto e^{-(x-\mu)^2/2\sigma^2} e^{tx} = e^{-(x-\mu-t\sigma^2)^2/2\sigma^2} e^{\mu t + t^2\sigma^2/2},$$

so that

$$q(x) = \mathcal{N}(\mu + t\sigma^2, \sigma^2), \quad M_p = e^{\mu t + t^2\sigma^2/2}.$$

Quite fortunately we are able to generate samples from the tilted distribution, since it is once again a Gaussian distribution (side note: this is not a coincidence, and will hold true for any distribution $p(x)$ within the exponential family). The importance sampling weight function is $w(x) = p(x)/q(x) = e^{-tx}M_p(t)$ so that

$$w(x) = e^{-t(x-\mu-t\sigma^2/2)}.$$

The importance sampling scheme using this tilted distribution is thus as follows:

Exponentially Tilted Importance Sampler for I

1. Generate samples $y_i \sim \mathcal{N}(\mu + t\sigma^2, \sigma^2)$, for $i = 1, \dots, n$.
2. Compute $w_i = e^{-t(y_i - \mu - t\sigma^2/2)}$, for $i = 1, \dots, n$.
3. Compute the estimator

$$\hat{I}_n^i = \frac{1}{n} \sum_{i=1}^{n} w_i \mathbf{1}\left[y_i > x_0\right].$$

We have not specified how to choose t. Heuristically, we would expect that a good choice of t is one such that the mean of the tilted distribution equals x_0, i.e. we would choose $\mu + t\sigma^2 = x_0$. To be a bit more precise, we could attempt to derive an optimate t by minimising the variance of the estimator, i.e. we minimise

$$\int \frac{(f(x)p(x))^2}{q(x)}\, dx - I^2,$$

with respect to t. Therefore, for the case where $f(x) = \mathbf{1}_{[x_0,\infty]}$, the optimal t is one which minimises

$$\int_{x_0}^{\infty} p(x) e^{-t(x-\mu-t\sigma^2/2)}\, dx = M_p(t) \int_{x_0}^{\infty} p(x) e^{-tx}\, dx.$$

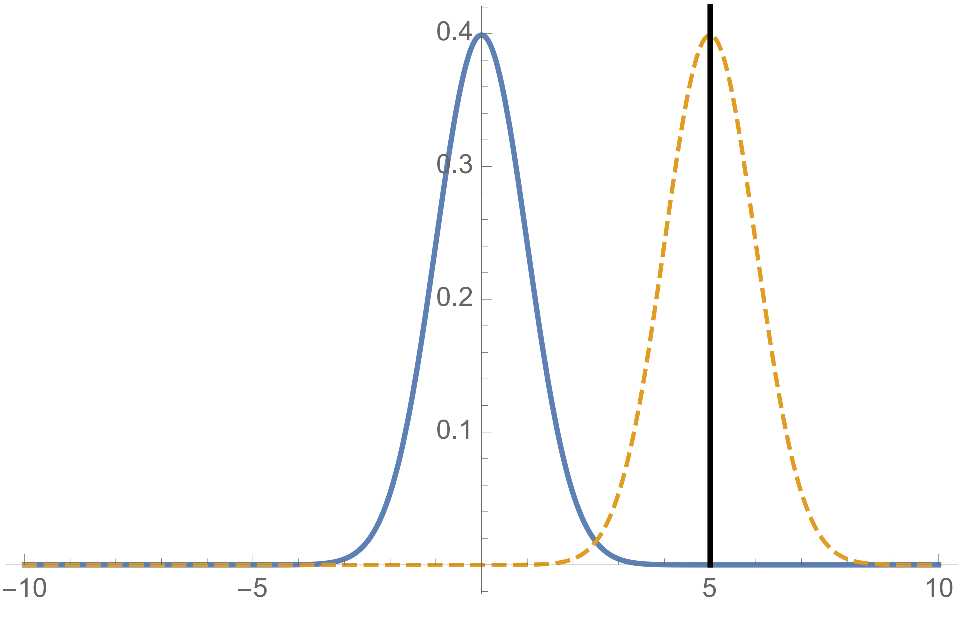


Figure 2.5: Importance sampling for computing $P[X > x_0]$, where $x_0 = 5$ and $X \sim \mathcal{N}(0, 1)$ (pdf in blue). Corresponding tilted potential has mean 0 and variance 1 (pdf dashed)

Sampling from Bimodal distributions (Not examinable)

In many applications we encounter distributions p which are multimodal, possessing well-separated modes. Alternatively, it could be that $f(x)p(x)$ is only non-zero in multiple distinct regions. In this case, a natural choice of q for importance sampling is a mixture distribution

$$q_\alpha = \sum_{j=1}^{J} \alpha_j q_j,$$

where $\alpha_j \geqslant 0$ and $\sum_{j=1}^{J} \alpha_j = 1$ and q_j are distributions. Based on knowledge of p (via exploratory runs), one then seeks a choice of the α_j and q_j which matches the peaks in fp. Mixtures are typically easy to sample from. To sample from q_α we generate a generalised Bernoulli random variable S taking values $i = 1, \dots, J$ with probabilities $\alpha_1, \dots, \alpha_J$, respectively. Then if $J = j$, we return a sample from distribution q_j. The corresponding importance sampling estimator is then given by

$$\hat{I}_n^i = \frac{1}{n} \sum_{i=1}^{n} \frac{f(y_i)p(y_i)}{\sum_{j=1}^{J} \alpha_j q_j(y_i)},$$

where $y_1, \dots, y_n$ are iid samples generated from q_α as described above. An example is shown in Figure 2.6 for a bimodal distribution using a Gaussian mixture $0.5\mathcal{N}(30, 10) + 0.5\mathcal{N}(-30, 10)$ Here we immediately notice an issue! Indeed, while the Gaussian mixture correctly captures the peaks, it will not produce enough samples near the origin, possibly giving rise to a increase in variance due to the correction $w(x)$ for these points. Situations like these can result in the importance sampling performing very poorly, and can arise very easily. One fix is to increase the variance of each component of the Gaussian mixture. In the worksheets, we will study an idea known as *defensive importance sampling.*

Self-Normalising Importance Sampling

In most practical situations we will not be able to compute the normalisation constants for $p(x)$ and $q(x)$, i.e. $\int p(x)\,dx = Z \neq 1$ and $\int q(x)\,dx = \tilde{Z} \neq 1$. Since we need to evaluate $w(x) = p(x)/q(x)$ the previous importance sampler is not applicable in general. However, it is possible to devise an alternative importance sampling estimator based on the observation that:

$$\mathbb{E}_{X\sim p}[f(X)] = \frac{\mathbb{E}_{Y\sim q}[f(Y)w(Y)]}{\mathbb{E}_{Y\sim q}[w(Y)]}, \tag{2.11}$$

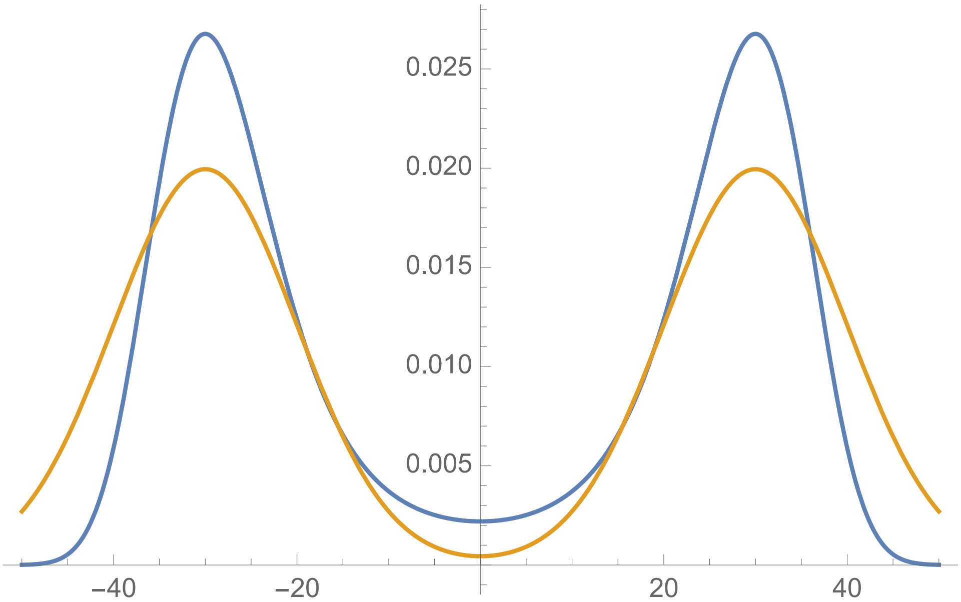


Figure 2.6: Sampling from a bimodal density $p(x)$ (blue) using a Gaussian mixture of $\mathcal{N}(\pm 30, 10)$ with $\alpha_1 = \alpha_2 = 0.5$ (orange). Can you spot the problem with using such a importance density?

where $w(x) = p(x)/q(x)$ is the (unnormalized) likelihood ratio. The proof of this fact is straightforward. Indeed,

$$\begin{aligned}
\mathbb{E}_{X\sim p}\left[f(X)\right] &= \frac{\int f(x)p(x)\,dx}{\int p(x)\,dx} \\
&= \frac{\int f(x)\frac{p(x)}{q(x)}q(x)\,dx}{\int \frac{p(x)}{q(x)}q(x)\,dx} \\
&= \frac{\int f(x)w(x)q(x)\,dx}{\int w(x)q(x)\,dx} \\
&= \frac{\mathbb{E}_{Y\sim q}[f(Y)w(Y)]}{\mathbb{E}_{Y\sim q}[w(Y)]}.
\end{aligned}$$

Note that unlike for the standard importance sampling scheme, it is not sufficient that $p(x)f(x) \neq 0 \Rightarrow q(x) > 0$. In this case, we require the stronger condition that $p(x) > 0 \Rightarrow q(x) > 0$. We can formulate the normalised importance sampling estimator as follows:

Self Normalised Importance Sampler

1. Draw y_i iid samples of q, for $i = 1, \dots, n$.
2. Compute $w_i = p(y_i)/q(y_i)$, for $i = 1, \dots, n$.
3. Generate the estimator
$$\hat{I}_n^n = \frac{\sum_{i=1}^n w_i f(y_i)}{\sum_{i=1}^n w_i}.$$

Note that $\frac{1}{n}\sum_{i=1}^n f(y_i)w(y_i)$ and $\frac{1}{n}\sum_{i=1}^n w(y_i)$ are both unbiased and consistent estimators of $\mathbb{E}_q[f(Y)w(Y)]$ and $\mathbb{E}_q[w(Y)]$, respectively. However, $\hat{I}_n^n$, being a ratio of estimates, will be a *biased* estimator for finite n. However, in the limit as $n \to \infty$, we are guaranteed that $\hat{I}_n^n$ will converge almost surely to the expectation I. Indeed, we have the following result, which we shall state on $\mathbb{R}$, nothing that it readily holds for more general domains.

Proposition 2.12. *Assume that $p(x) > 0 \iff q(x) > 0$, and let $f : \mathbb{R} \to \mathbb{R}$such that $\mathbb{E}_{X\sim p}[f(X)]$ exists. Let $y_1, \dots, y_n$ be iid random variables distributed according to q, then the estimator*
$$\hat{I}_n^n = \frac{\sum_{i=1}^n w_i f(y_i)}{\sum_{i=1}^n w_i},$$
is strongly consistent.

Proof. **[The proof of this result is not examinable]** There exist two sets A and B in the sigma algebra $\mathcal{F}$ such that
$$A = \left\{\omega \in \Omega \, : \, \frac{1}{n}\sum_{i=1}^n f(y_i)w(y_i) \to \mathbb{E}[f(Y)w(Y)]\right\}, \quad \text{and } \mathbb{P}(A) = 1$$
and
$$B = \left\{\omega \in \Omega \, : \, \frac{1}{n}\sum_{i=1}^n w(y_i) \to \mathbb{E}[w(Y)]\right\}, \quad \text{and } \mathbb{P}(B) = 1$$
Then $A \cap B$ has measure 1, and so, there exists a set C of measure 1 such that
$$C = \left\{\omega \in \Omega \, : \, \frac{\sum_{i=1}^n f(y_i)w(y_i)}{\sum_{i=1}^n w(y_i)} \to I\right\},$$
so that $\hat{I}_n^n \to I$, holds $\mathbb{P}$-a.s. □

The self–normalised importance sample estimator can only be considered reliable when the weights are not too varied. In extreme cases, one of the w_i may be much larger than all the others, so effectively we only have one sample. A natural way of identifying this collapse if via the *effective sample size*:
$$ESS = \frac{(\sum_{i=1}^n w_i)}{\sum_{i=1}^n w_i^2} = \frac{\overline{w}^2}{\overline{w^2}},$$

where $\overline{w} = \frac{1}{n}\sum_{i=1}^{n} w_i$ and $\overline{w^2} = \sum_{i=1}^{n} w_i^2$. The effective sample size gives the average number of samples that would have been required had naive MC sampling been used instead. If the weights are unbalanced, this this will be small, which implies that the result is similar to perfoming standard MC over ESS $\times$ n samples.

Chapter 3

Markov Chains and Markov-Chain Monte-Carlo

One of the most complete accounts on Markov chains can be found in the book [11]. For Markov Chains on discrete state spaces, one can consult [1]. In the beginning of this section we shall formulate all our ideas on discrete state space, noting that in many cases the concept extends naturally to continuous state spaces. First we recall some importance definitions. In the following, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space.

Definition 3.1 (Markov Chain)**.** *A discrete-time stochastic process* $\{X_n\}_{n\in\mathbb{N}}$*,* $X_n : (\Omega, \mathcal{F}, \mathbb{P}) \to (S, \mathcal{S})$ *on a state space* $\mathcal{S}$ *is a Markov chain if*

$$\mathbb{P}(X_{n+1} \in B \,|X_0, X_1, \ldots, X_n) = \mathbb{P}(X_{n+1} \in B \,|\, X_n), \quad \forall B \in \mathcal{S}, n \in \mathbb{N}.$$

If the state space $\mathcal{S}$ is discrete we assume that $\mathcal{S}$ is the σ-algebra of all subsets of $\mathcal{S}$. If the chain is started at x, we shall use the notation

$$\mathbb{P}_x(X_n \in B) := \mathbb{P}(X_n \in B \,|\, X_0 = x).$$

The finite dimensional distributions of a Markov chain with initial distribution μ can be expressed through the relation

$$\begin{aligned} &\mathbb{P}_\mu(X_0 \in B_0, X_1 \in B_1, \ldots, X_n \in B_n) = \\ &= \int_{B_0} \mu(dy_0) \int_{B_1} \mathbb{P}(X_1 \in dy_1 \,|\, X_0 = y_0) \ldots, \int_{B_n} \mathbb{P}(X_n \in dy_n \,|X_{n-1} = y_{n-1}). \end{aligned} \tag{3.1}$$

Our main interest will be a particular class of Markov chain, namely time-homogenous markov chains:

Definition 3.2 (Time–homogeneous markov chain)**.** *A markov chain* $\{X_n\}$ *is* time-homogeneous *if*

$$\mathbb{P}(X_{n+1} \in B \,|\, X_n) = \mathbb{P}(X_1 \in B \,|X_0), \quad \forall n \geqslant 0.$$

The Markov property and time-homogeniety imply that we can write

$$\mathbb{P}(X_{n+1} \in B \,|\, X_n = x) =: p(x, B),$$

for some function $p : \mathcal{S} \times \mathcal{S} \to [0,1]$ called the *transition function*. For fixed $x \in \mathbb{S}$, $p(x,\cdot)$ is a probability measure, and $p(\cdot, B)$ is a measurable map. Therefore we rewrite (3.1) as

$$\mathbb{P}_\mu(X_0 \in B_0, X_1 \in B_1, \dots, X_n \in B_n) = \int_{B_0} \mu(dy_0) \int_{B_1} p(y_0, dy_1) \dots \int_{B_n} p(y_{n-1}, dy_n).$$

When the state space S is discrete (finite or countable), we can introduce a *transition matrix* $p = \{p(x,y), x, y \in S\}$, where

$$\sum_{y \in S} p(x,y) = 1, \qquad \text{and } p(x,y) \geqslant 0, \quad \forall x, y \in S.$$

Clearly

$$p(x,y) := \mathbb{P}(X_1 = y \,|X_0 = x) = \mathbb{P}_x(X_1 = y).$$

Clearly

$$\mathbb{P}_\mu(X_0 = x_0, X_1 = x_1, \dots, X_n = x_n) = \mu(x_0)p(x_0, x_1) \dots, p(x_{n-1}, x_n).$$

For all $n \geqslant 1$, we denote $p^n(x,y) = \mathbb{P}_x(X_n = y)$.

The next theorem is an extremely importance consequence of Markovianity.

Theorem 3.1 (Chapman-Kolmogorov Equation). *Let X_n be a time-homogeneous Markov chain with discrete state space. Then for any $m, n \geqslant 0$,*

$$\mathbb{P}_x(X_{n+m} = y) = \sum_{z \in S} \mathbb{P}_x(X_n = z)\mathbb{P}_z(X_m = y).$$

Example 3.1 (Finite state Markov Chain). *Consider the Markov chain on a state space having finitely many states. Then we can express the transition probabilities as a* transition matrix *$P = (p(x,y))_{x,y \in S}$. For example, the following matrix defines a Markov chain on a finite state space taking* 5 *values.*

$$P = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0 & 0.5 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{3.2}$$

The condition that $\sum_{y \in S} p(x,y) = 1$ translates to $P.\mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ denotes the all-ones vector.

Example 3.2 (Random Walk on the Integers). *Let $\xi_1, \xi_2, \dots$ be iid random variables taking values in $\mathbb{Z}$ with $\Gamma(j) = \mathbb{P}(\xi_n = j)$. The random walk Ψ_n is defined as $\Psi_n = \Psi_{n-1} + \xi_n$, $n \geqslant 1$. Let us calculate the transition probabilities of the chain:*

$$\begin{aligned} \mathbb{P}(\Psi_1 = y \,|\, \Psi_0 = x) &= \mathbb{P}(\Psi_0 + \xi_1 = y |\, \Psi_0 = x) \\ &= \mathbb{P}(x + \xi_1 = y) = \Gamma(y - x). \end{aligned}$$

Therefore, $p(x,y) = \Gamma(x - y)$. Notice that the transition probability of going from x to y depends only on the increment $x - y$ and not on the particular values of x and y.

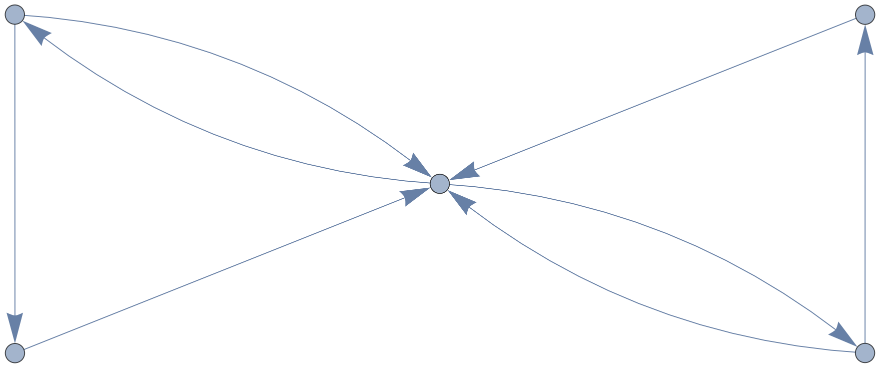


Figure 3.1: Graph structure corresponding to the Markov chain with transition matrix (3.2)

Example 3.3 (Ehrenfest chain)**.** *A box contains N air molecules. The box is divided into two chambers, that communicate through a small hole. The state of the system is determined once we know the number k of molecules contained in the left chamber at each moment in time. Assuming only one molecule per timestep can go through the hole, at time $n+1$ either one molecule has gone from left to right (so that state goes from $k \to k-1$), or one molecule from right to left (so that $k \to k+1$). The transition probabilities of the chain on state space $S = \{0, 1, \ldots, N\}$, are given by*

$$p(k, k-1) = k/N \quad \text{and} \quad p(k, k+1) = (N-k)/N,$$

for all $k \geqslant 0$, with $p(j,k) = 0$ otherwise.

3.0.1 Stationary Processes

We say that a Markov chain X_n is *stationary* if, for every $n \in \mathbb{N}$, the joint distribution

$$\mathbb{P}(X_k \in B_0, X_{k+1} \in B_1, X_{k+2} \in B_2, \ldots, X_{k+n} \in B_n)$$

is independent of the offset $k > 0$. In particular, from the $n = 1$ case

$$\mathbb{E}[X_i] = \mathbb{E}[X_0] =: I, \quad \forall i \in \mathbb{N}$$

and for $n = 2$,

$$\text{Var}[X_i] = \text{Var}[X_0] =: \sigma^2, \quad \forall i \in \mathbb{N},$$

and

$$\text{Cov}[X_i, X_j] = C(j-i), \quad \forall i, j \in \mathbb{N}.$$

In the previous section we considered stochastic processes of the form $\{X_n\}_{n \geqslant 0}$ where each rv X_n was identically distributed with distribution π, but also independent. For stationary processes we relax the assumption of independence, allowing correlation between values of the chains at different times.

As in the previous section, given a stationary sequence Z_n, we can construct time-averages of the form

$$I_n = \frac{1}{n}\sum_{i=1}^{n} Z_i.$$

Then $\mathbb{E}[I_n] = I =: \mathbb{E}[X_0]$, for all $n \geqslant 1$. Therefore, if we could generate realisations of the chain Z_i, then I_n would be an unbiased estimator for I. What is the variance of I_n?

$$\begin{aligned}\text{Var}\left[\sum_i Z_i\right] &= \sum_{i=1}^{n}\sum_{j=1}^{n}\text{Cov}(Z_i, Z_k) \\ &= \sum_{i=1}^{n}\text{Var}(Z_i) + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\text{Cov}(Z_i, Z_j).\end{aligned}$$

Since the process is stationary, then $\text{Var}(Z_n)$ is independent of n, and similarly $\text{Cov}(Z_n, Z_{n+k})$ is independent of n. Thus

$$\begin{aligned}\text{Var}\left[\sum_i Z_i\right] &= n\text{Var}(Z_1) + 2\sum_{k=1}^{n-1}(n-k)\text{Cov}(Z_j, Z_{j+k}) \\ &= n\text{Var}(Z_1) + 2\sum_{k=1}^{n-1}(n-k)C(k)\end{aligned}$$

Therefore

$$n\text{Var}\left[I_n\right] = \sigma^2 + 2\sum_{k=1}^{n-1}\frac{n-k}{n}C(k) \tag{3.3}$$

Taking $n \to \infty$, suppose that the limit

$$\lim_{n\to\infty}\left[\sigma^2 + 2\sum_{k=1}^{n-1}\frac{n-k}{n}C(k)\right] = \sigma^2 + 2\sum_{k=1}^{\infty}C(k), \tag{3.4}$$

exists, then we might hope for a central limit theorem to hold, similar to Theorem 2.6 for the process I_n, with $\sqrt{n}(I_n - I)$ converging in distribution to a Gaussian distribution with mean 0 and variance given by the RHS of (3.4). As we shall see in the following section, this will hold true, provided an additional condition holds.

