Lecture notes

Stochastic Simulation

Fabio Nobile

A.Y. 2021-2022
Last update: November 3, 2021

EPFL

Contents

Chapter 1

Uniform Pseudo Random Number Generation

At the heart of any Monte Carlo method, is a *Random Number Generator* (RNG), i.e. a procedure that produces an infinite stream of random variables $U_1, U_2, \ldots \stackrel{iid}{\sim} \mu$ that are independent and identically distributed (i.i.d.) according to some probability distribution μ. In particular, if μ is the *uniform* distribution on $[0,1]$, i.e. $\mu = \mathcal{U}([0,1])$, the generator is called a *Uniform Random Number Generator*.

Although generators based on physical devices that exploit universal background radiation or quantum mechanics effects exist, the vast majority of current random number generators are based on algorithms that can be implemented on a computer. As such, these algorithms produce a *purely deterministic* stream of numbers $U_1, U_2, \ldots$, which, however, resembles a stream of iid random variables in the sense that the stream is indistinguishable from a random one according to a number of statistical tests. Algorithmic generators are called *Pseudo-Random Number Generators* (Pseudo-RNG).

Pseudo-RNG have the general structure, illustrated in Algorithm 1.1, where $\mathcal{S}$ is a *finite* state space, $\mathcal{U}$ the output space, $f : \mathcal{S} \to \mathcal{S}$ and $g : \mathcal{S} \to \mathcal{U}$ two given functions.

Algorithm 1.1: General structure of a Pseudo-RNG

take $X_0 \in \mathcal{S}$; `// seed`
for $k = 1, 2, \ldots$ **do**
    $X_k = f(X_{k-1})$; `// recursion on state variable` $X_k \in \mathcal{S}$
    $U_k = g(X_k)$; `// output` $U_k \in \mathcal{U}$
end

Few remarks are in order:

- The initial state X_0 is called the **seed**. *A Pseudo-RNG starting from a given seed will always produce the same sequence* $U_1, U_2, \ldots$. This is actually a convenient feature when testing or debugging a code.

- Since the state space $\mathcal{S}$ is finite, the generator eventually will repeat itself (i.e. it will revisit an already visited state). *All Pseudo-RNGs are periodic.*

We call **period** the largest number of steps ℓ taken before visiting an already visited state. The *maximal period* that a generator can have is $\ell = |\mathcal{S}|$ (where $|\mathcal{S}|$ denotes the cardinality of the state space).

A good uniform Pseudo-RNG should possibly:

1. *Have a large period*: if we need to run a Monte Carlo analysis using M (pseudo) random variables, the period ℓ of the generator should be $\ell \gg M$ (otherwise the property of independent samples is clearly broken).

2. *Pass a battery of statistical tests* for uniformity and independence.

3. *Be fast and efficient*: many MC techniques require the generation of billions of random variables. In certain fields (e.g. finance) the generation time is a big issue.

4. *Be reproducible*: in certain cases it is important to be able to reproduce a stream $U_1, U_2, \dots$ without the need of storing it (debugging purposes, advanced MC variance reduction techniques etc.)

5. *Have the possibility to generate multiple streams.* This is important when running a Monte Carlo analysis in a parallel environment: each processor should use a stream not overlapping with the ones used by the other processors.

6. *Avoid producing the numbers 0 and 1.* The value zero might produce undesirable results as "division by zero". Since the event "$U = 0$" has zero probability, the Pseudo-RNG should never produce the value zero.

1.1 Some common uniform Pseudo-RNG

The most commonly used generators are based on *linear* recurrences. We present hereafter some examples.

Linear Congruential Generator (LCG)

It is characterized by a state space $\mathcal{S} = \{0, 1, \dots, m-1\}$ (m is called the modulus), two natural numbers $a, b \in \mathbb{N}$ and the following recurrence and output

$$X_k = (aX_{k-1} + b) \mod m, \qquad U_k = \frac{X_k}{m}, \ k \geq 1.$$

LCG have been popular for many years but are now somewhat outdated (e.g. Matlab versions up to 5 were using one of those). LCG can generate any number in $\{0, \frac{1}{m}, \dots, \frac{m-1}{m}\}$ and m^{-1} should be chosen of the order of the floating point machine precision (ε-machine).