3.0.2 Stationary Distributions and Ergodicity

Definition 3.3 (Stationary Measure). *A probability measure π on S is a stationary distribution for the Markov chain X_n with transition matrix p if*

$$\sum_x \pi(x)p(x,y) = \pi(y). \tag{3.5}$$

Equivalently, π is a stationary distribution if $X_n \sim \pi$ implies that $X_{n+1} \sim \pi$.

Clearly, if π is a stationary distribution then

$$\sum_x \pi(x)p^n(x,y) = \pi(y), \qquad \text{for all } n \geqslant 1.$$

For a N state Markov chain, a stationary distribution can be expressed as a N–dimensional vector. Then if the chain has transition matrix P, condition (3.5) is equivalent to

$$P^*\pi = \pi.$$

Definition 3.4 (Reversibility)**.** *If a measure π satisfies*

$$\pi(x)p(x,y) = \pi(y)p(y,x), \quad \forall x, y \in S, \tag{3.6}$$

then π is a reversible measure.

The equality (3.6) is also called the *detailed balance condition*, and is a fundamental building block of the Metropolis-Hastings algorithm.

Exercise 3.1. *Identify a stationary distribution of the matrix P given in (3.2), and determine whether the matrix is reversible.*

Theorem 3.2. *If a measure π satisfies the detailed balance condition (3.6) then it is stationary*

Proof. Just sum over x on both sides of (3.6) to get

$$\sum_{x\in S} \pi(x)p(x,y) = \pi(y)\sum_{x\in S} p(y,x) = \pi(y),$$

where the last equality follows from the fact that p is a transition matrix. □

So why do we call chains that satisfy (3.6) reversible? Let X_n be Markov chain which admits a stationary distribution π and suppose that $X_0 \sim \pi$. For fixed $n \in \mathbb{N}$ consider the "time–reversed" process $Y_m = X_{n-m}$. Then for every $n \in \mathbb{N}$, Y_m is a time-homogenous Markov chain with $Y_0 \sim \pi$. The transition probabilities $q(x,y)$ of Y_m are given by:

$$\begin{aligned} q(x,y) &= \mathbb{P}(Y_1 = y \,|\, Y_0 = x) = \mathbb{P}(X_{n-1} = y \,|\, X_n = x) \\ &= \mathbb{P}(X_n = x \,|\, X_{n-1} = y)\frac{\pi(y)}{\pi(x)} = \frac{p(y,x)\pi(y)}{\pi(x)}. \end{aligned}$$

If the DB condition holds, it follows that $q(x,y) = p(x,y)$, for all x, y, and in this case, the chain X_n is said to be *time–reversible*. We now introduce the most important theoretical concept in this section

Definition 3.5 (Ergodicity)**.** *A Markov chain is said to be* ergodic *if it admits a unique stationary probability distribution. In this case, the invariant distribution is said to be the ergodic measure for the chain.*

Roughly speaking, in order for a process to be ergodic it has to

1. Eventually explore the entire space, i.e. for every point x in space, there will exist a n for which X_n is in some sense close to x.
2. Explore the space in a "homogenous way": the measure π controls how frequently the process will explore a given region of space, i.e. if for some set A, $\pi(A)$ is small then X_n will visit A rarely, whereas if $\pi(A)$ is large the process will visit X_n often. Indeed, the physicists' interpretation of ergodicity is "space averages equals time average".
3. The limiting behaviour forgets the initial condition from which it started from.

There are a number of very specific conditions which are sufficient for a given Markov chain to be ergodic. The precise details are beyond the scope of this course, and we shall defer the interested reader to [11]. We shall only mention them briefly. The following conditions[1] are required for a Markov chain to be ergodic, i.e. to possess a unique invariant distribution.

- **Irreducible:** Any set A can be reached from any other set B with nonzero probability.
- **Positive Recurrent:** For any set A, the expected number of steps required for the chain to return to A is finite.
- **Aperiodic:** For any set A, the number of steps required to return to A must not always be a multiple of some value k.

Roughly speaking, positive recurrence ensures the existence of an invariant measure, while irreducibility ensures the uniqueness of the invariant measure. The final question of establishing the convergence of $p^(x,y)$ to $\pi(y)$ for all x, namely

$$\sum_{y \in S} |p^n(x,y) - \pi(y)| \to 0,$$

is guaranteed by aperiodicity. Clearly, $\pi(y)$ represents the probability for the chain to be in state y as $n \to \infty$. It is important that convergence to π happens irrespective of the initial data that we pick, i.e. the initial condition is forgotten in the limit.

In the previous chapter we considered stochastic processes of the form $\{X_n\}_{n \geqslant 0}$, where Z_n were independent random variables distributed according to some common distribution π. The existence of a limit for the sample mean

$$S_n = \frac{1}{n} \sum_{i=0}^{n} Z_i,$$

was guaranteed by the strong law of large numbers. The Markov chains we are now considering do not fall in this class of process, indeed, the random variables X_n will not be identically distribution, and certainly not independent (indeed, we expect correlation between X_n and X_{n+1} in general). Nonetheless, in some sense, we can obtain a similar result, at least when the Markov Chain X_n is ergodic. Indeed, this is how the concept of "space averages equals time averages" is formalized.

[1] these conditions are valid for discrete state spaces, for continous state spaces strong conditions are required

Theorem 3.3 (Ergodic Theorem)**.** *Let X_n be an ergodic Markov chain with unique invariant distribution π. Then for any integrable function, the limit*

$$\frac{1}{n}\sum_{i=0}^{n-1} f(X_i) \to \int f(x)\pi(dx),$$

as $n \to \infty$, for $\mathbb{P}$-almost surely every x.

When $f = \mathbf{1}_A$ for some $A \in \mathcal{F}$, then the above limit says exactly that, asymptotically, space averages equal time averages.

A natural question is whether a corresponding central limit holds to characterise the fluctuations of $n^{-1}\sum_{i=0}^{n-1} f(X_i)$ around the mean $\mathbb{E}[f(X)]$, for $X \sim \pi$. In the case that the chain is reversible, then a central limit was established by Kipnis and Varadhan [7], using a proof, which although is extremely elegant, will not be studied in this course.

Theorem 3.4 (Central Limit Theorem for Stationary, Reversible Markov Chains)**.** *If X_n is an ergodic reversible Markov chain with invariant distribution π, and suppose that $X_0 \sim \pi$, so that X_n is stationary. Then the central limit theorem applies, i.e.*

$$\sqrt{n}\left(\frac{1}{n}\sum_{i=0}^{n-1}(f(X_i) - \mathbb{E}_\pi[f])\right) \xrightarrow{d} \mathcal{N}(0, \sigma^2(f)),$$

provided that

$$0 < \sigma^2(f) = \mathit{Var}[f(X_0)] + 2\sum_{i=1}^{\infty} \mathit{Cov}\left[f(X_0), f(X_k)\right] < \infty. \tag{3.7}$$

If it exists, then $\sigma^2(f)$ is known as the asymptotic variance.

Note that (3.7) is the $n \to \infty$ limit of (3.3) where $Z_i = f(X_i)$.

3.1 Markov Chain Monte Carlo

In the previous section we studied the very basic theory of Markov chains. While we focused on finite or countable state spaces, much of what we said can be readily extended to $\mathbb{R}^d$–valued Markov chains, either directly, or with some small modifications of the required assumptions. Rather than delve into the details of this, we shall simply claim that all the previously stated results hold for general state spaces, referring the interested reader to [11] for justification. In this section we want to look at a major application of theory that we have described so far: Markov Chain Monte Carlo methods (MCMC)

In Chapter 2 we introduced Monte Carlo simulation as a means to compute expectations of the form $\mathbb{E}[f(X)]$, where $X \sim \pi$. The entire method hinges on a single assumption: that it is possible and feasible to generate samples π. In many scenarios this is simply not true, particularly when working with high-dimensional models. Suppose however that we have a Markov chain X_n which is erogdic with unique stationary distribution π. This implies three things:

1. If $X_n \sim \pi$, then $X_{n+k} \sim \pi$, for all $n \geqslant k$.
2. Denoting the distribution of the random variable X_n by $\mathcal{L}(X_n)$, then $\mathcal{L}(X_n) \to \pi$, as $n \to \infty$, regardless of the distribution of X_1.
3. The ergodic theorem implies that

$$\frac{1}{n}\sum_{i=1}^{n} f(X_i) \to \mathbb{E}_{\pi}[f], \text{ as } n \to \infty,$$

for all integrable functions f.

Suppose our objective is to compute

$$I = \mathbb{E}_{\pi}[f] = \int_{\mathbb{R}^d} f(x)\pi(x)\,dx,$$

then if we could somehow construct an ergodic process with unique invariant distribution π then the most natural estimator for I would be to simulate X_i up to time n and use the time-average:

$$\hat{I}_n = \frac{1}{n}\sum_{i=1}^{n} f(X_i),$$

as an estimator for I. Clearly, the iid chains where $X_i \sim \pi$, which we considered in Chapter 2 are a special case of what we are describing. However, we are describing a far more general situation where we generate a sequence of *correlated* random variables from a Markov chain. This idea of specifically constructing a Markov chain which is ergodic with respect to a given target distribution is the underpinning of *Markov Chain Monte Carlo* methods (MCMC). These methods have been around almost as long as standard MC, both of which originate from Los Alamos in the 1940s. It is not a coincidence that the invention of MC and MCMC methods coincide with the development of the first computer, the ENIAC. Indeed, one of the first algorithms to run on the ENIAC was a MC method used by Von Neumann to solve some problems related to fission and themonuclear physics. While MCMC methods were used throughout various areas of physics after the 1940s, their impact on Statistics wasnt really felt until the early 1990s. A nice, short-history of Markov Chain Monte Carlo methods can be found in [14]

The consistency of $\hat{I}_n$ is guaranteed by the ergodic theorem, however one fundamental difference is that:

$$\mathbb{E}[\hat{I}_n] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[f(X_i)] \neq \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{X\sim\pi}[f(X)] = I,$$

i.e. $\hat{I}_n$ is a biased estimator for I. Of course, if we set $X_1 \sim \pi$, then it follows from property (1) that $\mathbb{E}[f(X_i)] = \mathbb{E}_{X\sim\pi}[f(X)]$, so that $\hat{I}_n$ will be unbiased. Of course, this assumes that we are somehow able to generate samples from π, which defeats the purpose of all this. The reason that $\hat{I}_n$ is unbiased is that in the distribution of X_n is very different from π, when n is not large (i.e. in the *transient phase*). Property 3 suggests that we can reduce this bias by introducing a *burn-in phase*, namely we discard the first n_0 samples for some $n_0 > 0$, and instead use the estimator:

$$\hat{I}_{n_0,n} = \frac{1}{n}\sum_{i=n_0}^{n_0+n} f(X_i).$$

As we shall see, constructing a Markov chain X_n which is ergodic with respect to π is possible for a wide range of target distributions. The main challenge faced with MCMC is deciding when to stop. Ideally, we could attempt to exploit any asymptotic normality (i.e. CLT) of $\hat{I}_n$ to construct confidence intervals, but since $\hat{I}_n$ is based on a single realisation of a Markov chain, computing variance estimators is problematic.

We shall address these problems throughout this chapter, but for now, let us focus on an extremely powerful method for constructing Markov chains.

3.2 The Metropolis-Hastings Algorithm

The Metropolis-Hastings (MH) algorithm is one method which can be used to construct Markov chains which are ergodic with respect to a given target distribution. The idea is a natural generalisation of the rejection algorithm which we described in Section 2.2.2. Indeed the MH method is based on an accept-reject step which is required to ensure that the resulting chain exhibits the correct stationary distribution.

Suppose that we are given a target density $\pi(x)$, known only up to a normalisation constant, and we have an associated conditional density $q(\cdot\,|\,x)$ which is easy to sample from, known as the *proposal density*. There are additional theoretical requirements on $\pi(x)$ and $q(y|x)$ which we shall elaborate further on, but first let us describe the algorithm. The Metropolis-Hastings algorithm associated with the target density π and conditional density q is then defined through the following algorithm:

Metropolis-Hastings algorithm

Suppose that we the chain has state X_n at time n. Define

1. Generate $Y \sim q(\cdot\,|\,X_n)$.
2. With probability $\alpha(X_n, Y)$ accept the proposal Y, i.e. set $X_{n+1} = X_n$, where

$$\alpha(x,y) = \min\left\{1, \frac{\pi(y)q(x\,|y)}{\pi(x)q(y\,|\,x)}\right\}.$$

3. Otherwise reject the proposed sample, and set $X_{n+1} = X_n$.

The probability $\alpha(x, y)$ is known as the *Metropolis-Hastings acceptance probability*. As for the rejection sampler, to accept a sample with probability p, we generate $U \sim U(0,1)$ and accept the sample if $U < p$, and reject otherwise. Note that since we have $\pi(y)$ above and below in the acceptance probability, the normalisation constants will cancel out. This is crucial as it allows us to sample from π without having an expression for the normalisation constant.

Suppose that we are working in a discrete state space [2]. The chain generated by the MH algorithm is clearly a time-homogeneous Markov process. What is the transition matrix of the

[2] the argument for continuous state spaces is entirely equivalent

chain X_n generated by the MH algorithm? Suppose that the current state is $X_n = x$, then the probability that $X_{n+1} = y$ where $x \neq y$ is given by

$$p(x, y) = q(y \,|\, x)\alpha(x, y), \quad x \neq y,$$

i.e. from state x the MH algorithm proposes y and it is accepted. On the other hand, if $x = y$, then either the MH algorithm proposes x and it is accepted, *or* some other z is proposed, and it is rejected. We write the correspnding transition matrix as:

$$p(x, x) = q(x \,|\, x)\alpha(x, x) + \sum_{z \in S} (1 - \alpha(x, z))q(z \,|\, x).$$

In a more compact form we can write the transition matrix as

$$p(x, y) = q(y \,|\, x)\alpha(x, y) + \delta_x(y) \sum_{z \in S} (1 - \alpha(x, z))q(z \,|\, x). \tag{3.8}$$

Given this expression we establish that π is a stationary distribution of X_n, by showing that π is reversible with respect to X_n.

Proposition 3.5. *The density π is reversible with respect to the transition density (3.8).*

Proof. First consider the case where $x \neq y$:

$$\begin{aligned}
\pi(x)p(x, y) &= \pi(x)q(y \,|\, x)\alpha(x, y) \\
&= \pi(x)q(y \,|\, x) \min\left(1, \frac{\pi(y)q(x \,|\, y)}{\pi(x)q(y \,|\, x)}\right) \\
&= \min\left(q(y \,|\, x)\pi(x), \pi(y)q(x \,|\, y)\right) \\
&= \min\left(\frac{q(y \,|\, x)\pi(x)}{\pi(y)q(x \,|\, y)}, 1\right) \pi(y)q(x \,|\, y) \\
&= \alpha(y, x)\pi(y)q(x \,|\, y).
\end{aligned}$$

The analogous condition for when $x = y$ then holds immediately. □

It follows directly from Theorem 3.2 that π is an invariant distribution of X_n. To be of practical use we must establish conditions for which X_n is ergodic. The following result establishes fairly general conditions under which this is true

Theorem 3.6. *Assume that π is bounded and positive on every compact set the domain. If there exist positive numbers ϵ and δ such that*

$$q(y \,|\, x) > \epsilon \quad \textit{if} \quad |x - y| < \delta,$$

then the Metropolis-Hastings Markov chain is ergodic with respect to π. In particular, for all integrable functions f:

$$\lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} f(X_i) = \mathbb{E}_\pi[f], \quad \textit{for} \quad a.e. X_1$$

and we have convergence of the distribution of X_n to π in total variation, i.e.

$$\lim_{n\to\infty}\sum_{y\in S}\left|\sum_{x\in S}p^n(x,y)\mu(x)-\pi(y)\right|=0,$$

for every initial distribution $X_1\sim\mu$, where $p^n(x,y)$ denotes the transition matrix for n steps of the MH chain.

The above two conditions are not very stringent and are very easy to verify in general. However, having merely an ergodic chain is not enough for a Markov chain to be useful in practice for sampling. In practice, we would want to quantify the rate of convegence to equilibrium. One "qualitative" convergence rate property is uniform ergodicity, i.e. that

$$\|P^n(x,\cdot)-\pi(\cdot)\|_{TV}\leqslant M\rho^n,\quad n=1,2,3,\ldots, \tag{3.9}$$

for some $\rho<1$ and $M<\infty$, where $P^n(x,\cdot)$ denotes the distribution of X_n at time t, given that $X_1=x$, and the distance $\|\cdot\|_{TV}$ denotes total variation distance[3], i.e.

$$\|\mu-\nu\|_{TV}=\sup_{A\in\mathcal{F}}|\mu(A)-\nu(A)|.$$

A slightly weaker property is *geometric ergodicity*, which holds if

$$\|P^n(x,\cdot)-\pi(\cdot)\|_{TV}\leqslant M(x)\rho^n,\quad n=1,2,3,\ldots,$$

for some $\rho<1$, where $M(x)<\infty$, for π-.a.e. $x\in S$. The difference between geometric ergodicity and uniform ergodicity is that now the constant M may depend on the initial state x. Of course, if the state space S is finite, then all irreducible and aperiodic Markov chains are geometrically (in fact, uniformly) ergodic. However, for infinite S this is not the case. The machinery for demonstrating these results can be found in [11]. What about the chain X_n generated by the Metropolis-Hastings chain? Is it uniformly, geometrically ergodic, or neither? The answer is that it depends on the proposal density and on the tails of the target distribution π.

Before we study some of standard choices of proposals let us consider a simple example to illustrate how the Metropolis–Hastings algorithm works. Suppose we wish to sample from a bimodal distribution with density (up to normalisation constant) is given by

$$\pi(x)=e^{-(x^2-1)^2}.$$

As a proposal density, we shall use a uniform distribution $U[x-r,x+r]$ centered around x with width r, i.e. $q(y\,|\,x)=\frac{1}{2r}\mathbf{1}[x-r,x+r](y)$. Noting that $q(y\,|\,x)=q(x\,|\,y)$, we can write the acceptance probability as:

$$\alpha(x,y)=\min\left(1,\frac{\pi(y)}{\pi(x)}\right).$$

The corresponding MH algorithm for this scheme is then given by

[3] for discrete spaces this is equivalent to $\sum_{y\in S}\left|\sum_{x\in S}p^n(x,y)\mu(x)-\pi(y)\right|\leqslant M\rho^n$, where $X_1\sim\mu$.

```
Given the state $X_n$
    1. Sample $Y \sim U[x-r, x+r]$ and $u \sim U[0,1]$
    2. If $u < \min\left(1, \frac{\pi(Y)}{\pi(X_n)}\right)$, accept $X_{n+1} = Y$
    3. Otherwise $X_{n+1} = X_n$.
```

Therefore, we see that the acceptance rule is straightforward: if $\pi(Y) > \pi(X_n)$ accept the proposed density, otherwise reject with probability $\pi(Y)/\pi(X_n)$. In Figure 3.5 we plot a simulation of the MH scheme for this problem for $N = 10^4$ steps starting from $X_1 = 1.0$. We choose $r = 1.0$. We plot the distribution of the 10^4 samples and compare with the exact density (normalised). We see that after 10^4 the distribution is very close to the target distribution. Since the proposal density can make jumps of size 1, it is relatively easy for the chain to jump from the $x = 1$ mode to the $x = -1$ mode. We also plot the time series of X_n over n, and see that the chain appears to explore the state space quite uniformly. We repeat the results in Figure 3.6 with $r = 0.1$. In this case the proposal density will only generate local moves of length 0.1. Therfore, once the chain is stuck in a given mode, it is relatively hard for it to escape to the other mode. Indeed, from the time series plot we see that the chain only spends about 20% of the time around the $x = -1$ mode, and the estimator $\hat{I}_n$ fails to converge within $N = 10^4$ time steps. Finally, we perform the same experiment with $r = 0.01$ in Figure 3.7. In this case we see the chain is not able to cross modes even once in $N = 10^4$ timesteps, since the proposal density will only take very small steps.

From this example it is clear that the choice of proposal distribution plays a huge role in the behaviour of the chain, and the quality of the estimator $\hat{I}_n$. This motivates us to explore a few possible choices for $q(\cdot \,|\, x)$.

3.2.1 The choice of proposal density

As with the rejection method, there are many possible choices for the conditional proposal density, and the choice of this proposal will affect the performance of the sampler. We now explore three classes of proposals.

The independence Sampler

Although the proposal distribution $q(\cdot \,|\, x)$ is allowed to depend on the current state of the chain, there is no obligation for this to be the case. Indeed we can choose a proposal which is independent of the present state of the chain, that is $q(y \,|\, x) = g(y)$. In this case, the resulting algorithm, known as the *independence sampler* becomes

The independence sampler

Given the state X_n

1. Sample $Y \sim g(y)$.
2. If $u < \min\left(1, \frac{\pi(Y)g(X_n)}{\pi(X_n)g(Y_t)}\right)$, accept $X_{n+1} = Y$
3. Otherwise $X_{n+1} = X_n$.

Note that although the proposal generates independent samples, the resulting chain X_n is not, because the probability of accepting the proposed state Y will depend on X_n. You might ask at this point, why not use the rejection algorithm straight off, with $g(\cdot)$ as the proposal density for the candidate distribution $p(x)$. One clear advantage of the independence sampler is that one doesn't need to know the uppoer bound $M = sup_x \pi(x)/g(x)$ beforehand. Therefore, in cases where we cannot efficiently compute M, or it is so large that the rejection algorithm performs poorly, then one should definitely consider using the independence sampler.

A natural observations is that when $M = \sup_x \pi(x)/g(x) = \infty$, while the rejection algorithm cannot be used, the independence sampler would still work in theory. However, as described in Robert and Casella (2004), the performance of the independence sampler in such cases tends to be very poor. On the other hand, suppose that $M = \sup_x \pi(x)/g(x) < \infty$, then we have the following result:

Theorem 3.7 (Mengersen and Tweedie (1996))**.** *The independence sampler produces a uniformly ergodic chain if there exists a constant such that*

$$\pi(x) \leqslant Mg(x), \quad x \in supp\pi. \tag{3.10}$$

In this case,

$$\|P^n(x, \cdot) - \pi\| \leqslant 2\left(1 - \frac{1}{M}\right)^n.$$

On the other hand, if there exists a set of x with positive measure such that (3.10) does not hold, then X_n is not geometrically ergodic.

Exercise 3.2. *Indeed, let's return to exercise 2.5, sampling a standard Cauchy random variable using a Gaussian proposal, i.e.*

$$p(x) = \frac{1}{\pi(1+x^2)} \quad \text{and} \quad g(x) = \frac{1}{2\pi}e^{-x^2/2}.$$

Implementing this in Julia, the histogram for 10^6 samples is shown in Figure (3.2). From this plot, we observe that the chain fails to capture the tail behaviour of the distribution. Indeed, when in the tails of the distribution the probability of acceptance is extremely small. For example, for $X_n = 10$, the probability of accepting a state is given by

$$\int_{-\infty}^{\infty} \frac{e^{-x^2/2}(1+x^2)}{1+y^2}\, dy \approx 6.12 \times 10^{-20},$$

Thus, the sampler is extremely unlikely to ever accept a state proposed too far away from the origin.

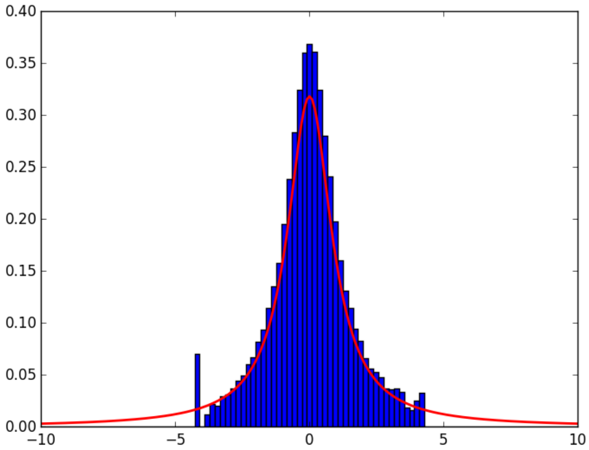


Figure 3.2: Distribution of independence sampler for standard Cauchy target distribution using Gaussian proposals.

Random Walk Metropolis Hastings

While the independence sampler will eventually produce of the target distribution, it is wasteful in this sense that the proposed state is independent from the current. A far more natural approach is to consider a local exploration of the neighbourhood of the current value of the Markov chain. The idea is to generate Y according to:

$$Y = X_n + \zeta,$$

where ζ is a random perturbation with some given distribution g indepedendent of X_n, which is assumed to be *symmetric around 0*, i.e. $g(y) = g(-y)$ for all y. Possible choices would be

1. $g(y)$ is uniform, so that $Y \sim U(X_n - \delta, X_n + \delta)$ item $g(y)$ is Gaussian, so that $Y \sim \mathcal{N}(X_n, \delta^2)$.

Associated with each distribution is a scaling parameter δ which controls the size of the "jump" from the current state X_n.

Note that the proposal density can be expressed as $q(y \,|X_n) = g(y - X_n)$, and the symmetry assumption implies that $q(y \,|X_n) = q(X_n \,|\, y)$. The corresponding MH algorithm, known as the Random-Walk Metropolis Hastings algorithm (RWMH) is given by

Random Walk Metropolis Hastings

Given the state X_n

1. Sample $Y = X_n + \xi$, where $\xi \sim g$.
2. If $u < \min\left(1, \frac{\pi(Y)}{\pi(X_n)}\right)$, accept $X_{n+1} = Y$
3. Otherwise $X_{n+1} = X_n$.

An interpretation of the acceptance probability is that “uphill" proposals (proposals which take the chain closer to a local mode) are always accepted, whereas “downhill” proposals are accepted with probability exactly equal to the relative “heights” of the posterior at the proposed and current values. Although the shape of the distribution g clearly has some effect on the chain’s performance, the most crucial parameter to calibrate is step-size δ. Intuitively, and from the examples we shall see in the worksheets, there is a tradeoff to be made when choosing δ: choosing δ too large, many proposals might be generated in regions where the target density $\pi(x)$ is much smaller, resulting in them being rejected, especially when the current state is close to a mode of π. On the other hand, if we choose δ too small, proposals are likely to be accepted, but the chain will not “explore” the state space too quickly, and it will take longer for the distribution of the chain to converge to equilibrium, and the estimator $\hat{I}_n$ to converge to I. We shall discuss this briefly in later in this chapter.

Although the random walk proposal is a very natural one, the RWMH algorithm does not give rise to a uniformly ergodic chain X_n. Indeed, if $\pi > 0$, then RWMH is never uniformly ergodic. However, it is possible to establish conditions under which the chain is geometrically ergodic, namely the *log-concavity* of π in the tails, i.e. if there exists $\kappa > 0$ and x_1 such that

$$\log \pi(x) - \log \pi(y) \geqslant \kappa |y - x|, \tag{3.11}$$

for $y < x < -x_1$ or $x_1 < x < y$. For positive symmetric densities (3.11) is enough to ensure geometric ergodicity.

Langevin proposals (MALA)

While the random walk proposal is a natural choice of proposal density, it does not make any effective use of the local structure of the target density. For, since the gradient $\nabla \pi(X_n)$ will point towards the local mode of the distribution, it would be natural to somehow bias proposals to prefer this direction, while still allowing a certain amount of randomness to promote exploration. This has motivated the introduction of proposals based on the *overdamped Langevin* SDE[4] :

$$dX_t = \nabla \log \pi(X_t)\, dt + \sqrt{2}\, dW_t,$$

where W_t is a standard Brownian motion. The proposal is given by an Euler-Maruyama discretisation of the proposal, with step size δ, namely

$$X_{n+1} = X_n + \nabla \log \pi(X_n)\delta + \sqrt{2\delta}\xi_n,$$

[4] Don’t worry if you’re unfamiliar with this notation, this will be the main topic of the next chapter

where $\xi_i \sim \mathcal{N}(0, I)$, iid. The magnitude of the random jump is therefore controlled by the stepsize δ. The proposal conditional density is given by

$$q(y|x) \propto \exp\left(-\frac{|y - x - \nabla \log \pi(x)\delta|^2}{4\delta}\right)$$

Note that the proposal density is not symmetric, so we will not obtain the same cancellations in the acceptance probability that we had for the independence sampler and RWMH. The main advantage of the MALA scheme as that it proposes moves into region of high target probability, thus which are more likely to be accepted. This comes at the cost of needing to compute the gradient of the log density. In many applications, this is known exactly, however when it isn't this can be replaced by numerical approximations. The qualitative rate of convergence to equilbrium depends strongly on the tails of the distribution π. Indeed, for a distribution with very light tails, for example if $\pi(x) \propto \exp(-\gamma|x|^\beta)$, for $\beta > 2$ then the corresponding Markov chain is not geometrically ergodic.

3.2.2 Performance and Tuning of Metropolis Hastings

The theoretical results of the previous section provide us with very natural conditions to ensure that the chain X_n converges to stationarity[5]. Under additional conditions, we can even show that the convergence is exponentially fast. Thus, even though $\hat{I}_n$ is a biased estimator of $\mathbb{E}_\pi[f]$, we know that we can mitigate this bias by discarding a sufficiently long burn-in simulation. However, these theoretical guarantees do not tell us when to stop with any confidence. Ideally we would like to have a test which based on a simple run tells us when the bias is sufficiently small, so that the chain has converged to stationarity.

This very important need has motivated a flurry of so called *convergence diagnostics*. These diagnostics are empirical tests which, given multiple realisations of the chain, can give a measure in confidence that the chain has reached stationarity. Examples of these methods are Geweke's statistic, Gelman and Rubin's method, Raftery and Lewis and Heidelberg and Welch Diagnostic. As much as they are important for the practical use of MCMC, we shall skip the details of these methods, and merely refer the interested reader to Chapter 8 of [13].

Even if we assume that the chain X_n is in stationarity (or sufficiently close), so that the bias in the estimator $\hat{I}_n$ can be neglected, we still have to deal with the fluctuations of $\hat{I}_n$ around the mean. Recall that for a stationary chain

$$\begin{aligned} n\mathrm{Var}[\hat{I}_n] &= \mathrm{Var}_\pi[f] + 2\sum_{k=1}^{n-1} \mathrm{Cov}[f(X_0), f(X_k)] \\ &= \mathrm{Var}_\pi[f]\left[1 + 2\sum_{k=1}^{n-1}\rho_k\right], \end{aligned}$$

where $\rho_k = \mathrm{Cov}[f(X_0), f(X_k)]/\sigma_f^2$ is the autocorrelation of $f(X_n)$. If the autocorrelations were zero, then the variance would be σ_f^2/n, which corresponds to having an iid chain. For general

[5] i.e. the distribution of X_n converges to target distribution π

MCMC however, ρ_k's will not be zero, and we clearly desire them to be as small as possible to ensure that the chain performs effectively. The autocorrelations are heavily dependent on the proposal distribution, so this question boils down to making a good choice of q. For the particular cases of RWMH or MALA, the autocorrelation will depend very strongly on the step size δ:

1. When δ is very small, proposals will be made which are very close to the current state. Even though such proposals are very likely to be accepted, the chain isn't really moving very quickly, and we expect there to be strong correlation between subsequent samples.

2. When δ is too large, proposals are very likely to be rejected, thus the chain will spend a lot of time in the same state before the next acceptance event occurs.

Clearly, there is a sweet-spot for δ between these two extremes which minimises the correlations. In an ideal world we'd like to make an "optimal" choice of proposal density. At this point, it might not even be clear that there is a well-defined criterion of optimality. Clearly, an ideal choice would be to choose $q(\cdot\,|x) = \pi(\cdot)$. This is optimal, in the sense that the autocorrelation will be zero, however it is obviously useless in practice. We need to adopt a practical criterion which allows the comparison of proposal densities in situations where we don't know much about the target distribution. One natural criterion would be an estimate of the autocorrelation, which can be easily estimated from a single realisation of the chain. Indeed, given values $X_1, \dots, X_n$ of the chain, the *sample autocovariance function* is

$$\hat{\gamma}_h = \frac{1}{n}\sum_{i=1}^{n-|h|}(X_{i+|h|} - \overline{X})(X_i - \overline{X}), \quad \text{for } -n < h < n,$$

and the *sample autocorrelation function* is given by

$$\hat{\rho}_h = \frac{\hat{\gamma}_h}{\hat{\gamma}_0}.$$

Plots of the autocorrelation for different lag provide a very convenient means of *eyeballing* the performance of a chain, and many statistical computing libraries come equipped with functions to compute sample autocorrelation from data sets. For example, `GNU-R` has the `acf` function which generates a very helpful plot, while the `Statsbase.jl` library in `Julia` provides the `autocor` function.

Due to the autocorrelations, the variance of the estimator $\hat{I}_n$ for $I = \mathbb{E}_\pi[f(X)]$ always be larger than that of an estimator generated via an empirical average of IID samples from π (assuming that exists). This gives rise to a useful criterion for performance: If we were able to produce IID samples Y_i of π, for what value of N would $\frac{1}{N}\sum_{i=1}^{N} f(Y_i)$ have the same variance as the MCMC estimator? That is, for what value of N do we have

$$\text{Var}[\hat{I}_n] = \frac{1}{n}\text{Var}_\pi[f]\left[1 + 2\sum_{k=1}^{n-1}\rho_k\right] = \frac{\text{Var}_\pi[f]}{N}.$$

This suggests that every n samples of the MCMC estimator corresponds to

$$N = \frac{n}{1 + 2\sum_{k=1}^{n-1}\rho_k} \leqslant n,$$

samples of a (hypothetical) IID sampler. This motivates the notion of *effective sample size*, defined by

$$\mathrm{ESS}[\hat{I}_n] = \frac{1}{1 + 2\sum_{k=1}^{\infty} \rho_k}$$

The effective sample size is typically approximated from a single time-series of the chain, either by computing the sample autocorrelation of the chain, or based on spectral methods. In the `coda` library in `R`, this functionality is provided by the `effectiveSize` function. In Julia, similar functionality is found in the `MCMC.jl` package.

Another useful criterion is the *acceptance rate*, i.e. the average rate at which states are accepted by the MH chain. This can be easily computed directly from the algorithm by measuring the empirical frequency of acceptance. While we can optimise the independence sampler by maximising the acceptance rates, as mentioned above, maximising the acceptance rate will *not* result in the best algorithm for the RWMH and MALA schemes. The question is whether one can find an "optimal" acceptance rate against which to calibrate the step size in the proposal. In Roberts et. al (1997) the authors study optimal proposals for RWMH, and they show that optimality in high dimensions $d \gg 1$ is achieved if $\delta = O(d^{-1})$ and the overall acceptance rate is ≈ 0.234. For MALA, Roberts and Rosenthal later showed that optimality is acheived when $\delta = O(d^{-1/3})$ and the overall acceptance rate is ≈ 0.574. These approximations are not universal (these hold specifically for certain classes of target distribution, and in stationarity), but they are a good "default" calibration candidate.

Constructing Confidence Intervals for MCMC

This is not examinable, but might be useful for assignment.
As in the case of MC estimators, we would like to construct confidence intervals for MCMC simulations to quantify the confidence confidence in the estimator $\hat{I}_n$ of I. Knowing that the CLT holds for the chain, and given an approximation for the asymptotic variance given by (3.7) one can then construct confidence intervals. Here we briefly describe a different approach based on batch means. Let $Z_i = f(X_i)$. The key idea is that, provided a CLT holds,[6] then the *batch-means*

$$\overline{Z}_k(n) = \frac{1}{n/m} \sum_{i=(k-1)n/m}^{kn/m} Z_i,$$

are asymptotically iid with $\mathcal{N}(I, \sigma^2 m/N)$ marginals. In particular, for $\overline{Z}(n) = (\overline{Z}_1(n) + \ldots + \overline{Z}_m(n))/m$, one can then use a standard result that

$$\sqrt{m}\frac{\overline{Z}(n) - I}{s_m(n)} \xrightarrow{D} T_{m-1},$$

where T_{m-1} is a Student t r.v. with $m-1$ degrees of freedom, and **[Fixed typo here]**

$$s_m^2(n) = \frac{1}{m-1}\sum_{k=1}^{m}(\overline{Z}_k(n) - \overline{Z}(n))^2.$$

[6] we actually need a functional CLT, something slightly stronger than the CLT described above

Hence, for an asymptotic $100(1-\delta)\%$ confidence interval, we compute the $1-\delta/2$ quantile of T_{m-1}, say $t_{1-\delta/2}$, and then the confidence interval for I is given by

$$\overline{Z}(n) \pm t_{1-\delta/2}\frac{s_m(n)}{\sqrt{m}}.$$

The quality of the confidence interval depends on the degree to which the batch means are iid normal, which can be improved by taking large batches, which suggests one should choose a small number m of batches, say $5-30$.

3.3 Multilevel Sampling

In this section we describe a few ideas used to improve the efficiency of Monte Carlo simulations. We shall focus on two general ideas:

- Introducing an auxiliary distribution, of which π is a marginal distribution.
- Run multiple "companion" chains in parallel which are ergodic with respect to different distributions, and use the information in each chain to "bridge" across difficult regions in the state space (i.e. across wells).

Given a target distribution of the form

$$\pi(x) \propto \exp(-H(x)),$$

our idea is to ontroduce an auxiliary "temperature" parameter, and consider the "tempered" distribution

$$\pi_T(x) \propto \exp(-H(x)/T).$$

The motivation behind the tempered distribution is that the modes in π_T "flatten out" as T increases, while the modes of π_T become more extreme as $T \to 0$. See Figure 3.3. The companion chains would consist of a collection of parallel running Markov chains indexed by T, each ergodic with respect to π_T. The idea is that if a MCMC sample can move freely among the augmented system according to the Metropolis rule, then good results can be obtained for the distribution with the lowest temperature $T = 1$.

This philosophy has motivated numerous different samplers which exploit this tempering to drastically improve the performance of MCMC, particularly for distributions possessing strong multimodality. Examples include umbrella sampling (which we'll discuss in workbooks), bridge sampling, path sampling, and numerous others. Here we will focus on two particular examples.

3.3.1 Simulated Tempering

This method was proposed by Marinari and Parisi [10] and Geyer and Thompson [3]. The idea is that one constructs a family of distributions

$$\Pi = \{\pi_i(x) \mid i \in I\},$$

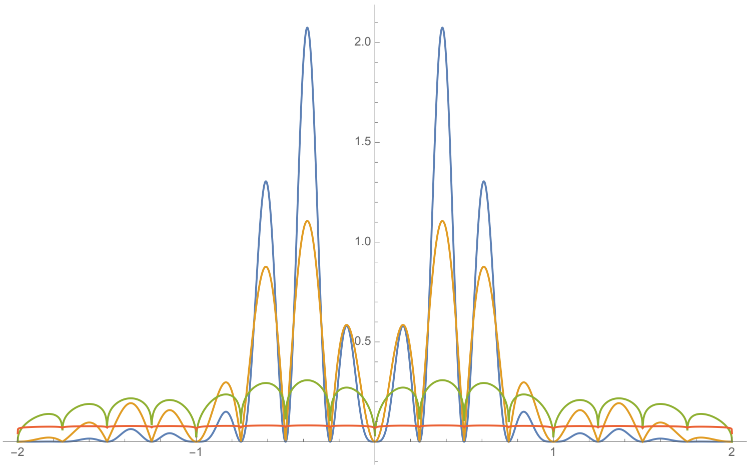


Figure 3.3: Example of tempering a probability distribution, comparing $T = 1$ (blue), $T = 2$ (orange), $T = 10$ (green) and $T = 100$ (red).

where

$$\pi_i(x) \propto \exp(-H(x)/T_i),$$

for an appropriate *temperature schedule* T_i The new target distribution

$$\pi_{st}(x, i) \propto c_i \exp(-H(x)/T_i),$$

is defined on the augmented space $(x, i) \in X \times I$. The constants c_i are chosen so that each tempered distribution has roughly equal chance of being visited. Roughly you'd want $c_i = Z_i = \int e^{-H(x)/T_i}\, dx$, but we are not able to compute these in general, and the c_i are typically tuned via approximate runs. After setting up the augmented distribution, a standard MCMC sampler can be used to draw samples from π_{st}. The intuition behind ST is that by heating up the distribution repeatedly, the new sampler can escape from local modes and increase its chance of reaching other parts of the state space.

Simulated Tempering

Start with $i_0 = 0$. Suppose that the current state is (X_n, i_n), then

1. Draw $u \sim U(0, 1)$.
2. If $u \leqslant \alpha_0$, let $i_{n+1} = i$ and sample X_{n+1} from an MCMC scheme for π_i.
3. If $u > \alpha_0$ let $X_{n+1} = X_n$ and propose a transition $i \to i'$, from a transition function $\alpha(i, i')$ and let $i_{n+1} = i'$ with probability

$$\min\left\{1, \frac{c_i'\pi'(x)\alpha(i', i)}{c_i\pi_i(x)\alpha(i, i')}\right\};$$

otherwise set $i_{n+1} = i_n$

The $\alpha(i,j)$ is the transition matrix for the Markov chain i_n. Geyer and Thompson suggest setting

$$\alpha(i, i+1) = \alpha(i, i-1) = 0.5,$$

and $\alpha(1,2) = 1 = \alpha(m, m-1)$, where $m = \max[I]$, which corresponds to a random walk on I with reflecting boundaries. The constant α_0 is chosen to determine the frequency with which a replica exchange occurs.

3.3.2 Parallel Tempering

The parallel tempering is a very powerful and widely used variant of the simulated tempering algorithm. Some people also refer to it as *replica exchange Monte Carlo*. Instead of augmenting the state space from X to $X \times I$ as in simulated tempering, we instead directly deal with the product space $X_1 \times \ldots \times X_{|I|}$, where X_i are identical copies of X. For a family of distributions $\Pi = \{\pi_i \mid i \in I\}$, we define a joint probability distribution on the produce space as

$$\pi_{pt}(x_1, \ldots, x_I) = \Pi_{i \in I} \pi_i(x_i),$$

and run parallel MCMC chains on all of the X_i. Instead of transitioning between temperatures $i \to i'$ as we did in simulated tempering, we instead swap replicas, see Figure 3.4. The algorithm is given as follows:

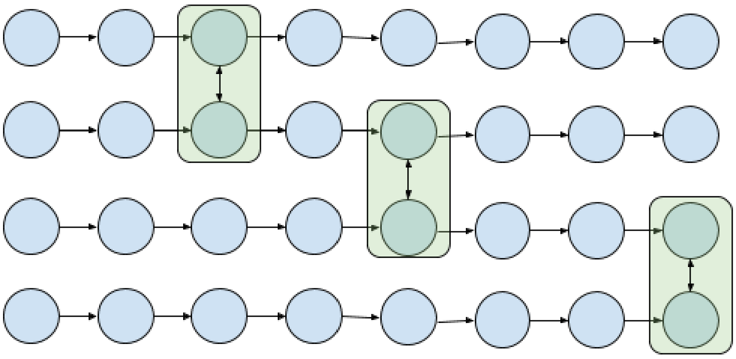


Figure 3.4: Illustration of replica exchange in parallel tempering

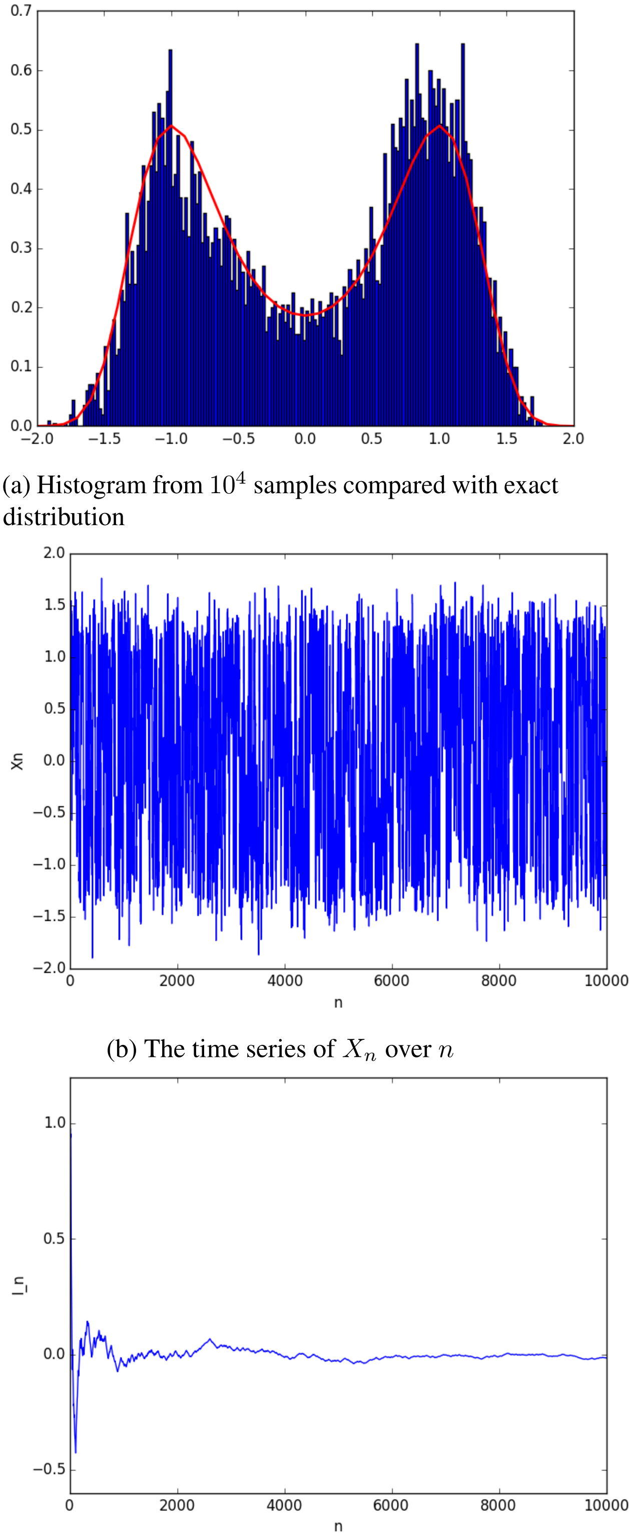
Parallel Tempering

Let the current state be $(X_n^1, \ldots, X_n^I)$.

1. Draw $u \sim U(0,1)$.
2. If $u \leqslant \alpha_0$ perform a *parallel step*: update every X_n^i to X_{n+1}^i via a standard MCMC scheme ergodic with respect to π_i.
3. If $u > \alpha_0$ we conduct a *swapping step*: we randomly choose a neighbouring pair, say i and $i+1$ and propose swapping X_n^i with X_n^{i+1} with probability:

$$\min\left\{1, \frac{\pi_i(X_n^{i+1})\pi_{i+1}(X_n^i)}{\pi_i(X_n^i)\pi_{i+1}(X_n^{i+1})}\right\}$$

This scheme is extremely powerful in simulating complicated systems such as bead polymers and other molecular structures. Also very popular to similar grid based statistical physics models, such as Ising models, etc. Note that unlike simulated tempering, parallel tempering does not need any fine-tuning to adjust the constants c_i. One still needs to make a good choice of the heating schedule $T_1, \ldots, T_N$. The problems of parallel tempering are mainly related to the obvious space cost of keeping track of multiple replicas of the Markov chain. Also, since the system is much larger, typically more time will be required to equilibrate it. Information between neighbouring chains is propagated via swap operations, and these are determined by a slow random walk. This will be a bottleneck of the algorithm.