A popular choice is the Lewis-Goodman-Miller LCG with $a = 7^5 = 16807$, $b = 0$, $m = 2^{31} - 1 \approx 2 \cdot 10^9$, which has a maximal period of $m - 1 \approx 4 \cdot 10^9$, too small for today's applications.

Multiple recursive generator (MRG) of order q

For natural numbers $a_1, \dots, a_q \in \mathbb{N}$ and seeds $X_0, X_{-1}, \dots, X_{-q+1} \in \{0, \dots, m-1\}$, it is defined by the recurrence and output

$$X_k = (a_1 X_{k-1} + a_2 X_{k-2} + \cdots + a_q X_{k-q}) \mod m, \qquad U_k = \frac{X_k}{m}, \quad k \geq 1. \tag{1.1}$$

A MRG can be written in the general form of Algorithm 1.1 by introducing the vector $\boldsymbol{X}^{(k)} = (X_{k-q+1}, \dots, X_k)^\top$ and the integer matrix $A \in \mathbb{N}^{q \times q}$,

$$A = \begin{pmatrix} 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ a_q & a_{q-1} & \dots & a_1 \end{pmatrix}.$$

As such, the recurrence (1.1) can be written equivalently as

$$\boldsymbol{X}^{(k)} = A \boldsymbol{X}^{(k-1)} \mod m, \qquad U_k = \frac{(\boldsymbol{X}^{(k)})_q}{m}, \quad k \geq 1, \tag{1.2}$$

for which the state space is $\mathcal{S} = \{0, 1, \dots, m-1\}^q$ and the maximal period can be up to $m^q - 1$. For a more general integer valued invertible matrix A, a generator of the form (1.2) is called *Matrix Congruential Generator of order q*.

Combined Generators

Here, the idea is to combine the output of several generators which, individually, may be of poor quality, to make a superior quality generator.

Example 1.1 (Wichman-Hill). *This combines 3 LCGs*

$$\begin{aligned} X_k &= (171 X_{k-1}) \mod m_1 \quad (m_1 = 30269) \\ Y_k &= (172 Y_{k-1}) \mod m_2 \quad (m_2 = 30307) \\ Z_k &= (170 Z_{k-1}) \mod m_3 \quad (m_3 = 30323) \end{aligned} \tag{1.3}$$

with

$$U_k = \frac{X_k}{m_1} + \frac{Y_k}{m_2} + \frac{Z_k}{m_3} \mod 1.$$

It has a period of $\ell \approx 6.95 \cdot 10^{12}$ (which is not very large for today's applications) and performs quite well in simple statistical tests.

Example 1.2 (MRG32k3a). *This is a combination of 2 MRGs:*

$$\begin{aligned} X_k &= (a_2 X_{k-2} + a_3 X_{k-3}) \mod m_1 \\ Y_k &= (b_1 Y_{k-1} + b_3 X_{k-3}) \mod m_2 \end{aligned}$$

with

$$U_k = \begin{cases} \frac{X_k - Y_k + m_1}{m_1 + 1}, & \textit{if } X_k \leq Y_k \\ \frac{X_k - Y_k}{m_1 + 1}, & \textit{if } X_k > Y_k \end{cases}$$

and suitable values of $a_2, a_3, b_1, b_3, m_1, m_2$. *This has a period of* $\ell \approx 3 \cdot 10^{57}$ *and passes all statistical tests. It has been implemented in many packages including Matlab, Mathematical, Intel's MKL library etc.*