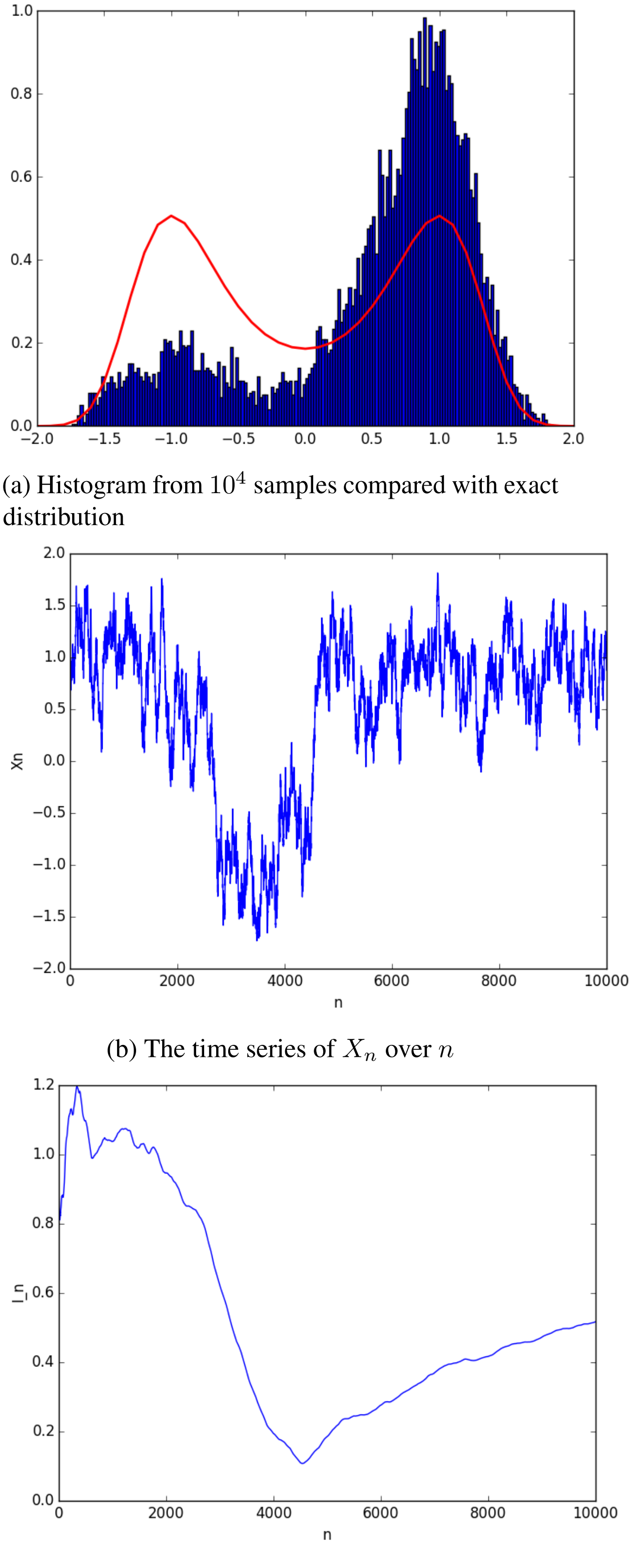


(a) Histogram from 10^4 samples compared with exact distribution

(b) The time series of X_n over n

(c) Estimator $\hat{I}_n$ for $\mathbb{E}_{X\sim\pi}[X]$ over n

Figure 3.5: Simulation results when $r = 1.0$.

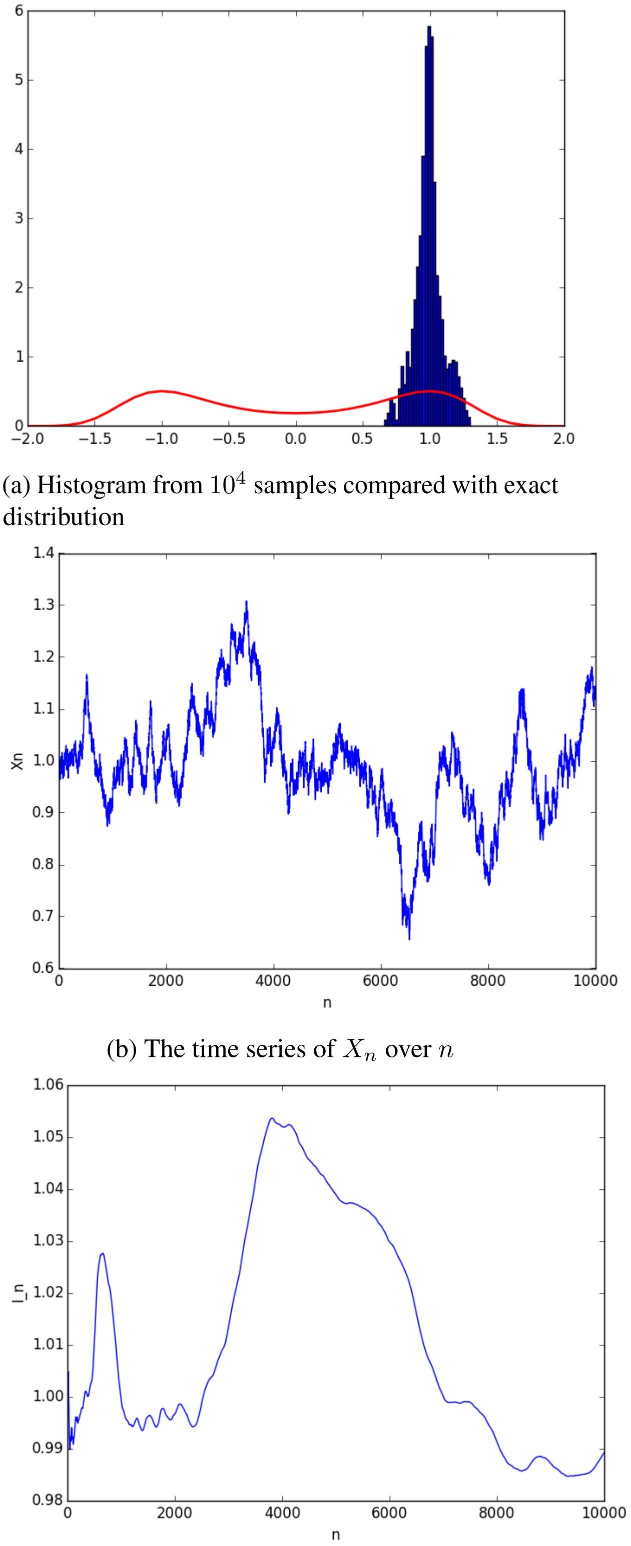


(a) Histogram from 10^4 samples compared with exact distribution

(b) The time series of X_n over n

(c) Estimator $\hat{I}_n$ for $\mathbb{E}_{X\sim\pi}[X]$ over n

Figure 3.6: Simulation results when $r = 0.1$.



(a) Histogram from 10^4 samples compared with exact distribution

(b) The time series of X_n over n

(c) Estimator $\hat{I}_n$ for $\mathbb{E}_{X \sim \pi}[X]$ over n

Figure 3.7: Simulation results when $r = 0.01$.

Chapter 4

Continuous Time Markov Processes

Introduction and Definitions, Simulating Gaussian Processes, Stochastic Differential Equations

Having explored Markov chains and their application to sampling from probability distributions, let us now turn our focus from discrete Markov chains to continuous time Markov processes. As opposed to the previous section, our main interest and application here will not be related to sampling or to computing expectations (although one can certainly do this). Rather our initial focus will be on methods for the accurate and efficient simulation of continuous time Markov processes. For the sake of completeness let us recall some definitions relating to continuous time Markov processes. More details can be found in [8].

Definition 4.1 (Continuous time stochastic process). *Let $(\Omega, \mathcal{F}, \mathbb{P})$ a probability space, and $(E, \mathcal{G})$ a measurable space. A continuous time stochastic process is a collection of random variables $X = \{X_t\,;\, t \in T\}$ such that for each fixed $t \in T$, X_t is a random variable from $(\Omega, \mathcal{F}, \mathbb{P})$ to $(E, \mathcal{G})$, where*

1. $T = [0, \infty)$*, or;*
2. $T = [0, M]$.

The set Ω is known as the sample space, and E is said to be the state space of the stochastic process X_t.

While for Markov chains it was at times more convenient to explain results for discrete state spaces, all the exposition here will be for continuous state spaces. Thus, for the remainder of this chapter, we shall work with the state space E being $\mathbb{R}^d$.

Recall that a stochastic process X may be viewed as a function of both $t \in T$ and $\omega \in \Omega$. We sometimes write $X(t), X(t, \omega)$ or $X_t(\omega)$. There are two ways of viewing the stochastic process: If we fix ω, we can consider the (non-random) map:

$$t \to X(t, \omega) \in E, \qquad \text{for fixed } \omega \in \Omega,$$

i.e. we are looking at the path $X_t(\omega) =: \omega(t)$, i.e. we identify the sample space Ω with the set of paths from 0 to T. Alternatively, we can fix t and consider the map

$$\omega \to X(t, \omega) \in E, \qquad \text{for fixed } t \in T,$$

then this is a random variable, which gives us a snapshot of what is happening (non-deterministically) to all sample points $\omega \in \Omega$ at a fixed time t. Heuristically, this view corresponds X_t being obtained by performing an experiment at each time $t \in T$, which determines the evolution of the stochastic process. Although both viewpoints are equivalent, both can useful in different contexts, as we shall see in the remainder of this chapter.

Definition 4.2 (**Finite-dimensional distributions**). *Given a stochastic process X_t, the family of distributions*

$$\mathbb{P}(X_{t_1} \in B_1, \ldots, X_{t_k} \in B_k),$$

for all $k \in \mathbb{N}$, $t_1, \ldots, t_k \in T$ and $B_1, \ldots, B_k \in \mathcal{G} = \mathcal{B}(\mathbb{R}^d)$ are the finite dimensional distributions of the process X_t.

One might wonder, whether, given a given set of distributions arise as the finite dimensional distributions of a given stochastic process. This is clearly not true in general. However, the Kolmogorov extension theorem (see for example [12, Theorem 2.1.5]) provides a set of consistency conditions in order for a family of distributions to be the FDDs of some stochastic process.

Definition 4.3. *Two stochastic processes X_t and Y_t taking values in E are (stochastically) equivalent if $\mathbb{P}[X_t = Y_t] = 1$ for all $t \in T$. If X_t and Y_t are stochastic equivalent, then X_t is said to be a* version *of Y_t (and vice versa).*

If two stochastic processes are equivalent, then they have the same finite dimensional distributions, the converse is not true, and there will be many versions of the same process. The stochastic processes we are mainly interested in for this chapter all possess an *continuous version*:

Definition 4.4 (**Continuous processes**.). *Let X_t be a continuous-time stochastic process. We will say that X_t is continuous if it has continuous paths, i.e. if the maps $t \to X_t(\omega)$ are continuous for a.e. $\omega \in \Omega$.*

As before, our main focus will be on Markov processes. In the continuous time scenario, the most natural characterisation of Markovianity is via filtrations.

Definition 4.5. *A filtration on $(\Omega, \mathcal{F})$ is a family of sub-σ algebras $\{\mathcal{F}_t\}_{t\in I}$ such that $\mathcal{F}_t \subset \mathcal{F}$ for all $t \in T$ and*

$$\mathcal{F}_s \subset \mathcal{F}_t, \quad \text{if } s \leqslant t.$$

The process $\{X_t\}_{t\geqslant 0}$ is $\mathcal{F}_t$-adapted if the random variable X_t is $\mathcal{F}_t$-measurable for every $t \geqslant 0$.

The most natural filtration to consider is the one generated by the process itself, i.e. the filtration

$$\mathcal{F}_t^X = \sigma(\{X_s\}_{0\leqslant s\leqslant t}),$$

being the smallest σ-field with respect to which X_s is adapted. The σ-algebra $\mathcal{F}_t$ contains all the information available to us about a process up to and including time t. We interpret $A \in \mathcal{F}_t^X$ to mean that by time t, an observer of X knows whether or not A has occurred.

Definition 4.6. *Let X_t be a stochastic process defined on $(\Omega, \mathcal{F}, \mathbb{P})$ with values in $\mathbb{R}^d$ and let $\mathcal{F}_t$ be the natural filtration generated by $\{X_t; t \geqslant 0\}$. Then $\{X_t; t \geqslant 0\}$ is a Markov process if*

$$\mathbb{P}(X_t \in \Gamma \,|\, \mathcal{F}_s) = \mathbb{P}(X_t \in \Gamma \,|\, X_s),$$

for all $s, t \in T$ with $t \geqslant s$ and $\Gamma \in \mathcal{B}(\mathbb{R}^d)$.

Definition 4.7. *A Markov process X_t is* time-homogeneous *if*

$$\mathbb{P}(X_t \in \Gamma \,|X_s) = \mathbb{P}(X_{t-s} \in \Gamma \,|\; X_0), \quad \forall \Gamma \in \mathcal{G}, \; and \; t \geqslant s \geqslant 0.$$

The function

$$P(x, t, B) = \mathbb{P}(X_t \in B \,|\, X_0 = x), \quad \forall t \geqslant 0$$

is the transition probability function *of the process X_t. If we can write*

$$P(x, t, \Gamma) = \int_\Gamma p(x, t, y)\, dy, \qquad t > 0, x \in \mathbb{R}^d.$$

then we call $p(x, t, y)$ the transition density.

Note that the filtration $\mathcal{F}_t$ is generated by events of the form $\{X_{t_1} \in \Gamma_1, X_{t_2} \in \Gamma_2, \ldots, X_{t_n} \in \Gamma_n\}$, for $0 \leqslant t_1, \ldots, < t_n \leqslant t$ and $\Gamma_i \in \mathcal{B}(\mathbb{R}^d)$. Markovianity of X_t is thus equivalent to the hierarchy of equations

$$\mathbb{P}(X_t \in \Gamma \,|X_{t_1}, \ldots, X_{t_n}) = \mathbb{P}(X_t \in \Gamma \,|\, X_{t_n}), \quad \text{a.s},$$

for $n \geqslant 1$ and $0 \leqslant t_1 < t_2 < \ldots < t_n \leqslant t$ with $\Gamma \in \mathcal{B}(\mathbb{R}^d)$.

The transition probability function $P(x, t, \Gamma)$ is a probability measure on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d)$, in particular, $P(x, t, \mathbb{R}^d) = 1$ for all $t \geqslant 0$ and $x \in \mathbb{R}^d$.

4.1 Gaussian Stochastic Processes

A very important class of continuous-time processes is that of Gaussian processes, which arise in many applications

Definition 4.8. *A one-dimensional continuous-time Gaussian process is a stochastic process for which $E = \mathbb{R}$ and all the FDDs are Gaussian, i.e., every finite dimensional vector $(X_{t_1}, X_{t_2}, \ldots, X_{t_k})$ is a $\mathcal{N}(\mu_{t_1,\ldots,t_k}, K_{t_1,\ldots,t_k})$ random variable for some vector $\mu_{t_1,\ldots,t_k}$ and a symmetric non-negative definite matrix $K_{t_1,\ldots,t_k}$, for all $k \in \mathbb{N}$ and $t_1, t_2, \ldots, t_k \in \mathbb{R}$.*

It is straightforward to extend the above definition to arbitrary dimensions. A key feature of Gaussian process is that they are completely characterised by their their mean $\mu(t) := \mathbb{E}X_t$ and covariance function

$$C(t, s) = \mathbb{E}\left[(X_t - \mu(t))(X_s - \mu(s))\right].$$

The natural question that arises is, given a mean function $\mu(t)$ and a covariance function $C(t,s)$, does there exist a Gaussian process X_t with the given mean and covariance. The answer is affirmative provided the covariance $C(t,s)$ is *non-negative definite*, that is,

$$\sum_{i=1}^{k}\sum_{j=1}^{k} C(t_i,t_j)c_i\bar{c}_j \geqslant 0, \tag{4.1}$$

for all $k \in \mathbb{N}$, $t_1,\dots t_k \in \mathbb{R}$, $c_1,\dots,c_k \in \mathbb{R}$.

Remark 4.1. *Note that condition (4.1) is equivalent to having the matrix $\mathcal{C}_{t_1,\dots,t_k}$ is positive definite on $\mathbb{R}^k$, for all $k \in \mathbb{N}$, where $(\mathcal{C}_{t_1,\dots,t_k})_{i,j} = C(t_i,t_j)$, for all i,j.*

Proposition 4.2. *For any function $\mu : T \to \mathbb{R}$ and any non-negative definite function $C : T \times T \to \mathbb{R}$, there exists a Gaussian process X_t on T such that*

$$\mathbb{E}[X_t] = \mu(t), \quad \textit{and} \quad Cov(X_t,X_s) = C(t,s).$$

The proof of this proposition follows by applying the Kolmogorov consistency theorem. See [16, Proposition 4.24.2]. Note that we have not made any claims on the continuity of the process whose existence is guaranteed by the previous proposition. Some well-known examples of Gaussian processes are the following:

1. **Brownian motion:**$\mu(t) = 0$, and $C(t,s) = \min(t,s)$.
2. **Ornstein Uhlenbeck Process:**$\mu(t) = 0$, $C(t,s) = \exp(-|t-s|)$.
3. **Squared Exponential Process:** $\mu(t) = 0$, $C(t,s) = \exp(-|t-s|^2)$.
4. **Fractional Brownian Motion:**$\mu(t) = 0$, $C(s,t) = (t^{2H} + s^{2H} - |t-s|^{2H})/2$, where $H \in (0,1)$ is called the Hurst parameter.

4.2 Stationary Processes

As in the discrete time case, the concept of stationarity carries over to continuous time processes. The general idea remains the same: their statistics remain invariant under time translations. We distinguish between two types of stationary processes: *strictly stationary processes* whose FDD are translation invariant with respect to time, and *weakly stationary processes* whose first two moments are constant over time.

Definition 4.9. *A stochastic process is called (strictly) stationary if all FDDs are invariant under time translation: for all $k \in \mathbb{N}$, for all times $t_i \in T$, and $\{\Gamma_i\}_{i=1}^{k} \subset \mathcal{B}$,*

$$\mathbb{P}(X_{t_1} \in \Gamma_1,\dots,X_{t_k} \in \Gamma_k) = \mathbb{P}(X_{s+t_1} \in \Gamma_1,\dots,X_{s+t_k} \in \Gamma_k),$$

for $s > 0$ such that $s + t_i \in T$, for every $i = 1,\dots k$.

In particular, setting $k = 1$, Definition 4.9 implies that the law of X_t does not depend of t. Stationary processes therefore describe phenomena which do not change in time.

Let X_t be a real-valued random process on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with finite second moment (i.e. $X_t \in L^2(\Omega, \mathbb{P})$ for all $t \in T$). Assume that X_t is strictly stationary. Then

$$\mathbb{E}[X_{t+s}] = \mathbb{E}X_t, \quad \forall s \in T,$$

from which we conclude that $\mathbb{E}X_t = \mathbb{E}X_0$ is constant, and moreover we have that

$$\mathbb{E}\left[(X_{t_1+s} - \mu)(X_{t_2+s} - \mu)\right] = \mathbb{E}\left[(X_{t_1} - \mu)(X_{t_2} - \mu)\right], \quad \forall s \in T.$$

This implies that the covariance function $C(t, s)$ only depends on the difference $t - s$:

$$C(t, s) = C(t - s).$$

This motivates the following definition.

Definition 4.10. *A continuous time stochastic process* $\{X_t\}_{t \in T}$ *is* wide sense stationary *(WSS) or* second order stationary *or* weakly stationary *if it has finite first and second moments and*

1. $\mathbb{E}(X_t)$ *is constant, i.e. it does not depend on t;*
2. $Cov(X_t, X_s)$ *is a function of the difference t − s;*

The function $C(t - s) = Cov(X_t, X_s)$ is the autocovariance function of the process X. Notice that for mean-zero processes, $C(t) = \mathbb{E}(X_t X_0)$, whereas $C(0) = \mathbb{E}X_t^2$, which is finite, by assumption. Since we have assumed that X_t is a real valued process, we have that $C(t) = C(-t)$, $\forall t \in \mathbb{R}$.

From the discussion above, it is clear that a strictly stationary $L^2(\Omega)$ random variable is also wide-sense stationary. The converse is not true in general. An exception to this is the case of Gaussian processes:

Lemma 4.3. *A Gaussian process is strictly stationary if and only if it is weakly stationary.*

Proof. We know that Gaussian distributions are determined by their mean vector and covariance matrix. Since the mean and covariance of a weakly stationary process do not change when the times are shifted, this implies that the finite dimensional distributions are invariant under time shift. □

4.3 Brownian Motion

Definition 4.11 (Standard Brownian Motion)**.** *We define a* Wiener Process *or* Brownain Motion *BM to be a real-valued stochastic process* $(B_t)_{t \geqslant 0}$ *such that*

1. $W_0 = 0$,
2. $W_t - W_s \sim \mathcal{N}(0, t - s)$ *for all* $0 \leqslant s \leqslant t$,

3. *Increments over non-overlapping time intervals are independent: for all $n \in \mathbb{N}$ and $t_1, \ldots, t_n$, such that $0 \leqslant t_1 < t_2 < \ldots < t_n$, the increments $W_{t_1}, W_{t_2} - W_{t_1}, \ldots, W_{t_n} - W_{t_{n-1}}$ are independent.*

From the definition it follows that

1. W_t is a Gaussian process.

2. $\mu(t) = \mathbb{E}[W_t] = 0$, for all $t \geqslant 0$.

3. $Cov(t, s) = \mathbb{E}(W_t W_s) = \min(t, s)$. Indeed, suppose $s \leqslant t$, then
$$\mathbb{E}[W_t W_s] = \mathbb{E}\left[(W_t - W_s + W_s)W_s\right] = \mathbb{E}\left[W_s^2\right] = s.$$

4. From (ii), it follows that for all $a \leqslant b$
$$\mathbb{P}[W_t \in (a, b)] = \frac{1}{\sqrt{2\pi t}} \int_a^b e^{-\frac{x^2}{2t}}\, dx.$$

One can easily generalise the above definition to higher dimensions: an *n-dimensional standard BM* is an n-vector $(W_1(t), \ldots, W_n(t))$ of indendent, one dimensional BMs.