Modulo 2 Linear Generators

These are Matrix Congruential Generators with modulus $m = 2$. Since binary operations are in general faster than integer operations, these generators are usually fast. To have long periods, the order q has to be large (the maximal period is $2^q - 1$). Among these generators a popular one is the **Linear Feedback Shift Register (LFSR) Generator** also called the Tausworthe generator. The recurrence formula is in the form of a MRG (1.1) with $m = 2$, whereas the output is given by

$$U_k = \sum_{\ell=1}^{w} X_{kw+\ell-1} 2^{-\ell},$$

where each word of w bits $(X_0, \dots, X_{w-1})$, $(X_w, \dots, X_{2w-1})$, $\dots$ is interpreted as a binary representation of a number in $[0, 1]$. For fast generation, most of the a_j are zero. In many cases there is only one non-zero multiplier a_r apart from a_q, and the operation in the recurrence correspods to a (modulo 2) bit addition $X_k = X_{k-r} \bigoplus X_{k-q}$. Generalizations of the LFSR generator include the *Mersenne Twister* generator that is now the default generator in Matlab, and R. It has a period of $2^{19937} - 1$, is very fast and passes all practical statistical tests. The default generator in Python (numpy) is instead a *Permuted Congruential Generator* (PCG). It uses a "medium quality" LCG with $m = 2^{128}$ (unsigned long long long integers represented by 128 bits) and improves its performance by preforming a state dependent permutation on the 128 bit and outputing only the first 64 of them. It has period of 2^{128}, excellent statistical properties and is very fast with jump ahead and multiple straming possibilities.

1.2 Empirical tests for RNG

Several statistical tests have been proposed to asses the quality of a RNG. Today's most comprehensive test suite is *TestU01* developed by L'Ecuyer and Simard [3]. In the next section we review some non-parametric Goodness-of-Fit tests that can be used to assess the uniformity of the sequence $U_1,\ U_2, \dots$ produced by a Pseudo-RNG. For generality purposes, we present these tests assuming that U_j has a general cumulative distribution function F not necessarily uniform. Then, in Section 1.2.2, we discuss some tests to assess the independence of the sequence.

1.2.1 Non-parametric Goodness-of-Fit Tests

Let U be a random variable with values in a certain interval $I \subset \mathbb{R}$, and cumulative distribution function (CDF) $F(x) = \mathbb{P}(U \leq x)$. We will assume that F is absolutely continuous so that a probability density function $f : I \to \mathbb{R}_+$ exists, such that $\int_{[a,b]\subset I} f(x)\, dx = F(b) - F(a)$.

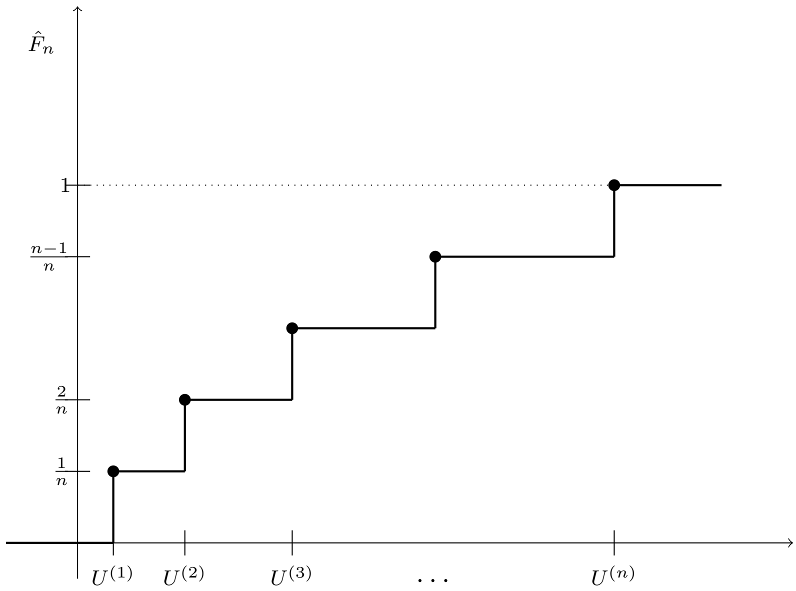


Figure 1.1: Empirical cumulative distribution function.

Let $\boldsymbol{U} = (U_1, \ldots, U_n)$ be a random sample and denote by $\hat{F}_n(x)$ the *empirical distribution function*

$$\hat{F}_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{U_i \le x\}} = \frac{\#\{U_i \le x,\, i = 1, \ldots, N\}}{n}.$$

See Figure 1.1 for an illustration. In the figure, $(U^{(1)}, U^{(2)}, \ldots, U^{(n)})$ denote the ordered sample $\boldsymbol{U}$. We want to test the hypothesis H_0 that $\boldsymbol{U}$ has been drawn independently from the distribution F.

Q-Q plot

A first simple graphical test to see if the sample $\boldsymbol{U}$ has been drawn from the distribution F is to plot the quantiles of $\hat{F}_n$ versus the corresponding quantiles of F. We recall that the t-quantile of F is defined as

$$q_t = \operatorname*{argmin}_x \{F(x) \ge t\},$$

and similarly for empirical distribution $\hat{q}_t = \operatorname{argmin}_x\{\hat{F}_n(x) \ge t\}$, which leads to $\hat{q}_{\frac{j}{n}} = U^{(j)}$, $\forall j = 1, \ldots, n$, i.e. the $\frac{j}{n}$ quantile of the empirical distribution is the j-th value in the ordered sample $\boldsymbol{U}$. A better quantile estimator is actually given by $\hat{q}_{\frac{j}{n+1}} = U^{(j)}$.

If the sample $\boldsymbol{U}$ is indeed drawn from the distribution F independently, the empirical quantiles $\hat{q}_{\frac{j}{n+1}}$, when plotted against the corresponding true quantiles $q_{\frac{j}{n+1}}$, should be well aligned on the diagonal, as in Figure 1.2.

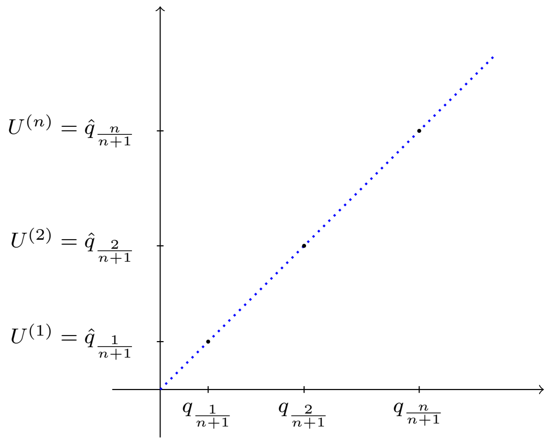


Figure 1.2: Q-Q plot.

Kolmogorov-Smirnov Test

This is a more quantitative test that compares the empirical distribution $\hat{F}_n$ with the true one F (see Figure 1.3). Let $D_n = \sup_x |\hat{F}_n(x) - F(x)|$ (which is a random variable as it depends on the random sample $\boldsymbol{U}$). For a continuous distribution F, and under the null hypothesis H_0, it is known that

$$\sqrt{n}D_n \xrightarrow{\mathrm{d}} K \quad \text{independently of } F$$

where K is a Kolmogorov random variable with CDF

$$F_K(x) = \mathbb{P}\left(K \le x\right) = \left(1 + 2\sum_{j=1}^{\infty}(-1)^j e^{-2j^2x^2}\right)\mathbb{1}_{\{x>0\}}$$

and corresponds to the distribution of $\max_{t\in[0,1]} |B(t)|$ where $B(t)$ is a Brownian bridge in $[0,1]$. This result shows that, under H_0, $\hat{F}_n \to F$ uniformly at a rate $O(1/\sqrt{n})$ in a probabilistic sense. Based on this result, we can reject H_0 at level α if $\sqrt{n}D_n > K_\alpha$ with K_α the α-quantile of K: $\mathbb{P}\left(K \le K_\alpha\right) = 1 - \alpha$. The quantiles K_α are tabulated.