Example 4.1 (Brownian motion as the limit of a Random Walk)**. *[I'm rewriting this with X_i taking values ± 1.]*** *We can use the CLT to show that Brownian Motion arises as a rescaled random walk. To this end, let $B^{\nu,\tau(t)}$ denote the position of our particle at time $t = n\tau$ and let $(X_i)_i$ be i.i.d random variables with $\mathbb{P}(X_i = -1) = \frac{1}{2} = \mathbb{P}(X_i = 1)$. The nwe can define the random walk $B^{\nu,\tau}(t)$ taking value*
$$B^{\nu,\tau}(t) = S_n \nu,$$
at time $t = n\tau$, where $S_n = \sum_{i=1}^n X_i$ One can check that $\mathbb{E}(B^{\nu,\tau}(t)) = 0$ and
$$Var(B^{\nu,\tau}(t)) = \nu^2 n = \nu^2 n = t\frac{\nu^2}{\tau}.$$
Now, assuming $\nu^2/\tau = 1$, we can rewrite the above as
$$B^{\nu,\tau}(t) = \frac{S_n}{\sqrt{n}}\sqrt{n}\nu = \frac{S_n}{\sqrt{n}}\sqrt{t}.$$

Applying the Central Limit Theorem for IID sequences, given in theorem 2.6, we obtain
$$\begin{aligned}\lim_{\substack{n\to\infty \\ t=n\tau}} \mathbb{P}(a \leqslant B^{\nu,\tau}(t) \leqslant b) = \lim_{n\to\infty} \mathbb{P}&\left(\frac{a}{\sqrt{t}} \leqslant \frac{S_n}{\sqrt{n}} \leqslant \frac{b}{\sqrt{t}}\right) \\ =&\frac{1}{\sqrt{2\pi}} \int_{\frac{a}{\sqrt{t}}}^{\frac{b}{\sqrt{t}}} e^{-x^2/2}\, dx.\end{aligned}$$

We have thus proved that, for each t, $B^{\nu,\tau}(t) \xrightarrow{\mathcal{D}} W_t$, as $n \to \infty$ where $\tau = t/n$.

Actually we have a far more powerful result known as a *functional central limit theorem* which shows that a similar limit exists in more general scenarios.

Theorem 4.4. *(Donsker's theorem) Let $\{X_i\}_{i \geqslant 0}$ be iid random variables with $\mathbb{E}X_1 = 0$ and $\mathbb{E}X_i^2 = 1$. Define*

$$S_n = X_1 + X_2 + \ldots + X_n.$$

Let

$$Z_n(t) = \frac{S_{[nt]}}{\sqrt{n}}, \quad 0 \leqslant t \leqslant 1.$$

Then $Z_n \overset{D}{\Rightarrow} W$ where W is a Brownian motion on $[0, 1]$.

Proof. The interested reader is invited to consult [8]. □

4.4 Simulating Gaussian Processes

Suppose we wish to simulate a Gaussian process $X(t)$ with given mean $\mu(t)$ and covariance $C(s, t)$ at a finite number of timesteps, say $t_0, \ldots, t_N$. By definition of the Gaussian process, the random vector

$$\mathbf{X} = (X(t_0), X(t_1), \ldots, X(t_N)),$$

is a multivariate Gaussian random variable with mean given by

$$\mathbf{m} = (\mu(t_0), \ldots, \mu(t_N)) \in \mathbb{R}^N \tag{4.2}$$

and $N \times N$ covariance matrix

$$\Sigma_{i,j} = \mathrm{Cov}(X(t_i), X(t_j)). \tag{4.3}$$

We know that the matrix Σ is both symmetric and non-negative definite. Thus, as described in Section 2.2.4, we can generate samples from $\mathcal{N}(\mathbf{m}, \Sigma)$ by using a Cholesky decomposition of Σ.

Sampling from Gaussian Process: Method 1

Suppose we wish to sample $X(t)$ over timesteps t_0 to t_N:

1. Generate the mean vector $\mathbf{m}$ as in (4.2).
2. Generate the covariance matrix Σ as in (4.3).
3. Generate sample from the distribution $\mathcal{N}(\mathbf{m}, \Sigma)$.

Exercise 4.1. *Implement code to generate samples of the four well known Gaussian processes described previously.*

The above method is exact, in the sense that we are able to exactly simulate the value of $X(t)$ at the fixed timesteps $t_0, \ldots, t_N$. Of course, this is only possible for finitely many points, and if we wish to generate samples from $X(t')$ where t' lies between two points, then we can employ some interpolation, which will result in bias being introduced. By increasing N the mesh will get finer, and the interpolation error (and thus the bias) will decrease, however we must be mindful of the ocmputational cost. The cost of the Cholesky algorithm is $O(N^3)$, which grows quite quickly.

A very useful property of multivariate Gaussian random variables is that if we condition on part of the random vector, the resulting distribution remains Gaussian. To see this, suppose that

$$\mathbf{X} \sim \mathcal{N}(\mathbf{m}, \Sigma),$$

and suppose that we can write $\mathbf{X}$ as $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2)^\top$, and

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$

Proposition 4.5. *Given $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2)^\top \sim \mathcal{N}(\mathbf{m}, \Sigma)$, then the marginal distributions of X_1 and X_2 satisfy:*

$$\mathbf{X}_1 \sim \mathcal{N}(\mathbf{m}_1, \Sigma_{11}),$$
$$\mathbf{X}_2 \sim \mathcal{N}(\mathbf{m}_2, \Sigma_{22}).$$

The conditional distribution of $\mathbf{X}_2$ conditional on $\mathbf{X}_1$ is a multivariate normal with

$$\mathbb{E}[\mathbf{X}_2 \,|\, \mathbf{X}_1] = \mathbf{m}_2 + \Sigma_{21}\Sigma_{11}^{-1}(\mathbf{X}_1 - \mathbf{m}_1).$$

and

$$\mathit{Var}(\mathbf{X}_2 \,|\, \mathbf{X}_1) = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}.$$

Proof. Exercise (to be put in Exercise sheet.) □

Using these properties we can develop are more efficient scheme to sample Gaussian processes, more specifically to interpolate between already simulated points of a Gaussian process. The idea is to draw new points conditional on the values that we have already generated. For example if we have previously generated a sample of the Gaussian process at values 0.5 and 1.0, then we subsequently generate exact samples at points 0.25 and 0.75 conditional on $X(0.5)$ and $X(1.0)$.

As a particular example consider the folowing recursive algorithm. Suppose we wish to generate X_{n+1} at time t_{n+1} given that we have already generated $X_0, \ldots, X_n$. We thus need to specify the condition distribution of X_{n+1} given $X_0, \ldots, X_n$. Writing

$$R_{n+1} = \begin{bmatrix} R_n & \mathbf{r}_n \\ \mathbf{r}_n^\top & C(t_n, t_{n+1}) \end{bmatrix},$$

where $\mathbf{r}_{n,i} = C(t_{n+1}, t_i)$. Using Proposition 4.5 we know that $X_{n+1} \sim \mathcal{N}(m, \sigma^2)$ where

$$m = \mu(t_{n+1}) + \mathbf{r}_n \cdot R_n^{-1}((X(t_0), \ldots, X(t_n)) - (\mu(t_0), \ldots, \mu(t_n)))^\top,$$

and

$$\sigma^2 = C(t_n, t_{n+1}) - \mathbf{r}_n \cdot R_n^{-1}\mathbf{r}_n.$$

A very efficient scheme can be constructed by combining this update formula with the Cholesky decomposition.

Exercise 4.2. *Suppose additionally that the Gaussian process $X(t)$ is Markovian, so that in particular, you only need to know the value of $X(t_n)$ to generate $X(t_{n+1})$. Construct a scheme to iteratively sample $X(t_i)$ over a sequence of points $t_0 < t_1 < t_2 < \ldots$.*

1. *In the case of Brownian motion, show that the update formula can be written as:*
$$X(t_{i+1}) = X(t_i) + \left(\sqrt{t_{i+1} - t_i}\right) Z,$$
where $Z \sim \mathcal{N}(0,1)$.
2. *Derive a similar update formula for the stationary Ornstein-Uhlenbeck process with mean* 0 *and covariance $C(s,t) = \exp(\alpha|t-s|/2)$.*

4.4.1 Simulating Stationary Gaussian Processes

While the above methods are applicable for simulating general Gaussian processes on general meshes $t_0 < t_1 < \ldots < t_N$ they are computationally expensive, since they ultimately will require $O(N^3)$ floating point operations to generate a single sample. However, in the particular case where we wish to simulate a *stationary* Gaussian process on a regular mesh $\{0, \Delta t, 2\Delta t, \ldots, N\Delta t\}$, then we can reduce the problem to a discrete Fourier transform and obtain an almost magical speedup by employing a Fast-Fourier Transform.

Indeed, suppose we wish to simulate a stationary Gaussian process $X(t)$ with mean $\mu = 0$ and covariance $C(t,s) = C(t-s)$. Then given timesteps $\{0, \Delta t, 2\Delta t, \ldots, n\Delta t\}$, the random vector $(X(t_1), X(t_2), \ldots, X(t_n))$ has a covariance matrix of the form:

$$\Sigma = \begin{pmatrix} c_0 & c_1 & c_2 & \cdots & c_n \\ c_1 & c_0 & c_1 & \cdots & c_{n-1} \\ c_2 & c_1 & c_0 & \ddots & c_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_n & c_{n-1} & c_{n-2} & \cdots & c_0 \end{pmatrix}.$$

Notice that the matrix Σ is symmetric, and is constant along the diagonals of the matrix. This matrix is a *symmetric Toeplitz matrix.*

Definition 4.12. *(Toeplitz Matrix) A matrix Σ is said to be a Toeplitz matrix, if each diagonal takes a constant value, that is*
$$\Sigma_{i,j} = \Sigma_{i+1,j+1},$$
for all $i, j \in \{1, \ldots, n-1\}$.

Definition 4.13. *A circulant matrix is a matrix of the form:*

$$A = \begin{pmatrix} a_0 & a_{n-1} & \cdots & a_2 & a_1 \\ a_1 & a_0 & \cdots & a_3 & a_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_n & a_{n-1} & a_{n-2} & \cdots & a_0 \end{pmatrix}.$$

A circulant matrix is constructed by starting with a vector $\mathbf{a} = (a_0, \dots, a_n)$ as the first row, and obtaining the each row by a periodic left shift of the previous row.

The important observation that we shall make use of is that we can embed Σ in a $2n \times 2n$ *circulant matrix* as follows:

$$\Pi = \begin{pmatrix} c_0 & c_1 & c_2 & \cdots & c_n & c_{n-1} & c_{n-2} & c_{n-3} & \cdots & c_1 \\ c_1 & c_0 & c_1 & \cdots & c_{n-1} & c_n & c_{n-1} & c_{n-2} & \cdots & c_2 \\ c_2 & c_1 & c_0 & \ddots & c_{n-2} & c_{n-1} & c_n & c_{n-1} & \cdots & c_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_n & c_{n-1} & c_{n-2} & \cdots & c_0 & c_1 & c_2 & c_3 & \cdots & c_{n-1} \\ c_{n-1} & c_n & c_{n-1} & \cdots & c_1 & c_0 & c_1 & c_2 & \cdots & c_{n-2} \\ c_{n-2} & c_{n-1} & c_n & \cdots & c_2 & c_1 & c_0 & c_2 & \cdots & c_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_1 & c_2 & c_3 & \cdots & c_{n-1} & c_{n-2} & c_{n-3} & c_{n-4} & \cdots & c_0 \end{pmatrix}.$$

Unfortunately, this matrix will not be nonnegative definite in general, which we require, however this will hold under some additional assumptions.

Lemma 4.6. *Suppose that* $c_0 \geqslant c_1 \geqslant \cdots \geqslant c_n \geqslant 0$ *and*

$$2c_k \leqslant c_{k-1} + c_{k+1},$$

for $k = 1, \dots, n-1$*, then* C *is a covariance matrix, i.e.* C *is nonnegative definite.*

Why are we interested in this representation? At first glance this might seem like a futile exercise, however the importance of this embedding arises from the connection between circulant matrices and the discrete Fourier transform.

Definition 4.14. *Given a vector* $\mathbf{x} = (x_0, \dots, x_n)^\top \in \mathbb{C}^n$*, define the discrete Fourier transform of* $\mathbf{x}$ *by the vector*

$$(\mathcal{F}\mathbf{x})_j = \sum_{k=0}^{n-1} e^{-(2\pi i/n)jk} x_k = \sum_{k=0}^{n-1} \omega^{jk} x_k,$$

for $j = 0, \dots, n-1$ *where* $\omega = e^{-2\pi i/n}$.

Therefore the computing discrete fourier transform of $\mathbf{x}$ is equivalent to computing $F\mathbf{x}$, where

$$F = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \ddots & \omega^{2(n-1)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)^2}. \end{pmatrix}.$$

It turns out that $F^{-1} = \overline{F}/n$. Normally, computing the matrix-vector produce $F\mathbf{x}$ would require $O(n^2)$ operations, however using a Fast Fourier Transform reduces this to $O(n \log n)$ steps. The connection to circulant matrices is the following. Let $\mathbf{b} = (b_0, b_1, \ldots, b_{n-1})^\top$ be a complex valued vector, and let B be the circulant matrix *generated* by $\mathbf{b}$, i.e.

$$B = \begin{pmatrix} b_0 & b_{n-1} & \cdots & b_2 & b_1 \\ b_1 & b_0 & \cdots & b_3 & b_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{n-1} & b_{n-2} & \cdots & b_1 & b_0 \end{pmatrix}.$$

We then have the following fundamental result: **[Typo here was fixed]**

Lemma 4.7. *The circulant matrix B is diagonalized by the DFT matrix F. More specifically,*

$$B = FDF^{-1},$$

where D is a diagonal matrix containing the eigenvalues of B, $\lambda_0, \ldots, \lambda_{n-1}$,

$$\lambda_i = (F\mathbf{b})_i.$$

This factorization is very commonly exploited in both numerical PDE schemes, as well as methods to perform efficient matrix-vector multiplication. Our objective is to generate a sample from the Gaussian distribution $\mathcal{N}(0, \Pi)$, which entails computing a square root of the matrix Π. To this end, assuming that B is non-negative definite (i.e. nonnegative eigenvalues) define

$$E = F\text{diag}\left(\left\{\sqrt{\lambda_i/n}\right\}_{i=0}^{n-1}\right).$$

Then

$$E\overline{E}^\top = F\text{diag}\left(\{\lambda_i/n\}_{i=0}^{n-1}\right)\overline{F} = F\text{diag}(\lambda_i)F^{-1} = B.$$

Therefore, E is the square root of B in $\mathbb{C}^{n \times n}$. Now that E is a complex–valued matrix. Let us write $E = E_1 + iE_2$ for real valued matrixes. Then

$$E\overline{E}^\top = E_1E_1^\top + E_2E_2^\top + i(E_2E_1^\top - E_1E_2^\top),$$

and since B is real-valued, $B = E_1E_1^\top + E_2E_2^\top$. Now let $\mathbf{Z}_1$ and $\mathbf{Z}_2$ be $\mathcal{N}(0, I)$ and define

$$\mathbf{X} = E\mathbf{Z} = (E_1 + iE_2)(\mathbf{Z}_1 + i\mathbf{Z}_2) = (E_1\mathbf{Z}_1 - E_2\mathbf{Z}_2) + i(E_2\mathbf{Z}_1 + E_1\mathbf{Z}_2).$$

Letting $\mathbf{X}_1 = \text{Re}[\mathbf{X}]$, then

$$\mathbb{E}[\mathbf{X}_1] = \mathbf{0},$$

and

$$\text{Var}(\mathbf{X}_1) = \text{Var}(E_1\mathbf{Z}_1 - E_2\mathbf{Z}_2) = E_1E_1^\top + E_2E_2^\top = B.$$

This give us a method for sampling a stationary Gaussain process at equidistant intervals.

Generating a stationary Gaussian process

Assume $\mu = 0$ and we are given covariance C and let $t_i = i\Delta t$, $i = 0, \ldots, n-1$.

1. Set $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1}, c_n, c_{n-1}, \ldots, c_1)$, where $c_i = C(t_0, t_i)$.
2. Set $\lambda = F\mathbf{c}$ using FFT.
3. Generate $\mathbf{Z} = \mathbf{Z}_1 + i\mathbf{Z}_2$, $\mathbf{Z}_1, \mathbf{Z_2} \sim \mathbf{N}(0, I)$.
4. Compute $\mathbf{Y} = \sqrt{\text{diag}(\lambda/n)}\mathbf{Z}$.
5. Compute $\mathbf{V} = F\mathbf{Y}$ using FFT.
6. Output $\mathbf{X} = \text{Re}(V_0, \ldots, V_n)^\top$.

If we performed naive matrix-vector multiplications this algorithm would cost $O(n^2)$, however using the FFT, the algorithm is $O(n \log n)$.

So what happens when the embedding circulant matrix is not nonnegative definite? Then we cannot use this approach directly. However, there are two possible approaches to generate a sample in this case:

1. Embed the symmetric Toeplitz in an even larger circulant matrix.
2. Use only the positive part of the circulant matrix.

As discussed in [18] it is typically always possible choose an large circulant matrix which is nonnegative definite. In this case, we can use the above exact scheme for generating the sample. If we must resort to option (2), then the procedure is approximate. However, one can typically quantify the error incurred in this case, so the approach is still feasible.

While this algorithm provides a perfectly adequate scheme for simulating stationary Gaussian processes, the true power of this method can be seen when using it to simulate stationary *Gaussian random fields*, i.e. a $\mathbb{R}^d$–indexed Gaussian process. Indeed, many scientific computing software libraries provide algorithms for simulating stationary GRFs based on circulant embeddings. See [9] for more information.

Exercise 4.3. *Consider the stationary Gaussian process with exponential covariance* $C(\tau) = e^{-|\tau|/l}$.

1. *Implement a method for simulating this process in a programming language of your choice.*
2. *(Challenging optional exercise:) Show that the eigenvalues* λ_i *in this case are always positive.*

4.5 SDEs and Diffusion Processes

One very important class of continuous-time stochastic processes arise as solutions of Stochastic Differential Equations (SDEs). These models play a prominent role in a range of application areas, including biology, chemistry, epidemiology, mechanics, microelectronics, economics, and finance. A complete understanding of SDE theory requires some familiarity with advanced probability and stochastic processes (though I certainly wouldn't want to deter an interested reader from [12] and [6]). Hopefully you will developed an understanding of the use of SDEs in the *Applied Stochastic Processes* or another equivalent module. In this section we shall describe how to simulate SDEs numerically, and analyse the behaviour and performance of each scheme.

At the simplest level, we can consider an SDE as adding a noise term to the right-hand side of a differential equation. We would like to construct the stochastic analogue of the ODE:

$$\dot{x}(t) = f(x), \quad x(0) = x_0,$$

which has solution of the form

$$x(t) = x_0 + \int_0^t f(x(s))\, ds,$$

that is, an SDE for X_t

$$\dot{X}_t = f(X_t) + g(X_t)\zeta_t, \quad X(0) = x_0, \tag{4.4}$$

where ζ_t is a source of noise. In most situations, scientists and engineers consider systems where ζ_t is *white*, in particular, the noise satisfies the following properties:

1. $\mathbb{E}\zeta_t = 0$.
2. ζ_t is independent of ζ_s if $s \neq t$. Formally $\mathbb{E}(\zeta_t \zeta_s) = \delta(t - s)$.
3. ζ_t is strictly stationary.

Unfortunately, no such process exists as a function in the space of real-valued paths. In any case, our intuition at this point is that a good candidate for ζ_t would be the derivative of Brownian motion, in some sense, so that $\zeta_t = \dot{W}_t = \frac{dW}{dt}$ is formally the derivative of Brownian motion. Clearly $\dot{W}_t$ doesn't exist in any ordinary sense of the derivative. Following our intuition from ODEs, by a solution of this SDE, we mean a stochastic process X_t which satisfies

$$X_t = x_0 + \int_0^t f(X_s)\, ds + \int_0^t g(X_s)\, dW_s.$$

The second term $\int g(X_s)\, dW_s$ is a *stochastic integral*, which has yet to be defined. In the next section we will construct a meaningful interpretation which corresponds to our intuition as to how a process driven by white noise should behave.

4.5.1 Stochastic Integrals

Unless otherwise stated, throughout this section we will be referring to real-valued stochastic processes. Inspired by the construction of the Riemann integral, we want to do something similar to define the stochastic integral. In doing so we need to bear in mind that we are attempting to integrate a stochastic process - as opposed to a function - with respect to another stochastic process, and while the integral of a function is, for a fixed T, the stochastic integral is a random variable.

1. Analogous to step functions, we define an *elementary* or *simple* process on $[0, T]$, to be a process of the form
$$\psi(t, \omega) = \sum_{j=0}^{K} \phi_j(\omega) \mathbf{1}_{[t_j, t_{j+1}]},$$
where $\{0 = t_0, t_1, \ldots, t_K = T\}$ is again some partition of the interval $[0, T]$ and the $\phi_j' s$ are random variables, (a.s.) uniformly bounded in j and ω. For processes of this form, it seems reasonable to define
$$\int_0^T \psi(t, \omega) dW_t := \sum_{j=0}^{K} \phi_j(\omega)(W(t_{j+1}) - W(t_j)),$$
thus mimicking a Riemann-Stieljes integral. The $W(t_{j+1}) - W(t_j)$ is an increment of the Brownian motion, which we know has distribution $\mathcal{N}(0, t_{j+1} - t_j)$.
2. Then given a process $f(t, \omega)$ we wish to find a sequence ψ_n of elementary processes that approximate $f(t, \omega)$ and define the stochastic integral $\int_0^T f(s, \omega) dW_s$ to be the limit of the stochastic integrals of the ψ_n.

For a very simple example, consider if $\phi_j = 1$ for all j, so that
$$\psi(t, \omega) = \sum_{j=0}^{K} \mathbf{1}_{[t_j, t_{j+1}]} = 1.$$

In this case,
$$\begin{aligned}\int_0^T 1 \, dW_t &= \sum_{j=0}^{K} W(t_{j+1}) - W(t_j)) \\ &= W_{t_K} - W_{t_0} \\ &= W_T.\end{aligned}$$

Thus in this case, we're just adding up the increments of $W(t)$, to get $W(T)$. In the more general case, roughly speaking, we're weighting the increments with a random variable $\phi_j(\cdot)$. This construction sounds reasonable, however, there are a couple of problems and points to clarify. Firstly, assuming that the process above does work, we would expect the limit to be independent of the point $t_* \in [t_j, t_{j+1}]$ that we choose to approximate the integrand. It turns out that this is not the case, even if the integrand is continuous. We can show this fact with an example

Example 4.2. *Suppose we want to calculate $\int_0^T W_t dW_t$. We consider a partition with mesh size 2^{-n} and approximate the integral with the elementary process*
$$\phi_n = \sum_{j \geqslant 0} W(t_j) \mathbf{1}_{[t_j, t_{j+1}]}.$$

Then, for every $n \in \mathbb{N}$,

$$\mathbb{E}\sum_{j\geqslant 0} W(t_j)[W(t_{j+1}) - W(t_j)] = 0,$$

since $W(t_{j+1}) - W(t_j)$ is independent of $W(t_j)$. However, if instead we approximate the integrand with the process $\tilde{\psi}_n = \sum_{j\geqslant 0} W(t_{j+1})\mathbf{1}_{[t_j,t_{j+1}]}$, then

$$\mathbb{E}\sum_{j\geqslant 0} W(t_{j+1})\,[W(t_{j+1} - W(t_j)] = \;\; \mathbb{E}\sum_{j\geqslant 0}(W(t_{j+1}) - W(t_j))\,[W(t_{j+1} - W(t_j)] \xrightarrow{n\to\infty} T. \tag{4.5}$$

Brownian motion is continuous so what is the problem? Both ψ_n and $\tilde{\psi}_n$ seem perfectly reasonable approximating sequences, so why are we obtaining different results? The problem is that Brownian motion, being a.s. non-differentiable, simply "varies too much" in the interval $t_* \in [t_j, t_{j+1}]$ and this leads to the phenomenon illustrated above.[1] There is no way around this pickle, it is simply a fact that different choices of $t_* \in [t_j, t_{j+1}]$ lead to different definitions of the stochastic integral. The most popular choices are

1. $t_* = t_j$ which gives the *Itô integral*, and
2. $t_* = (t_j + t_{j+1})/2$ which gives the *Stratonovich integral*.

We will discuss the differences between these two stochastic integrals later on. For the moment, we will stick to the choice $t_* = t_j$, i.e. the Itô interpretation of the stochastic integral.

4.6 Stochastic integral in the Itô sense.

In these notes we will not go through all the details of the proofs of the construction of the Itô integral. A very accessible construction of the stochastic integral can be found in [12].

To construct an Itô integral $\int_0^T f(t,\omega)\,dW_t$, we require the following assumptions on f:

1. $f(t,\omega)$ is $\mathcal{B}\times\mathcal{F}$-measurable where $\mathcal{B}$ is the Borel σ-algebra of $[0,T]$.
2. $f(t,\cdot)$ is $\mathcal{F}_t$ adapted for all $t \geqslant 0$, where $\mathcal{F}_t$ is the natural filtration associated with the BM W_t;
3. $\mathbb{E}\int_0^T f(t,\omega)^2 dt < \infty$.

Definition 4.15. *We denote by $\mathcal{J}(0,T)$, or simply $\mathcal{J}$ the class of stochastic processes $f(t,\omega)$: $\mathbb{R}_{\geqslant 0}\times\Omega \to \mathbb{R}$ for which the above three properties hold.*

Now that know how to interpret $\int_0^T f(t,\omega)dW_t$, at least in the Itô sense. Let us now list all the properties that help in the calculation of the stochastic integral

Theorem 4.8 (Properties of the Itô integral.). *For every $f, g \in \mathcal{J}(0,T)$, $t, s \in [0,T]$ and for every $\alpha, \beta \in \mathbb{R}$:*

[1] Note that, Riemann-Stieljes integrals $\int_0^T f(x)dg(x)$ over f with respect to g requires that g has bounded total variation. However, Brownian motion has a.s. infinite total variation.

1. *Additivity:* $\int_0^t f dW_t = \int_0^S f dW_t + \int_S^T f dW_t$
2. *Linearity:* $\int_0^T (\alpha f + \beta g) dW_t = \alpha \int_0^T f dW_t + \beta \int_0^T g dW_t$*;*
3. $\mathbb{E} \int_0^T f dW_t = 0$*,*
4. $I_T := \int_0^T f dW_t$ *is* $\mathcal{F}_T$*-measurable*
5. I_t *admits a continuous version*
6. *Itô isometry:*
$$\mathbb{E}\left(\int_0^T f dW_t\right)^2 = \mathbb{E}\int_0^T f^2 dt,$$
or, more generally,
$$\mathbb{E}\left(\int_0^t f(u) dW_u \int_0^s g(u) dW_u\right) = \mathbb{E}\int_0^{\min(t,s)} f(u)g(u)\, du$$
7. *If* $f(t)$ *is a deterministic function, then* I_t *is a Gaussian random variable with mean zero, and variance* $\int_0^t f^2(s)\, ds$*.*

Definition 4.16. *Let* $(\Omega, \mathcal{F}, \mathbb{P})$ *be a probability space and* $\mathcal{F}_t$ *a filtration to which the process* M_t *is adapted. We say that* M_t *is a (square-integrable)* $\mathcal{F}_t$*-martingale if* $\mathbb{E}|M_t|^2 < \infty$ *and*

$$\mathbb{E}[M_t \,|\mathcal{F}_s] = M_s, \quad \forall t \geqslant s.$$

Proposition 4.9. *For* $f \in \mathcal{J}([0,T])$*, the Itô stochastic integral* $I_t = \int_0^t f(s,\omega) dW_s$ *is a square-integrable* $\mathcal{F}_t$*-martingale.*

4.7 The Itô Formula

Having constructed the Itô integral, we can now make sense of what it means for a process X_t to satisfy

$$dX_t = b(t,\omega)\, dt + \sigma(t,\omega) dW_t,$$

namely, a process which satisfies

$$X_t = X_0 + \int_0^t b(s,\omega)\, ds + \int_0^t \sigma(s,\omega) dW_s.$$

We will call X_t an *Itô process*. At this point you should be convinced that the Itô integral doesn't follow the usual integration rules. The Itô formula provides us with a chain rule for Itô processes.

Theorem 4.10 (Itô's formula)**.** *Let* X_t *be given by*

$$dX_t = b(t,\omega) dt + \sigma(t,\omega) dW_t.$$

Let $f(t,x)$ *be a* $C^{1,2}$ *function (i.e.* C^1 *in time,* C^2 *in space),then the process* $Y_t = f(X_t)$ *satisfies*

$$df(X_t) = \left(\partial_t f(t,X_t) + \partial_x f(t,X_t) b(t,\omega) + \frac{1}{2}\partial_{x,x} f(t,X_t)\sigma^2(t,\omega)\right) dt + \partial_x f(t,X_t)\sigma(t,\omega) dW_t. \tag{4.6}$$

4.7.1 Multidimensional Itô Processes

Our construction of Itô processes can be generalised to higher dimensions. Indeed, we can construct an $\mathbb{R}^d$-valued Itô process, written as:

$$dX_t = b(t,\omega)\,dt + \sigma(t,\omega)dW_t, \tag{4.7}$$

where

$$b : [0,T] \times \Omega \to \mathbb{R}^d,$$
$$\sigma : [0,T] \times \Omega \to \mathbb{R}^{d\times m},$$

and where W_t is an m-dimensional Brownian motion. In this case $\int_0^t \sigma(s,\omega)dW_s$ is simply a d-vector of Itô integrals, where the i^{th} component is given by

$$\left[\int_0^t \sigma dW_s^i\right]_i = \int_0^t \sum_{j=1}^m (\sigma)^{i,j} dW_s^j.$$

Itô's formula is then generalised as follows. If $g = g(t,x) : [0,\infty) \times \mathbb{R}^d \to \mathbb{R}$ and $Y_t = g(t,X_t)$, the multidimensional Itô formula reads:

$$\begin{aligned} dY_t =& \partial_t g(t,X_t)\,dt + \sum_{i=1}^d \partial_{x_i} g(t,X_t)\,dX_t^i + \frac{1}{2}\sum_{i,j=1}^d \partial_{x_i,x_j} g(t,X_t)\,d[X^i,X^j]_t \\ =& \partial_t g(t,X_t)\,dt + \sum_{i=1}^d \partial_{x_i} g(t,X_t)\,dX_t^i + \frac{1}{2}\sum_{i,j=1}^d \partial_{x_i,x_j} g(t,X_t) \sum_{l=1}^m \sigma^{i,l}\sigma^{j,l}\,dt, \end{aligned}$$

or in vector notation

$$dY_t = \partial_t g(t,X_t)\,dt + \nabla g(t,X_t)\cdot dX_t + \frac{1}{2}(\sigma\sigma^\top) : \nabla\nabla g(t,X_t)\,dt,$$

where $A : \nabla\nabla f = \sum_{i,j=1}^d A_{i,j}\partial_{x_i,x_j} f$.

4.8 Stochastic Differential Equations

Let W_t, $t \geqslant 0$ be a Brownian motion process. An equation of the form

$$dX_t = b(t,X_t)\,dt + \sigma(t,X_t)\,dW_t, \quad X_0 = \eta \tag{4.8}$$

where $b(t,x)$ and $\sigma(x,t)$ are given, the initial state η is a given random variable and X_t is the unknown process is called a (Itô) *stochastic differential equation* (SDE), driven by Brownian motion. The functions $b(t,x)$ and $\sigma(x,t)$ are known as the *drift* and *diffusion* coefficients respectively. The solution of such an equation, if it exists, is called an Itô diffusion.

First we will specify what it means to be a solution of an SDE. Let W_t be a Brownian motion, and let $\mathcal{G}_t$ be the filtration generated by W_t and the initial state η.

Definition 4.17. *An process X_t is called a* strong solution *of the SDE (4.8) if it is $\mathcal{G}_t$-adapted, and for all $t > 0$, the integrals $\int_0^t b(s, X_s)\, ds$ and $\int_0^t \sigma(s, X_s)\, dW_s$ exist, with the second being an Itô integral, and*

$$X_t = \eta + \int_0^t b(s, X_s)\, ds + \int_0^t \sigma(s, X_s) dW_s.$$

Such a solution is unique if, whenever $\hat{X}_t$ is another strong solution, $\mathbb{P}(X_t \neq \hat{X}_t) = 0$, for all $t \geqslant 0$.

As with ODEs, we want to establish some general conditions under which the strong solution exists and is unique. The following theorem provides sufficient conditions to existence. Perhaps unsurprisingly, the proof is quite similar to the proof of the analogous Picard existence theorem for ODEs.

Theorem 4.11 (Existence and Uniqueness)**.** *If the following conditions are satisfied*

1. *b and σ are locally Lipschitz in x uniformly in t, that is, for every T and N, there exists a constant $K = K(N, T)$ such that, for all $|x|, |y| \leqslant N$ and $0 \leqslant t \leqslant T$:*

$$|b(t, x) - b(t, y)| + |\sigma(t, x) - \sigma(t, y)| \leqslant K|x - y|,$$

2. *The coefficients b and σ satisfy the* linear growth condition*:*

$$|b(t, x)| + |\sigma(t, x)| \leqslant K(1 + |x|),$$

3. *The initial state η is independent of $(W_t, 0 \leqslant t \leqslant T)$ and $\mathbb{E}\eta^2 < \infty$.*

Then there exists a unique strong solution X_t of the SDE (4.8). X_t has continuous paths, and moreover

$$\mathbb{E}\left(\sup_{0 \leqslant t \leqslant T} X_t^2\right) \leqslant C(1 + \mathbb{E}\eta^2),$$

where the constant C depends only on K and T.

Remark 4.12. *In particular, if the coefficients b and σ have continuous first derivatives, then the local-Lipschitz condition holds.*

Theorem 4.13. *Assume the conditions of Theorem 4.11 hold. The strong solution X_t of the SDE (4.8) is a Markov process.*

4.8.1 Some examples of SDEs

Example 4.3. *Ornstein-Uhlenbeck process Consider a particle of unit mass moving with momentum $p(t)$ at time t moving within a fluid, subject to irregular "kicks" due to neighbouring particles. The dynamics of the particle can be modelled by a dissipative force $-\lambda p(t)$ and a fluctuating force $\sigma\dot{\xi}(t)$, where $\xi(t)$ is the white noise process. Newton's second law gives us:*

$$\dot{p}(t) = -\lambda p(t) + \sigma\dot{\xi}(t),$$

which can be interpreted as the following SDE:

$$dP_t = -\lambda P_t\, dt + \sigma\, dW_t.$$

This is known as the Ornstein Uhlenbeck *process. It is one of the few SDEs that can be solved exactly. Those who attended the Applied Stochastic Processes course will be very familiar with this process. We shall revisit properties of this process later on.*

Example 4.4. *Langevin equation If the particular described above is imbued with a potential energy $V(q)$ at position $q \in \mathbb{R}$, then the dynamics of the system will be described by the following SDE*

$$\begin{aligned} dQ_t &= P_t\, dt \\ dP_t &= [-\lambda P_t - V'(Q_t)]\, dt + \sigma dW_t, \end{aligned}$$

for parameters $\lambda, \sigma > 0$. The particle is then characterised by a position Q_t and a momentum P_t. This is a multidimensional SDE which can be written as

$$dX_t = b(X_t)\, dt + \Sigma\, dW_t,$$

for $X_t = (q_t, p_t)^\top$, where

$$b((q,p)) = \begin{pmatrix} p \\ -\lambda p - V(q) \end{pmatrix},$$

and

$$\Sigma = \begin{pmatrix} 0 & 0 \\ 0 & \sigma \end{pmatrix}$$

An interesting feature of this process is that the noise acts only on the momentum. We shall revisit properties of this process later on.

Example 4.5. *Duffing-van der Pol equation Consider the following SDE*

$$\begin{aligned} dQ_t &= P_t\, dt \\ dP_t &= [-P_t(\lambda + Q_t^2) + (\alpha Q - Q^3)]\, dt + \sigma Q_t\, dW_t, \end{aligned}$$

which comprises a nonlinear dissipation with parameter λ, and a conservative force arising from the potential $V(q) = q^4/4 - \alpha q^2/2$ where α is a tilting parameter.

Unlike the previous two examples, this SDE has *multiplicative* noise, i.e. the noise term depends on the state of the process.

4.9 Numerical methods for Itô Diffusions

While there are a few examples where SDEs can be solved explicitly, like ODEs these tend to be quite exceptional. In general, for nonlinear drift or diffusion functions, the explicit solution will not be available and one must resort to numerical techniques. In this section we shall examine numerical approximations X_n of the solution $X(t_n)$ of an SDE

$$dX_t = b(X_t)\, dt + \sigma(X_t)\, dW_t, \tag{4.9}$$

where $t_n = n\Delta t$ for some $n \in \mathbb{N}$.

4.9.1 The Euler-Maruyama Scheme

A very simple approach is the Euler-Maruyama method, whose derivation we now sketch. Given a time interval $[t_n, t_{n+1}]$, since X_t is markovian

$$X(t_{n+1}) = X(t_n) + \int_{t_n}^{t_{n+1}} b(X(s))\,ds + \int_{t_n}^{t_{n+1}} \sigma(X(s))\,dW_s.$$

Assuming that σ and b are sufficiently smooth so that they do not vary to much over $[t_n, t_{n+1}]$, we would feel justified in making the following approximation:

$$\begin{aligned} X_{n+1} &= X_n + b(X_n)\int_{t_n}^{t_{n+1}} + \sigma(X_n)\int_{t_n}^{t_{n+1}} dW_s \\ &= X_n + b(X_n)\,\Delta t + \sigma(X_n)\Delta W_n \end{aligned}$$

where $\Delta W_n = W(t_{n+1}) - W(t_n) \sim \mathcal{N}(0, \Delta t I)$. Note that we approximate $\sigma(X_s)$ with $\sigma(X(t_n))$ consistent with the definition of the Itô integral. This is the Euler-Maruyama method:

Definition 4.18 (Euler-Maruyama method)**.** *For time step $\Delta t > 0$ and initial condition X_0, the Euler-Maruyama approximation X_n to $X(n\Delta t)$ is defined by*

$$X_{n+1} = X_n + b(X_n)\,\Delta t + \sigma(X_n)\Delta W_n,$$

where $\Delta W_n = \int_{t_n}^{t_{n+1}} dW(r) = W(t_{n+1}) - W(t_n)$ and W is a standard Brownian motion.

The Gaussian increments are IID and easy to sample. A typical algorith for generating a sample path $X_0, \ldots, X_N$ approximating $X(0), \ldots, X(t_N)$ using the Euler-Maruyama discretisation is given by:

Euler-Maruyama Approximation

```
Set $X_0$ to be initial state and let $\Delta t > 0$,
    1. For $n = 1$ to $N$:
        a) Sample $\xi \sim \mathcal{N}(0, I)$.
        b) Set $X_{n+1} = X_n + b(X_n)\Delta t + \sqrt{\Delta t}\sigma(X_n)\xi$.
```

4.9.2 The Milstein scheme

The idea of Milstein's method is to improve on the approximation

$$\int_{t_n}^{t_{n+1}} \sigma(X_u)\,dW_u \approx \sigma(X_n)\int_{t_n}^{t_{n+1}} dW_u.$$

To do this, we use the Itô formula to get an expression for $g(X_u)$. We will focus on the case where X_t is a scalar diffusion. We shall use the following lemma

Lemma 4.14. *Let $\Delta t > 0$ and $n \in \mathbb{N}$. Then we have the following result:*

$$\int_{n\Delta t}^{(n+1)\Delta t} \int_{n\Delta t}^{s} dW_u \, dW_s = \frac{1}{2}(\Delta W_n^2 - \Delta t),$$

where $\Delta W_n = W_{(n+1)\Delta t} - W_{n\Delta t}$.

Proof. Let $f(x) = x^2$, then applying Itô's formula to $f(W_t)$:

$$dW_t^2 = 2W_t \, dW_t, +dt,$$

which implies that

$$\int_{n\Delta t}^{(n+1)\Delta t} W_t \, dW_t = \frac{1}{2}\left(\int_{n\Delta t}^{(n+1)\Delta t} dW_t^2 - \int_{n\Delta t}^{(n+1)\Delta t} dt\right) = \frac{1}{2}\left(W_{(n+1)\Delta t}^2 - W_{n\Delta t}^2 + \Delta t\right),$$

as required. Furthermore, we have that

$$\begin{aligned}
\int_{n\Delta t}^{(n+1)\Delta t} \int_{n\Delta t}^{s} dW_u \, dW_s &= \int_{n\Delta t}^{(n+1)\Delta t} W_s \, dW_s - W_{n\Delta t}(W_{(n+1)\Delta t} - W_{n\Delta t}) \\
&= \frac{1}{2}W_{(n+1)\Delta t}^2 + \frac{1}{2}W_{n\Delta t}^2 - \frac{1}{2}\Delta t - W_{n\Delta t}W_{(n+1)\Delta t} \\
&= \frac{1}{2}(\Delta W_n^2 - \Delta t).
\end{aligned}$$

□

As before a single increment of the solution X_t from t_n to t_{n+1} is given by

$$X_{t_{n+1}} = X_{t_n} + \int_{t_n}^{t_{n+1}} b(X_s) \, ds + \int_{t_n}^{t_{n+1}} \sigma(X_s) \, dW_s. \tag{4.10}$$

Applying Itô's formula to the drift and diffusion coefficients we obtain

$$b(X_s) = b(X_{t_n}) + \int_{t_n}^{s} \mathcal{L}b(X_u) \, du + \int_{t_n}^{s} b'\sigma(X_u) \, dW_u,$$

and

$$\sigma(X_s) = \sigma(X_{t_n}) + \int_{t_n}^{s} \mathcal{L}\sigma(X_u) \, du + \int_{t_n}^{s} \sigma'\sigma(X_u) \, dW_u,$$

where $\mathcal{L}f(x) = b(x)f'(x) + \frac{1}{2}\sigma(x)f''(x)$ is the generator of the SDE. Substituting these formulas into (4.10) writing $X_k = X_{k\Delta t}$, to get

$$\begin{aligned}
X_{n+1} = {} & X_n + b(X_n)\Delta t + \sigma(X_n)\Delta W_n \\
& + \int_{t_n}^{t_{n+1}} \int_{t_n}^{s} \mathcal{L}b(X_u) \, du \, ds + \int_{t_n}^{t_{n+1}} \int_{t_n}^{s} (b'\sigma)(X_u) \, dW_u \, ds \\
& + \int_{t_n}^{t_{n+1}} \int_{t_n}^{s} \mathcal{L}\sigma(X_u) \, du \, dW_s + \int_{t_n}^{t_{n+1}} \int_{t_n}^{s} \sigma'\sigma(X_u) \, dW_u \, dW_s.
\end{aligned}$$

Up to now we have not made any approximations. We now discard any terms which are higher than Δt. To do so, we note that for $\alpha, \beta \geqslant 0$, we have $(\Delta t)^{\alpha}(\Delta W_n)^{\beta} = O((\Delta t)^{\alpha+\beta/2})$. Applying Lemma 4.14 we have that

$$\begin{aligned} X_{n+1} &= X_n + b(X_n)\Delta t + \sigma(X_n)\Delta W_n + (\sigma'\sigma)(X_n)\int_{t_n}^{t_{n+1}}\int_{t_n}^{s} dW_u\, dW_s + o(\Delta t) \\ &\approx X_n + b(X_n)\Delta t + \sigma(X_n)\Delta W_n + \frac{1}{2}(\sigma'\sigma)(X_n)(\Delta W_n^2 - \Delta t), \end{aligned}$$

where $\Delta W_n^2 = \left(W_{(n+1)\Delta t} - W_{n\Delta t}\right)^2$. This leads to the approximating process defined on $0 = t_0 < t_1 < \ldots < t_N = T$, where $t_{n-1} - t_n = \Delta t$, by

$$X_{n+1} = X_n + b(X_n)\Delta t + \sigma(X_n)\Delta W_n + \frac{1}{2}(\sigma'\sigma)(X_n)(\Delta W_n^n - \Delta t).$$

Noting that

$$W_{(n+1)\Delta t} - W_{n\Delta t} \sim \mathcal{N}(0, \Delta t),$$

we have that ΔW_n^2 has the same distribution as ξ_n^2 where $\xi_n \sim \mathcal{N}(0,1)$. The Milstein algorithm to approximate a scalar diffusion process is thus given as follows:

Milstein Approximation

Set X_0 to be initial state and let $\Delta t > 0$,

1. For $n = 1$ to N:

   a) Sample $\xi \sim \mathcal{N}(0,1)$.

   b) Set $X_{n+1} = X_n + b(X_n)\Delta t + \sqrt{\Delta t}\sigma(X_n)\xi + \frac{1}{2}\Delta t(\sigma'\sigma)(X_n)(\xi^2 - 1)$.

Note that for processes with additive noise (i.e. constant diffusion coefficient), the milstein approximation reduces to the Euler Maruyama approximation. It is possible to derive an analogous Milstein approximation for multivariate processes. In general, however, one will have to deal with *Lévy area* terms of the form

$$A_{ij} = \int_s^t \int_s^r dW_i(p)dW_j(r) - \int_s^t \int_s^r dW_j(p)dW_i(r).$$

which arise from non-diagonal terms in the diffusion tensor σ. These cannot be handled in the same manner we dealt with the iterated integral that arose in the scalar version. However, in very specific cases, sampling of these Levy areas A_{ij} can be avoided. For example, if the diffusion tensor $\sigma(x)$ is a diagonal matrix, then we have that

$$X_{n+1} = X_n + b(X_n)\Delta t + \sigma(X_n)\Delta W_n + \frac{1}{2}\sum_{k=1}^{d}\frac{\partial \sigma_{k,k}}{\partial x_k}(X_n)\sigma_{k,k}(X_n)(\Delta W_{k,n}^2 - \Delta t).$$

4.10 Discretisation Error

Unlike the methods we discussed for Gaussian processes, the discretisation schemes that we have introduced for diffusion processes are not exact. In particular the distribution of the random vector $(X_0, X_1, \ldots, X_N)$ will be different from that of $(X_0, X_{\Delta t}, \ldots, X_{\Delta N})$, although we would expect that in some sense, the difference vanishes as $\Delta t \to 0$. Since both the exact and numerical approximations are random, different means of quantifying the error can be considered.

4.10.1 Strong Error

Let X_t be the solution of the SDE (4.9), and let $\hat{X}_n$ be a numerical approximation to X_t, using the same Brownain motion as X_t does. The strong error of the approximation $\hat{X}_n$ at time $N\Delta t$ is given by

$$e_{strong} = \mathbb{E}|X_{N\Delta t} - \hat{X}_N|,$$

for N sufficiently large. For a fixed realisation of the Brownian motion, both X_t and $\hat{X}_n$ are deterministic processes, and $|X_{N\Delta t} - \hat{X}_N|$ measures the distance between the two solutions after N steps. The strong error e_{strong} then averages this distance over all realisations of Brownian motion W_t. We say that a numerical approximation $\hat{X}_n$ has *strong order r error* if, for all $N > 0$ there exists $\delta = \delta(N)$ and a constant $K = K(N, \delta)$ such that for $\Delta t \leqslant \delta$:

$$\mathbb{E}|\hat{X}_n - X_{n\Delta t}| \leqslant K(\Delta t)^r, \quad \forall n \leqslant N.$$

Let $\hat{X}_n^{EM}$ be the Euler-Maruyama approximation given above, with step size Δt. Under appropriate assumptions on the drift b and diffusion coefficient σ, given $N > 0$, there is a constant $K > 0$ such that

$$e_{strong}^{EM} = \mathbb{E}|\hat{X}_n^{EM} - X_{n\Delta t}| \leqslant K\sqrt{\Delta t}, \quad n \leqslant N,$$

for all sufficiently small Δt, so that the Euler-Maruyama approximation has strong order $\frac{1}{2}$ error.