χ^2 Test

We split I in $m+1$ non-overlapping subintervals (classes) I_j, $j = 1, \ldots, m+1$ such that $\bigcup_{j=1}^{m+1} I_j = I$. For each j, let $p_j = \mathbb{P}\left(U \in I_j\right)$ be the probability that U is in I_j and define

$$N_j = \sum_{i=1}^{n} \mathbb{1}_{\{U_i \in I_j\}} = \#\{U_i \text{ that fall in } I_j\},$$

Then, under H_0, we have $\mathbb{E}\left[N_j\right] = np_j$. We define then the statistics

$$\hat{Q}_m = \sum_{j=1}^{m+1} \frac{(N_j - np_j)^2}{np_j}$$

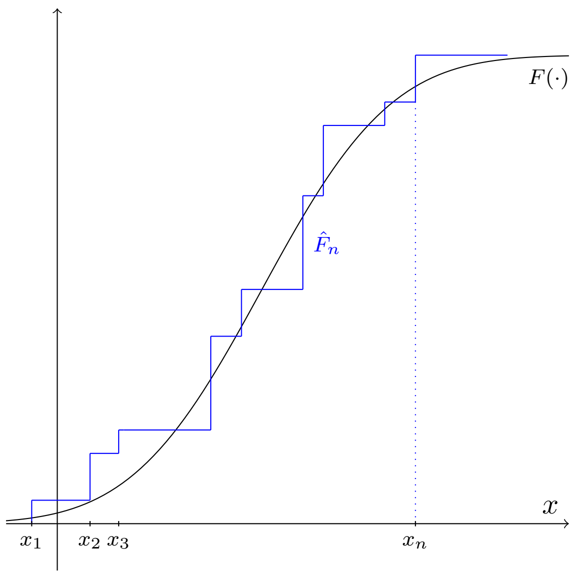


Figure 1.3: Kolmogorov-Smirnov test

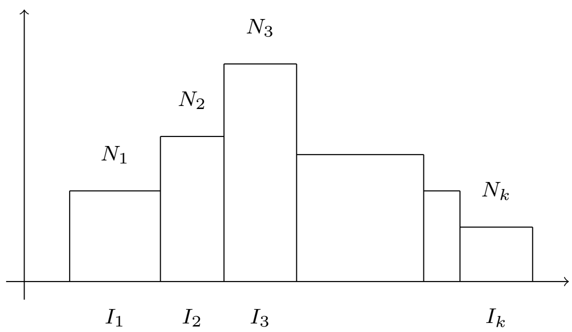


Figure 1.4: χ^2 test

which has an asymptotic $\chi^2(m)$ distribution with m degrees of freedom ($m = \#$ classes $-$ 1). We can then reject the null hypothesis H_0 at level α if $\hat{Q}_m > q_{1-\alpha}$ where $q_{1-\alpha}$ is the $1-\alpha$ quantile of the $\chi^2(m)$ distribution. Notice that $\{(I_j, N_j),\ j = 1, \ldots, m+1\}$ defines a histogram of the sample and $\hat{Q}_m$ estimates the deviation from the "true" histogram $\{(I_j, np_j),\ j = 1, \ldots, m+1\}$, as in Figure 1.4.

1.2.2 Empirical tests for independence

We consider here a sample $\boldsymbol{U} = (U_1, U_2, \ldots, U_n)$ produced by a uniform Pseudo-RNG and present two statistical tests that can be used to test the null hypothesis H_0 that $\{U_i\}_i$ are mutually independent and uniformly distributed in $(0, 1)$.

Serial Test

We test whether groups of variables are jointly uniformly distributed. Namely we group $\boldsymbol{U}$ in groups of length d: $\boldsymbol{U}_1 = (U_0, \ldots, U_{d-1})$, $\boldsymbol{U}_2 = (U_d, \ldots, U_{2d-1}), \ldots$ and test whether $\{\boldsymbol{U}_j,\ j = 1, \ldots, \frac{n}{d}\}$ are drawn independently from a multivariate uniform distribution $\mathcal{U}([0,1]^d)$, using for instance a χ^2 test on the partition $I_{j_1 \cdots j_d} = [\frac{j_1-1}{m}, \frac{j_1}{m}] \times \cdots \times [\frac{j_d-1}{m}, \frac{j_d}{m}]$, $(j_1, \ldots, j_d) \in \{1, \ldots, m\}^d$. Of course, n/d should be sufficiently large compared to m^d so that each class has enough samples and one can apply the asymptotic result.