Similarly, let X_n^{MIL} be the Milstein approximation, with step size Δt. Then, under appropriate conditions on the drift b and diffusion coefficient σ, given $N > 0$, there exists $K > 0$ such that

$$e_{strong}^{MIL} = \mathbb{E}|\hat{X}_n^{MIL} - X_{n\Delta t}| \leqslant K\Delta t, \quad n \leqslant N$$

for all sufficiently small Δt. Thus, the Miltein scheme has strong order 1 error.

4.10.2 Weak Error

In some applications we are less interested in the accuracy of the paths of $\hat{X}_n$ compared to $X_{n\Delta t}$ and more interested that distribution of $\hat{X}_n$ is accurate. This motivates the concept of *weak error*. As before, let X_t be the solution of (4.9) and let $\hat{X}_n$ be a numerical approximation of X_t. Let $\mathcal{F}$ be a class of functions from $\mathbb{R}^d \to \mathbb{R}$, typically the set of all bounded, twice differentiable functions. Then the error in the distribution $\hat{X}_n$ can be quantified by

$$e_{weak}(f) = \left|\mathbb{E}[f(\hat{X}_n)] - \mathbb{E}[f(X_{\Delta tn})]\right|,$$

for all $f \in \mathcal{A}$, and sufficiently small Δt. The supremum of this quantity

$$e_{weak} = \sup_{f \in \mathcal{A}} e_{weak}(f),$$

is called the *weak error* of $\hat{X}_n$ at time $n\Delta t$. If the weak error is small, the distribution of the numerical solution is close to the distribution of the exact solution. We say that a numerical approximation $\hat{X}_n$ to $X_{\Delta tn}$ has weak error order r if, for all $N \in \mathbb{N}$, there exists $K > 0$ and $\delta > 0$ such that

$$\sup_{f \in \mathcal{A}} \left| \mathbb{E}[f(\hat{X}_n)] - \mathbb{E}[f(X_{n\Delta t})] \right| \leqslant K(\Delta t)^r \quad \forall n \leqslant N.$$

Let $\hat{X}_n^{EM}$ be the Euler-Maruyama approximation with step-size Δt. Then under appropraite conditions on the drift and diffusion coefficients, and $\mathcal{A}$, there exists $K > 0$ such that

$$e_{weak}^{EM} = \sup_{f \in \mathcal{A}} \left| \mathbb{E} f(\hat{X}_n^{EM}) - \mathbb{E} f(X_{n\Delta t}) \right| \leqslant K\Delta t,$$

for all $f \in \mathcal{A}$ and sufficiently small Δt. Similarly, the Milstein approximation X_n^{MIL} satifies

$$e_{weak}^{MIL} = \sup_{f \in \mathcal{A}} \left| \mathbb{E}[f(\hat{X}_n^{MIL})] - \mathbb{E}[f(X_{n\Delta t})] \right| \leqslant K\Delta t,$$

for all $f \in \mathcal{A}$ and sufficiently small Δt.

4.10.3 An explicit computation of the error

As an explicit demonstration of the above error estimates, let's focus on a specific diffusion process, namely Geometric Brownian motion given by

$$dX_t = \lambda X_t \, dt + \sigma X_t \, dW_t, \tag{4.11}$$

where λ and σ are constants. The benefit of considering this SDE is that we can solve it explicitly. Indeed, we have the following result.

Lemma 4.15. *The solution X_t of the geometric Brownian motion defined by (4.11) is given by*

$$X_t = X_0 \exp\left(\left(\lambda - \frac{\sigma^2}{2} \right) t + \sigma W_t \right).$$

Proof. This can be shown immediately by applying Itô's formula to the function $\log X_t$. Details of the proof are left as an exercise. □

Consider the Euler-Maruyama discretisation of (4.11) given by $\hat{X}_n$, so that

$$\hat{X}_{n+1} = \hat{X}_n + \lambda \hat{X}_n \Delta t + \sigma \hat{X}_n \Delta W_n,$$

where $\Delta W_n = W_{(n+1)\Delta t} - W_{n\Delta t}$. We can rewrite this as

$$\hat{X}_{n+1} = (1 + \lambda \Delta t + \sigma \Delta W_n) \hat{X}_n,$$

or equivalently,

$$\hat{X}_n = \prod_{i=0}^{n-1} (1 + \lambda\Delta t + \sigma\Delta W_i)\, \hat{X}_0.$$

Consider now

$$\begin{aligned}\mathbb{E}\left|\hat{X}_n - X_{n\Delta t}\right| &= \mathbb{E}\left|\prod_{i=0}^{n-1} (1 + \lambda\Delta t + \sigma\Delta W_i)\, \hat{X}_0 - e^{(\lambda-\sigma^2/2)n\Delta t+\sigma W_{n\Delta t}}\right| \\ &= \mathbb{E}\left|\prod_{i=0}^{n-1} (1 + \lambda\Delta t + \sigma\Delta W_i)\, \hat{X}_0 - \prod_{i=0}^{n-1} e^{(\lambda-\sigma^2/2)\Delta t+\sigma\Delta W_i}\right|\end{aligned}$$

Taylor expanding $e^{(\lambda-\sigma^2/2)\Delta t+\sigma\Delta W_i}$ up to $(\Delta t)^2$:

$$\begin{aligned}e^{(\lambda-\sigma^2/2)\Delta t+\sigma\Delta W_i} = 1 &+ \left[\left(\lambda - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\Delta W_i\right] \\ &+ \frac{1}{2}\left[\left(\lambda - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\Delta W_i\right]^2 \\ &+ \frac{1}{6}\left[\left(\lambda - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\Delta W_i\right]^3 + \ldots \\ =&1 + \left[(\lambda - \sigma^2/2)\Delta t + \sigma\Delta W_i\right] + (\lambda - \sigma^2/2)\sigma\Delta t\Delta W_i \\ &+ \frac{1}{2}\sigma^2[\Delta W_i]^2 + \frac{1}{6}\sigma^3[\Delta W_i]^3 + O(\Delta)^2.\end{aligned}$$

From the properties of the quadratic variation of W_t, we have that, locally $\Delta W_i \Delta W_i = \Delta t$, it follows that:

$$e^{(\lambda-\sigma^2/2)\Delta t+\sigma\Delta W_i} = 1 + \lambda\Delta t + \sigma\Delta W_i + (\lambda - \sigma^2/2)\sigma\Delta t\Delta W_i + \frac{1}{6}\sigma^2[\Delta W_i]^3 + O(\Delta t)^2.$$

It follows that

$$\begin{aligned}\prod_{i=0}^{n-1}[1 + \lambda\Delta t + \sigma\Delta W_i] &= \prod_{i=0}^{n-1}\left[e^{(\lambda-\sigma^2/2)\Delta t+\sigma\Delta W_i} - (\lambda - \sigma^2/2)\sigma\Delta t\Delta W_i - \frac{1}{6}\sigma^3[\Delta W_i]^3\right] \\ &= e^{(\lambda-\sigma^2/2)n\Delta t+\sigma W_{n\Delta t}} + nO(\Delta t\Delta W) + nO(\Delta W)^3 + nO(\Delta t)^2.\end{aligned}$$

Therefore, the strong error is

$$\mathbb{E}\left|nO(\Delta t\Delta W) + nO(\Delta W)^3 + nO(\Delta t)^2\right| = \mathbb{E}\left|\frac{T}{\Delta t}O(\Delta t\Delta W) + \frac{T}{\Delta t}O(\Delta W)^3 + \frac{T}{\Delta t}O(\Delta t)^2\right|,$$

but $\frac{1}{\Delta t}O(\Delta t\Delta W)$ is $O(\Delta t)^{1/2}$, which is the dominant term in sum. Therefore, the EM scheme has strong error order $1/2$.

Exercise 4.4. *To demonstrate the weak order of convergence of the Euler-Maruyama scheme, consider*

$$e = \left|\mathbb{E}f(\hat{X}_n) - \mathbb{E}f(X_{n\Delta t})\right|,$$

for the particular case that $f(x) = x$, and show that it is $O(\Delta t)$.

4.10.4 Implicit Discretisation and Stability Analysis

Besides accuracy, another property which is desired from a numerical approximation of an SDE is *stability*, namely that if the solution of the original SDE remains bounded for all time, then so does the numerical approximation. An unstable numerical scheme can result in the discrete approximation "exploding", particularly, if the step size Δt is sufficiently large. Ensuring stability can impose strong constraints on the maximum value of the step size Δt, which, for *stiff* problems motivates the use of implicit schemes which are unconditionally stable.

A good illustration of the problems arising from stability can be seen when considering Geometric Brownian motion

$$dX_t = \lambda X_t + \sigma X_t \, dW_t.$$

In Figure 4.1 we plot the trajectories of a single path, with parameters $\lambda = -10$ and $\sigma = 4$ using an Euler-Maruyama scheme with timesteps $\Delta t = 0.25$ and $\Delta t = 0.00125$. We see that for the large timestep, the numerical approximation of the solution becomes very large, while for the small timestep the solution remains stable.

Suppose that $X_0 \neq 0$ is a constant. Consider the mean-square of the exact solution:

$$\mathbb{E}[X_t^2] = e^{(2\lambda+\sigma^2)t} X_0^2.$$

Clearly, $\mathbb{E}[X_t^2] \to 0$ as $t \to \infty$, if and only if

$$2\lambda + \sigma^2 < 0.$$

Our question is thus: what conditions must λ and σ^2 satisfy for a numerical discretisation to satify $\lim_{n\to} \mathbb{E}[|\hat{X}_n^2|] \to 0$. Clearly, the constraint will depend on the time-step Δt.

Definition 4.19. *A stochastic process is said to be* mean-square stable *if* $\lim_{n\to\infty} \mathbb{E}|\hat{X}_n|^2 \to 0$. *The set of parameters such that this condition holds is called the* stability region.

As an example, condition the Euler-Maruyama approximation. As discussed above, we can write the Euler Maruyama approxiation of Geometric Brownian motion as

$$\hat{X}_n = \prod_{i=0}^{n-1} (1 + \lambda\Delta t + \sigma\Delta W_i) X_0.$$

The second moment of $\hat{X}_n$ is

$$\begin{aligned}\mathbb{E}[\hat{X}_n^2] &= \mathbb{E}\left[\prod_{i=0}^{n-1}(1+\lambda\Delta t + \sigma\Delta W_j)\right]^2 X_0^2 \\ &= \prod_{i=0}^{n-1} \mathbb{E}\left[(1+\lambda\Delta t+\sigma\Delta W_j)^2\right] X_0^2.\end{aligned}$$

Now

$$\mathbb{E}\left[(1+\lambda\Delta t+\sigma\Delta W_j)^2\right] = 1 + 2\lambda\Delta t + \lambda^2(\Delta t)^2 + \sigma^2\Delta t = 1 + \Delta t(2\lambda + \sigma^2 + \Delta t\lambda^2).$$

For $\mathbb{E}|X_n|^2$ to converge to zero, we require $\mathbb{E}\left[(1 + 2\lambda\Delta t + \sigma\Delta W_j)^2\right] < 1$ or equivalently:

$$2\lambda + \sigma^2 + \Delta t\lambda^2 < 0.$$

This condition is more restrictive than the stability condition $2\lambda + \sigma^2 < 0$ for the true solution X_t. Thus, to achieve stability of the Euler-Maruyama approximation, we must choose the time-step so that

$$0 < \Delta t < \frac{-2(r + \sigma^2/2)}{\lambda^2}.$$

Exercise 4.5. *Repeat the above argument to identify the stability region for the Milstein-Scheme, and thus identify conditions for which the scheme is mean-square stable.*

In many applications, this region is often far to restrictive. As in the case for ODEs, this motivates the use of so-called *implicit* schemes. As an example, consider the following implicit version of the Euler-Maruyama method, known as the θ-Euler-Maruyama method

Definition 4.20. *Consider a time-step $\Delta t > 0$ and initial condition $X_0 \in \mathbb{R}$. The θ-Euler-Maruyama approximation X_n of $X(\Delta tn)$ is given by*

$$X_{n+1} = X_n + [(1-\theta)b(X_n) + \theta b(X_{n+1})]\Delta t + \sigma(X_n)\Delta W_n,$$

where $\theta \in [0, 1]$ is a parameter which controls the degree of implicitness in the drift term.

Thus, in this scheme, the diffusion term is treated explicitly, while the drift term is treated implicitly according to the parameter θ. When implementing this scheme, given X_n, one must solve a nonlinear equation to obtain X_{n+1}, typically using a Newton-Raphson scheme which is iterative. Thus, in terms of computational cost, we expect the θ method to be far more expensive.

Exercise 4.6. *Repeat the above stability calculation for the solution X_n of the θ-Euler-Maruyama approximation to geometric Brownian motion and show that when*

$$2\lambda + \sigma^2 + \Delta t(1 - 2\theta)\lambda^2 < 0,$$

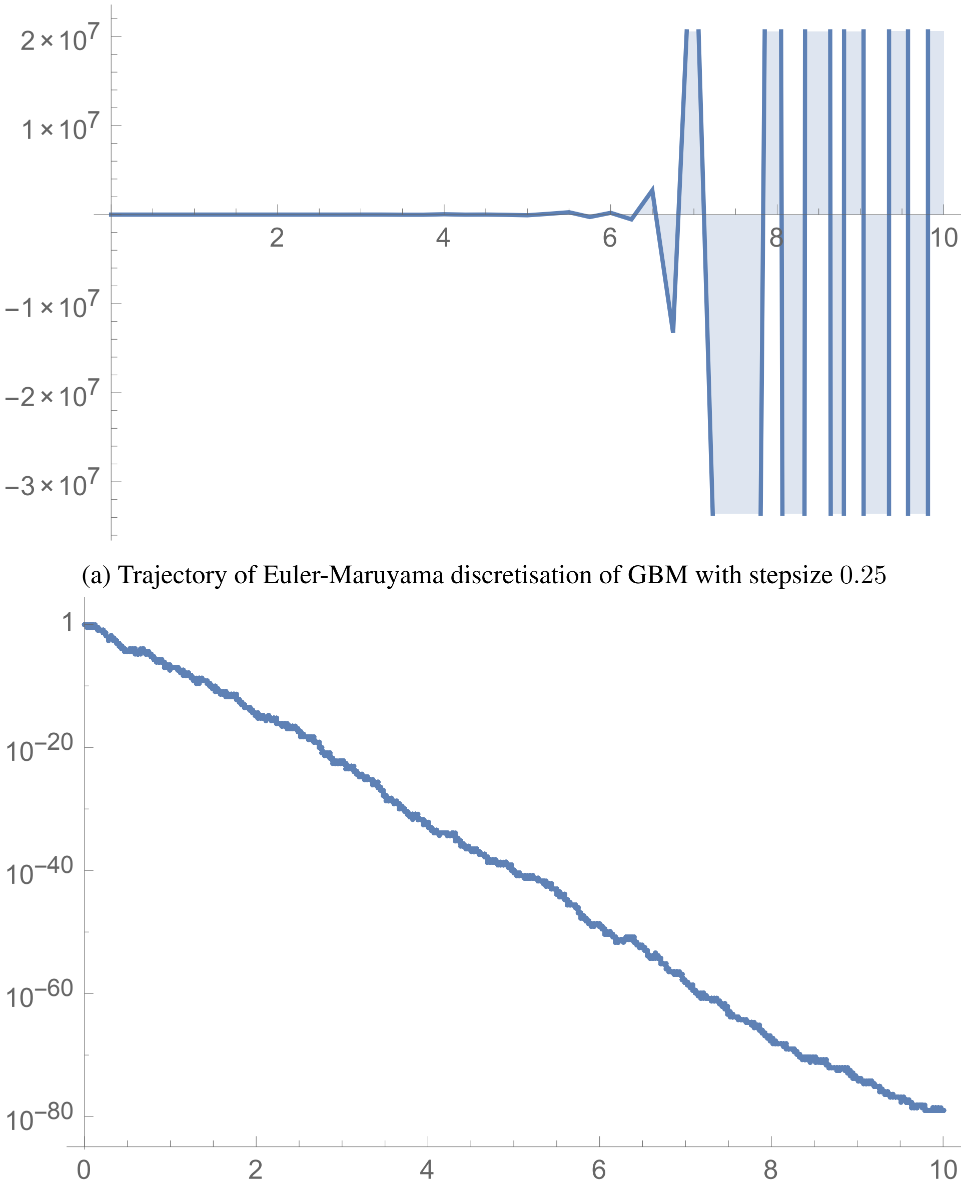
the scheme is mean-square stable. In particular, if $\theta = 1/2$, the stability condition reduces to

$$2\lambda + \sigma^2 < 0,$$

independent of the choice of time-step and identical to the stability condition of geometric Brownian motion.

The above exercise was worked out in class

Exercise 4.7. *Repeat the above stability calculation for the solution X_n of the Milstein approximation to geometric Brownian motion.*



(a) Trajectory of Euler-Maruyama discretisation of GBM with stepsize 0.25

(b) Trajectory of Euler-Maruyama discretisation of GBM with stepsize 0.00125

Figure 4.1: Stability of the Euler-Maruyama discretisation of Geometric Brownian motion

Chapter 5

Further topics: Non-Examinable

5.1 Monte Carlo Estimates of SDEs

An important application of these numerical approximations of SDEs is to generate Monte Carlo estimates of statistical quantities which depend on the solution of a given SDE. More specifically, given an SDE

$$dX_t = b(X_t)\,dt + \sigma(X_t)\,dW_t,$$

many applications often involve computing averages of observables of the solution X_t over a time interval $[0, T]$. These observables can either

1. Depend on X_t at a particular time, such as for example $\mathbb{E}[f(X_T)]$,
2. Depend on an entire path of the process, for example $\mathbb{E}\left[\int_{T_0}^{T} g(X_s)\,ds\right]$, for some function g and $T_0 \leqslant T$, or for example $\mathbb{E}\left[\sup_{t\in[0,T]} |X_t|\right]$.

Both cases can be expressed as $I = \mathbb{E}[F(X.)]$, where F is a function of the path X_t over $[0, T]$. In order to estimate this quantity using Monte Carlo integration, we generate independent realisations $\hat{X}^{(1)}$, $\hat{X}^{(2)}$, …, $\hat{X}^{(N)}$ of the solution X_t using a numerical approximation. Each realisation $\hat{X}^{(i)}$ is a time-series with $\lfloor T/\Delta t \rfloor$ values. We then use the Monte Carlo estimator:

$$\hat{I}_N^n = \frac{1}{N}\sum_{j=1}^{N} F(\{\hat{X}_k^{(j)}\}_{k=1,\dots,n}).$$

To make things clearer, let's focus on the case when $F(X) = f(X_T)$, for some function f. As with all Monte-Carlo methods, we can get arbitrarily accurate results provided we are willing to expend sufficient computational effort. What is different in this scenario, is that while for the standard Monte-Carlo methods described in Chapter 2 the only error was statistical error (i.e. due to the variance of the fluctuations around the mean), in this case, there is also the discretisation error of the numerical approximation used. As we discussed in Section 2.4 a natural measure of accuracy of an estimator is the mean square error. In question 6 of Problem sheet 1 you had considered an estimator for the density of a distribution, and using MSE studied the tradeoff between bias and variance, and based on this found an optimal choice of bin-size h.

Let's perform a similar analysis for the estimator $\hat{I}_N^n$. As we've seen before, the mean square error of an estimator $\hat{I}_n$ can be decomposed into the variance and the square of the bias. More specifically

$$\text{MSE}(\hat{I}_N^n) = \mathbb{E}\left[\left(\hat{I}_N^n - \mathbb{E}[f(X_T)]\right)^2\right] = \text{Var}[\hat{I}_N^n] + \text{Bias}(\hat{I}_N^n)^2.$$

Using the fact that the realisations of the chain are IID, the variance is given by

$$\text{Var}[\hat{I}_N^n] = \frac{1}{N}\text{Var}[f(\hat{X}_N)].$$

The bias is given by

$$\left|\text{bias}(\hat{I}_N^n)\right| = \left|\mathbb{E}[\hat{I}_N^n] - \mathbb{E}[f(X_T)]\right| = e_{\text{weak}}(f),$$

i.e. the weak error of the numerical approximation $\hat{X}_n$ of X_T, for the observable f. Therefore we have that

$$\text{MSE}(\hat{I}_N^n) = \frac{\sigma^2}{N} + e_{\text{weak}}(f)^2,$$

where $\sigma^2 = \text{Var}[f(\hat{X}_N)]$.

The first term of the MSE corresponds to the *Monte-Carlo error* and will go to zero as N increases. The second term is the *discretization error*, and will go to zero as $\Delta t = \lfloor T/n \rfloor \to 0$. On the other hand, the total cost of computing a single approximation $\hat{I}_N^n$ is $O(nN)$. Here we see the tradeoff between the bias and the variance. Suppose we have a fixed computational budget K, then increasing n (to decrease discretization error) means we must decrease N (thus increasing Monte Carlo error) proportionally to maintain the same computational cost, and vice versa.

It would be nice if we could identify an "optimal" choice of n based on N, like we did in Question 6 of Problem Sheet 1 for the kernel density estimator. Suppose that we are using an Euler-Maruyama discretisation to approximate X_t, which has weak error of order 1, and assume that we can write

$$e_{\text{weak}}(f) \approx C\Delta t = \frac{K_f T}{n},$$

for some constant $K_f > 0$. The mean-square error can then be written as

$$\text{MSE}(\hat{I}_N^n) \approx \frac{\sigma^2}{N} + \left(\frac{K_f T}{n}\right)^2.$$

Assume also that the computational cost for computing $\hat{I}_N^n$ is approximately

$$C(N, n) \approx CNn.$$

Suppose we have a fixed computational budget W and we wish to choose N and n accordingly to minimize the mean square error, i.e. we wish to find a solution to the following constrained optimisation problem

$$\text{Minimise } \frac{\sigma^2}{N} + \left(\frac{K_f T}{n}\right)^2,$$

subject to $C(N,n) = W$. We can find the optimal choice of n using standard calculus by introducing a lagrange multiplier λ. We solve

$$\partial_N \left(\frac{\sigma^2}{N} + \left(\frac{K_f T}{n}\right)^2 - \lambda(CNn - W) \right) = 0$$
$$\partial_n \left(\frac{\sigma^2}{N} + \left(\frac{K_f T}{n}\right)^2 - \lambda(CNn - W) \right) = 0,$$

for the optimal values of N and n so that [1]

$$\frac{\sigma^2}{N^2} = \lambda C n,$$

and

$$\frac{2K_f T}{n^3} = \lambda C N.$$

At the optimal value, using the constraint $W = CNn$ we have

$$\frac{2(K_f T)^2}{W n^2} = \frac{\sigma^2}{N},$$

so that

$$n^2 = 2\frac{(K_f T)^2}{W\sigma^2} N.$$

so that the optimal scaling for N and n in terms of W is

$$n \propto W^{1/3} \quad \text{and} \quad N \propto W^{2/3}.$$

Therefore, assuming that the parameters σ^2 and K_f are both of order 1 the optimal computational effort is determined by choosing n to be roughly $\sqrt{N}$. Of course, while we can easily approximate σ^2 numerically, it is not so easy to get accurate estimates of the bias, and so, this approximation should only be considered a rule of thumb.

5.2 Variance Reduction methods for SDEs

Suppose we wish to use Monte Carlo simulation to approximate $\mathbb{E}[F(X.)]$, where F is an observable of the path of the solution of the SDE X_t, over $[0,T]$. When trying to measure quantities which involve rare events, then using the standard Monte Carlo estimator we outlined in the previous section might be prohibitively expensive to procude a sufficiently accurate approximation. Consider the following example.

Example 5.1. *Given the process $X_t = W_t$, i.e. a standard Brownian motion, suppose we wish to compute the probability that X_t exceeds $c > 0$ within the time $[0,1]$, i.e.*

$$I = \mathbb{P}\left[\sup_{t\in[0,1]} X_t > c \right],$$

[1] note that we assume these are continuous quantities. We expect that the integer part of the optimal values will not be too far off.

where $c > 0$. Then by the reflection principle,

$$\mathbb{P}\left[\sup_{t\in[0,1]} X_t > c\right] = 2\mathbb{P}\left[W_t \geqslant c\right] = 2\left(1 - \Phi\left(c\right)\right) \leqslant 2e^{-c^2/2}.$$

So that for $c = 5$, the probability is of the order 10^{-6}. Consider a second process Y_t defined by the SDE:

$$Y_t = a\,dt + W_t,$$

where $a > 0$ is a constant. Computing the hitting probability for Y_t, is given by

$$I_a = \mathbb{P}\left[\sup_{t\in[0,1]} Y_t > c\right] = \int_0^t \frac{c}{\sqrt{2\pi s^3}} \exp\left(-\frac{(c-as)^2}{2s}\right) ds,$$

so that for $a = 1$, $I_a = 5\cdot 10^{-5}$, $a = 2$, $I_a = 2\cdot 10^{-3}$, for $a = 5$, $I_a = 0.539$, etc.

Thus we would be able to compute the hitting probability of Y_t much more efficiently. If possible we would like to implement some form of importance sampling scheme, to be able to compute I using trajectories of Y_t. Girsanov's theorem provides us with the means to do this, in quite some generality.

Theorem 5.1 (Oksendal Theorem 8.6.5)**.** *Consider the SDEs*

$$dX_t = b(X_t)\,dt + \sigma(X_t)\,dW_t, \tag{5.1}$$

and

$$dY_t = b(Y_t) + \gamma(t,\omega)\,dt + \sigma(Y_t)\,dW_t, \tag{5.2}$$

on the time interval $t \in [0,T]$, where W_t is a standard Brownian motion and where $X_0 = Y_0 = x$. Assuming that there exists a process $u(t,\omega)$ such that

$$\sigma(Y_t)u(t,\omega) = \gamma(t,\omega),$$

and that $u(t,\omega)$ satsifies Novikov's condition

$$\mathbb{E}\left[\exp\left(\frac{1}{2}\int_0^T u^2(s,\omega)\,ds\right)\right] < \infty.$$

For $t \leqslant T$ define

$$M_t = \exp\left(-\int_0^t u(s,\omega)\,d\tilde{W}_s - \frac{1}{2}\int_0^t u^2(s,\omega)\,ds\right),$$

where $\tilde{W}_s$ is a Brownian motion with respect to the measure $\mathbb{Q}$, then

$$d\mathbb{P}(\omega) = M_T(\omega)d\mathbb{Q}(\omega),$$

and so we have that

$$E[F(X)] = \mathbb{E}\left(F(Y)M_T\right). \tag{5.3}$$

Girsanov's theorem provides us with a basic tool for importance sampling. To compute (5.3) numerically we would approximate

$$\int_0^t u(s,\omega)\, d\tilde{W}_s \approx \sum_{n=0}^{N} u(s_n,\omega)\Delta\tilde{W}_n,$$

and

$$\int_0^t u^2(s,\omega)\, ds \approx \sum_{n=0}^{N} u^2(s_n,\omega)\Delta t,$$

which would itself contribute additional discretisation error to the estimator, but which can also be controlled.

Example 5.2. *Let's return to Example 5.1. Then in this case, applying Girsanov theorem:*

$$\mathbb{E}[F(X.)] = \mathbb{E}[F(Y.)M_T],$$

where

$$M_T = \exp\left(-a\tilde{W}_T - \frac{a^2T}{2}\right)$$

5.3 Inference for Stochastic Differential Equations

Once a stochastic model for a given physical system has been derived, we must choose the parameters such that the output of the stochastic model agrees with the observed data. In this section, we present some simples techniques for estimating the diffusion coefficient and parameters in SDEs. As usual, we shall focus on the one-dimensional case. We shall consider the following one dimensional Itô SDE of the form:

$$dXt = b(X_t;\theta)\, dt + \sigma(X_t;\theta)\, dW_t, \quad X_0 = x, \tag{5.4}$$

where $\theta \in \Theta \subset \mathbb{R}^N$ is a finite set of parameters that we want to estimate from the observations. The initial conditions can be taken to be either deterministic or random. We assume that we are provided with observations of the path of the process. This can be either be:

1. Discrete observations $X_{t_0}, X_{t_1}, \ldots, X_{t_N}$, or
2. The entire path X_t, $t \in [0,T]$.

Some simple examples

- The Ornstein-Uhlenbeck process with unknown drift coefficient α:

$$dX_t = -\alpha X_t\, dt + dW_t.$$

- Brownian motion in a bistable potential, with unknown parameters A, B:

$$dX_t = (AX_t - BX_t^3)\, dt + dW_t.$$

5.3.1 Inferring the diffusion coefficient

In order to estimate parameters in the diffusion coefficient, it is natural to use the *quadratic variation* of the solution X_t of the SDE (5.4):

$$\langle X_t, X_t \rangle := \int_0^t \sigma^2(X_s;\theta)\, ds = \lim_{\Delta t_k \to 0} \sum_{t_k \leqslant t} \left| X_{t_{k+1}} - X_{t_k} \right|^2, \tag{5.5}$$

where the limit is in probability. When the diffusion coefficient is constant, i.e. $\sigma(x;\theta) \equiv \sigma$, the convergence (5.5) is almost sure, i.e.

$$\lim_{n \to +\infty} \sum_{i=1}^{n} \left[X_i T 2^{-n} - X_{(i-1)T2^{-n}} \right]^2 = \sigma^2 T \quad a.s.$$

Therefore, if we fix the length of the observation $[0,T]$, and let the number of observations become infinite, i.e. *the high-frequency limit*, taking $n \to \infty$, we can determine the diffusion coefficient. We prove a slightly simpler result

Proposition 5.2. *Let $\{X_j\}_{j=0}^J$ be a sequence of equidistant observations of*

$$dX_t = b(X_t;\theta)\, dt + \sigma\, dW_t,$$

with timestep $\Delta t = \delta$ and $J\delta = T$ fixed. Assuming that the drift $b(x;\theta)$ is bounded, and define

$$\hat{\sigma}_J^2 = \frac{1}{J\delta} \sum_{j=0}^{J=1} (X_{j+1} - X_j)^2. \tag{5.6}$$

Then

$$\left| \mathbb{E} \hat{\sigma}_J^2 - \sigma^2 \right| \leqslant C(\delta + \delta^{1/2}).$$

In particular,

$$\lim_{J \to +\infty} \left| \mathbb{E} \hat{\sigma}_J^2 - \sigma^2 \right| = 0.$$

Proof. We have that

$$X_{j+1} - X_j = \int_{j\delta}^{(j+1)\delta} b(X_s;\theta)\, ds + \sigma \Delta W_j,$$

where $\Delta W_j = W_{(j+1)\delta} - W_{j\delta} \sim \mathcal{N}(0,\delta)$. We substitute this into (5.6) to obtain

$$\hat{\sigma}_J^2 = \sigma^2 \frac{1}{\delta J} \sum_{j=0}^{J-1} (\Delta W_j)^2 + \frac{2}{\delta J} \sum_{j=0}^{J-1} I_j M_j + \frac{1}{\delta J} \sum_{j=0}^{J-1} I_j^2,$$

where

$$I_j := \int_{j\delta}^{(j+1)\delta} b(X_s;\theta)\, ds,$$

and

$$M_j := \sigma \Delta W_j.$$

Note that $\mathbb{E}\,(\Delta W_n)^2 = \delta$. From the boundedness of $b(x;\theta)$ and using the Cauchy-Schwarz inequality:

$$\mathbb{E}I_j^2 \leqslant \delta \int_{j\delta}^{(j+1)\delta} \mathbb{E}\,(b(X_s;\theta))^2 \, ds \leqslant C\delta^2.$$

Consequently,

$$\begin{aligned}\left|\mathbb{E}\hat{\sigma}_J^2 - \sigma^2\right| &\leqslant \frac{1}{\delta}\mathbb{E}I_j^2 + \frac{2}{\delta}\mathbb{E}|I_j M_j| \\ &\leqslant C\delta + \frac{C}{\delta}\left(\frac{1}{\alpha}\mathbb{E}I_j^2 + \alpha\mathbb{E}M_j^2\right) \\ &\leqslant C(\delta + \delta^{1/2}).\end{aligned}$$

In the above, we used Young's inequality with $\alpha = \delta^{1/2}$. □

5.3.2 Estimating the drift coefficient

From now on, we assume that we have already estimated the diffusion coefficient, so that we just set $\sigma = 1$, so that (5.4) becomes

$$dX_t = b(X_t;\theta)\,dt + dW_t. \tag{5.7}$$

Our objective is to estimate the unknown parameters in the drift $\theta \in \Theta$ from a time-series of observations. As we described in the introduction, we use the *maximum likelihood estimator* (MLE). Let us describe the general intuition of the MLE. Suppose we have N iid observations of a random variable X with probability density function $f(x\,|\theta)$. Define the *likelihood function* is then defined to be

$$L\left(\{x_i\}_{i=1}^N \,|\, \theta\right) = \prod_{i=1}^N f(x_i\,|\theta).$$

The maximum likelihood estimator (MLE) is then

$$\hat{\theta} = \arg\max L(\mathbf{x}\,|\,\theta),$$

with $\mathbf{x} = \{x_i\}_{i=1}^N$.

Example 5.3. *Suppose that $\mathbf{x} = \{x_i\}_{i=1}^N$ are iid samples from a Gaussian $\mathcal{N}(\mu,\sigma^2)$ with unknown parameters μ and σ^2. The likelihood function takes the form*

$$L(\mathbf{x}\,|\mu,\sigma) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left(-\frac{\sum_{i=1}^N (x_i-\mu)^2}{2\sigma^2}\right).$$

The maximum likelihood estimator $\hat{\mu}$ for μ is given by

$$\left(\hat{\mu}, \hat{\sigma^2}\right) = \arg\max_{\mu,\sigma^2} L(\mathbf{x},\mu,\sigma^2)$$

Maximizing with respect to μ:

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^N x_i.$$

Maximizing with respect to σ^2:

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{\mu})^2.$$

We want to derive maximum likelihood estimators for the parameters in the drift of (5.7). The observations will either be

1. a series of discrete equidistant observations of the process X_t, $\{X_{i\Delta t}\}_{i=1}^{N}$, where $N\Delta t = T$, or
2. The entire path X_t, where $t \in [0, T]$.

As a simple demonstration, let us first derive the Maximum Likelihood estimator based on the Euler-Maruyama discretization of (5.7):

$$X_{n+1} - X_n = b(X_n; \theta)\Delta t + \Delta W_n.$$

The distribution function for Brownian motion is

$$p_W^N = \prod_{i=0}^{N-1} \frac{1}{\sqrt{2\pi\Delta t}} \exp\left(-\frac{1}{2\Delta t}(\Delta W_i)^2\right) = \frac{1}{(\sqrt{2\pi\Delta t})^N} \exp\left(-\frac{1}{2\Delta t}\sum_{i=0}^{N-1}(\Delta W_i)^2\right).$$

Similarly, for the law of the discretized process $\{X_n\}_{n=0}^{N-1}$ using the fact that $p(X_{i+1} | X_i) \sim \mathcal{N}(X_i + b_i\Delta t, \Delta t)$, we can write

$$p_X^N = \frac{1}{(\sqrt{2\pi\Delta t})^N} \exp\left(-\sum_{i=0}^{N-1}\left(\frac{1}{2\Delta t}(\Delta X_i)^2 + \frac{1}{2}(b_i)^2\Delta t - b_i\Delta X_i\right)\right).$$

Now we can calculate the ratio of the laws of the two processes, evaluated at the path $\{X_n\}_{n=0}^{N-1}$:

$$\frac{p_X^N}{p_W^N} = \exp\left(-\frac{1}{2}\sum_{i=0}^{N-1} b_i^2\Delta t + \sum_{i=0}^{N-1} b_i\Delta X_i\right).$$

Taking the limit as $N \to \infty$, we get the likelihood:

$$L\left(\{X_t\}_{t\in[0,T]}; \theta, T\right) := \exp\left(\int_0^T b(X_s;\theta)\, dX_s - \frac{1}{2}\int_0^T b(X_s;\theta)^2\, ds\right).$$

You might have recognized this expression before. Indeed, from Girsanov's theorem (previous section) we know that the law of X_t, denoted by $\mathbb{P}_X$ is absolutely continuous with respect to Brownian motion $\mathbb{P}_W$, with Radon-Nikodym derivative:

$$\frac{d\mathbb{P}_X}{d\mathbb{P}_W} = \exp\left(\int_0^T b(X_s;\theta)\, dX_s - \frac{1}{2}\int_0^T b(X_s;\theta)^2\, ds\right)$$

The maximum likelihood estimator given the observed path $(X_t)_{t\in[0,T]}$ is given by

$$\hat{\theta} = \arg\max_{\theta\in\Theta} L(\{X_t\}_{t\in[0,T]}; \theta).$$

Assume that there are N parameters to be estimated $\theta = (\theta_1, \ldots, \theta_N)$, then the MLE is obtained by solving the equation

$$\nabla_\theta L(\mathbf{x} \,|\, \theta) = 0.$$

Example 5.4 (MLE for the stationary Ornstein-Uhlenbeck process)**.** *Consider the stationary Ornstein-Uhlenbeck process*

$$dX_t = -\alpha X_t \, dt + dW_t,$$

with $X_0 \sim \mathcal{N}\left(0, \frac{1}{2\alpha}\right)$. The log-likelihood function is

$$\log L = -\alpha \int_0^T X_t \, dX_t - \frac{\alpha^2}{2} \int_0^T X_t^2 \, dt.$$

The Maximum Likelihood estimator is

$$\hat{\alpha} = -\frac{\int_0^T X_t \, dX_t}{\int_0^T X_t^2 \, dt}.$$

Of course, we wouldn't be able to evaluate this estimator. Given a set of discrete equidistant observations $\{X_j\}_{j=0}^J$,

$$\hat{\alpha} = -\frac{\sum_{j=0}^{J-1} X_j \Delta X_j}{\sum_{j=0}^{J-1} |X_j|^2 \Delta t},$$

where $X_j = X_{j\Delta t}$ and $\Delta X_j = X_{j+1} - X_j$. We can show that this Maximum Likelihood estimator becomes asymptotically unbiased in the large sample limit *$J \to +\infty$, for Δt fixed.*

Exercise 5.1 (Maximum Likelihood estimator for a stationary bistable SDE)**.** *Consider the SDE*

$$dX_t = (\alpha X_t - \beta X_t^3) \, dt + dW_t.$$

Our objective is to derive maximum likelihood estimatorrs for α and β for a given observation of the path X_t, $t \in [0, T]$.

1. *Show that the log of the likelihood function is*

$$\log L = \alpha B_1 - \beta B_3 - \frac{1}{2}\alpha^2 M_2 - \frac{1}{2}\beta^2 M_6 + \alpha\beta M_4,$$

where

$$M_n\left(\{X_t\}_{t\in[0,T]}\right) = \int_0^T X_t^n \, dt,$$

and

$$B_n(\{X_t\}_{t\in[0,T]}) := \int_0^T X_t^n \, dX_t.$$

2. *Consequently show that the MLE for α and β are given by*

$$\hat{\alpha} = \frac{B_1 M_6 - B_3 M_4}{M_2 M_6 - M_4^2},$$

and

$$\hat{\beta} = \frac{B_1 M_4 - B_3 M_2}{M_2 M_6 - M_4^2}.$$

5.3.3 Inference for SDEs using Bayesian Data Augmentation

This is an extremely brief introduction to the topic of Bayesian parametric inference for SDEs. The literature on the topic is extensive, the interested reader is invited to consult for example [15, 17, 2].

As before, we assume that our model is described by a (one-dimensional) Itô SDE of the form

$$dX_t = b(X_t; \theta)\, dt + \sigma(X_t; \theta)\, dW_t, \tag{5.8}$$

where $\theta \in \Theta \subset \mathbb{R}^K$ is an unknown parameter. As before, it is assumed that the conditions under which the SDE is will posed are satisfied. That is, for all $\theta \in \Theta$, the SDE has a unique, nonexploding solution. We adopt a *Bayesian imputation approach*, and work with a discretized version of (5.8):

$$X_{n+1} = X_n + b(X_n; \theta)\, \Delta t + \sigma(X_n; \theta)\, \Delta W_n.$$

Clearly,

$$X_{n+1}|X_n, \theta \sim \mathcal{N}(X_n + b(X_n; \theta)\Delta t, \sigma^2(X_n; \theta)\Delta t),$$

which has probability density function

$$p(y|X_n, \theta) = \frac{1}{\sqrt{2\pi|\sigma^2(X_n; \theta)|\Delta t}} \exp\left(-\frac{(y - X_n + b(X_n; \theta)\, \Delta t)^2}{2\sigma^2(X_n; \theta)\Delta t}\right) \tag{5.9}$$

Let us suppose that the observations X_{t_i} are available at evenly spaces intervals $t_0, t_1, \ldots,$ with intervals of length $\delta = t_{i+1} - t_i$. It is typically realistic to assume that $\Delta t \ll \delta$. In particular, we shall assume that $\Delta t = \delta/m$, for some positive integer $m > 1$. Choosing m large makes Δt small, thus reducing the discretisation bias, but also introduces $m - 1$ "missing values" between each pair of observations which must be integrated out of the problem.

To deal with these missing values, we assume that the time interval $[0, T]$ is divided into $mT + 1$ equidistant points, and X_t is observed at $0 = t_0', t_1', \ldots, t_n' = T$. Collecting all the augmented data, both missing and observed, we obtain:

$$\mathbf{X} = (x_{t_0'}, X_{t_1'}, \ldots, X_{t_{m-1}'}, x_{t_m}, X_{t_{m+1}}, \ldots, X_{t_{n-1}}, x_{t_n}).$$

The observed data is thus $D_n = (x_{t_0'}, \ldots, x_{t_n'})$. By adopting a fullying Bayesian approach, we formulate the joint posterior for parameters and missing data as :

$$p(\theta, \mathbf{X}\,|D_n) \propto p(\theta) \prod_{i=0}^{n-1} p(X_{t_{i+1}'}\,|X_{t_i'}, \theta),$$

where $p(\theta)$ is a prior density for the parameter, and $p(\cdot, X_n, \theta)$ is given by (5.9). To infer the parameters, we must sample from this distribution. Inference may proceed by alternating between draws of the missing data conditional on the current state of the model parameters, and the parameters conditional on the augmented data, as follows:

Bayesian inputation method for nonparametric inference of SDE

1. Initialise all the unknowns. We use linear interpolation of the observed data, to initialise the missing data points in $\mathbf{X}^{(1)}$.
2. Draw $\mathbf{X}^{(s)} \sim p(\cdot|\theta^{(s-1)}, D_n)$.
3. Draw $\theta^{(s)} \sim p(\cdot|\mathbf{X}^{(s)})$.
4. Continue until sufficiently many simulations have been performed.

Steps 2 and 3 involve sampling from a probability distribution for which the normalizing constant is intractible to compute. This makes Metropolis-Hastings a natural candidate for producing the samples $\mathbf{X}^{(s)}$ and $\theta^{(s)}$. Step 3 can be implemented in a relatively straightforward manner using Random-Walk proposal, however it is far less clear what is a good proposal for step 2 without getting an extremely low acceptance rate. For this reason, Gibbs sampling (which we have not discussed in this module) is typically used instead, for more details see [5, 4].

Bibliography

[1] Rick Durrett. *Probability: theory and examples.* Cambridge university press, 2010.

[2] Ola Elerian, Siddhartha Chib, and Neil Shephard. Likelihood inference for discretely observed nonlinear diffusions. *Econometrica*, 69(4):959–993, 2001.

[3] Charles J Geyer and Elizabeth A Thompson. Annealing markov chain monte carlo with applications to ancestral inference. *Journal of the American Statistical Association*, 90(431):909–920, 1995.

[4] Andrew Golightly and Darren J Wilkinson. Bayesian inference for stochastic kinetic models using a diffusion approximation. *Biometrics*, 61(3):781–788, 2005.

[5] Andrew Golightly and Darren J Wilkinson. Markov chain monte carlo algorithms for sde parameter estimation. *Learning and Inference for Computational Systems Biology*, pages 253–276, 2010.

[6] Ioannis Karatzas and Steven Shreve. *Brownian motion and stochastic calculus*, volume 113. Springer Science & Business Media, 2012.

[7] Claude Kipnis and SR Srinivasa Varadhan. Central limit theorem for additive functionals of reversible markov processes and applications to simple exclusions. *Communications in Mathematical Physics*, 104(1):1–19, 1986.

[8] Thomas Milton Liggett. *Continuous time Markov processes: an introduction*, volume 113. American Mathematical Soc., 2010.

[9] Gabriel J Lord, Catherine E Powell, and Tony Shardlow. *An Introduction to Computational Stochastic PDEs.* Number 50. Cambridge University Press, 2014.

[10] Enzo Marinari and Giorgio Parisi. Simulated tempering: a new monte carlo scheme. *EPL (Europhysics Letters)*, 19(6):451, 1992.

[11] Sean P Meyn and Richard L Tweedie. Markov chains and stochastic stability. communication and control engineering series. *Springer-Verlag London Ltd., London*, 1:993, 1993.

[12] Bernt Øksendal. *Stochastic differential equations.* Springer, 2003.

[13] Christian Robert and George Casella. *Introducing Monte Carlo Methods with R*. Springer Science & Business Media, 2009.

[14] Christian Robert and George Casella. A short history of markov chain monte carlo: subjective recollections from incomplete data. *Statistical Science*, pages 102–115, 2011.

[15] Gareth O Roberts and Osnat Stramer. On inference for partially observed nonlinear diffusion models using the metropolis–hastings algorithm. *Biometrika*, 88(3):603–621, 2001.

[16] L Chris G Rogers and David Williams. *Diffusions, Markov Processes, and Martingales: Volume 1, Foundations*. Cambridge university press, 2000.

[17] Helle Sørensen. Parametric inference for diffusion processes observed at discrete points in time: a survey. *International Statistical Review*, 72(3):337–354, 2004.

[18] Andrew TA Wood and Grace Chan. Simulation of stationary gaussian processes in [0, 1] d. *Journal of computational and graphical statistics*, 3(4):409–432, 1994.