Gap Test

Let $T_1, T_2, \ldots$ denote the times when the process $\{U_i\}_{i=1}^n$ visits a given interval $(\alpha, \beta) \subset [0,1]$, namely T_j is such that $U_{T_j} \in (\alpha, \beta)$ and $U_K \notin (\alpha, \beta)$, $K \notin \{T_1, T_2, \ldots\}$. Let $Z_i = T_i - T_{i-1} - 1$ be the gap length between two consecutive visits (here $T_0 = 0$). Under H_0, Z_i are iid with a geometric distribution with parameter $p = \beta - \alpha$, i.e.

$$\mathbb{P}\left(Z = j\right) = p(1-p)^j, \quad j = 0, 1, 2, \ldots.$$

One can use a $\chi^2(m)$ test to test whether the $\{Z_i\}_i$ have the correct geometric distribution, using the classes $Z = 0, Z = 1, \ldots, Z = r$, $Z > r$.

Chapter 2

Random Variable Generation

From a uniform (pseudo) random number generator one can construct (pseudo) random generators for many other distributions. We discuss hereafter a few approaches.

2.1 Inverse-transform method

The inverse transform method is probably the most straightforward method to generate a random variable with a given distribution and relies on the possibility to invert the cumulative distribution function. We present it separately in the case of a discrete and a continuous random variable.

Discrete random variable

Consider a discrete random variable X, which can take the values $x_1 < x_2 < \cdots < x_n$ with probability mass function (pmf) $p_i = \mathbb{P}(X = x_i)$. Let $F_i = \sum_{j=1}^{i} p_j = \mathbb{P}(X \leq x_i)$, $i = 1, \ldots, n$ and $F_0 = 0$ be the cumulative probabilities. Then X can be generated starting from a uniform random variable $U \sim \mathcal{U}([0,1])$ by the following

Algorithm 2.1: Discrete inverse-transform.

1 Generate $U \sim \mathcal{U}([0,1])$
2 Set $X = x_i$ if $F_{i-1} < U \leq F_i$

That this algorithm generates the correct random variable is easily seen since $\mathbb{P}(X = x_i) = \mathbb{P}(F_{i-1} < U \leq F_i) = \mathbb{P}(U \subset (F_i - p_i, F_i]) = p_i$. Figure 2.1 gives a graphical illustration of the method.

Example 2.1 (Bernoulli)**.** *Let $X \sim \mathrm{Be}(p)$ be a Bernoulli random variable that satisfies $\mathbb{P}(X = 0) = 1 - p$, $\mathbb{P}(X = 1) = p$. Given $U \sim \mathcal{U}([0,1])$, one sets $X = 1$ if $U > 1 - p$ and $X = 0$ otherwise.*

Continuous random variable

Consider a continuous random variable X taking values in an interval $[a,b]$ with continuous and strictly increasing cumulative distribution function (cdf) $F : [a,b] \to [0,1]$, $F(x) =$

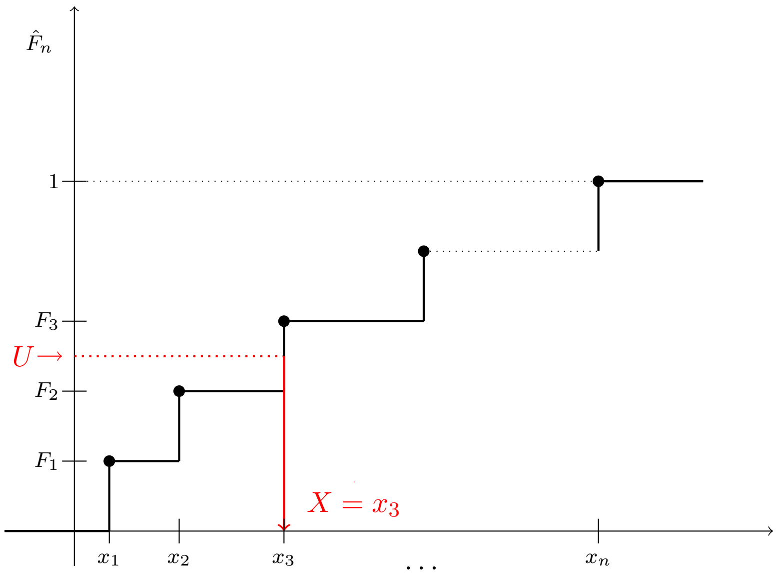


Figure 2.1: Discrete inverse transform method.

$\mathbb{P}(X \leq x)$, with $F(a) = 0$ and $F(b) = 1$. In this case the inverse function $F^{-1} : [0,1] \to [a,b]$ is uniquely defined and X can be generated starting from a uniform random variable $U \sim \mathcal{U}([0,1])$ by the following

Algorithm 2.2: Continuous inverse-transform

1 Generate $U \sim \mathcal{U}([0,1])$
2 Set $X = F^{-1}(U)$

Again, one verifies easily that this algorithm generates a random variable with the correct distribution. Indeed $\mathbb{P}(X \leq x) = \mathbb{P}\left(F^{-1}(U) \leq x\right) = \mathbb{P}(U \leq F(x)) = F(x)$. Figure 2.2 gives a graphical interpretation of the method.

Example 2.2 (Exponential)**.** *Let $X \sim \mathrm{Exp}(\lambda)$ be an exponential random variable with pdf $f(x) = \lambda e^{-\lambda x}$ and cdf $F(x) = 1 - e^{-\lambda x}$. Inversion gives $X = F^{-1}(U) = -\frac{1}{\lambda}\log(1-U)$. Since $\tilde{U} = 1 - U$ has the same distribution as U, an equivalent inversion formula is $X = -\frac{1}{\lambda}\log U$ with $U \sim \mathcal{U}([0,1])$.*

Both the discrete and the continuous case can be combined together by defining a proper right inverse of F when it is not continuous or not strictly monotone. Let X be a random variable with cdf F. Its *Generalized inverse* is defined as $F^{-}(u) = \inf\{x : F(x) \geq u\}$. Actually, the infimum can be replaced by a minimum since F is right continuous. Then X can be generated as $X = F^{-}(U)$ with $U \sim \mathcal{U}([0,1])$. Notice that with this definition of F^{-} we recover the discrete inverse-transform as a particular case.

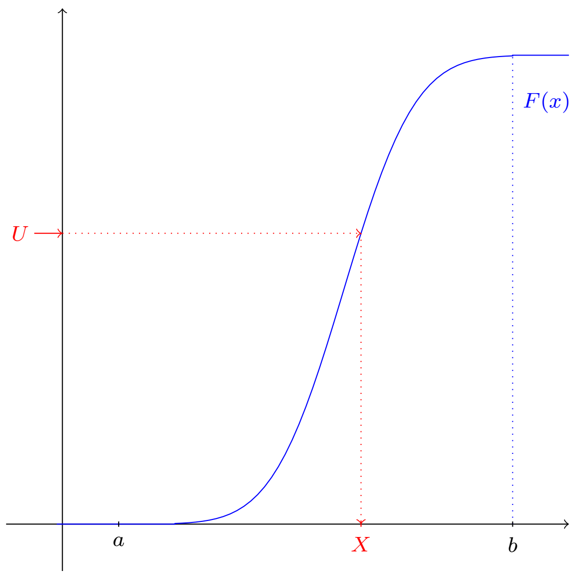


Figure 2.2: Continuous inverse transform

2.2 Composition method

Suppose that a random variable X has a mixture distribution, i.e. its cdf has the form $F(x) = \sum_{i=1}^{n} p_i F_i(x)$ where F_i, $i = 1, \dots, n$ are cdf functions and p_i, $i = 1, \dots, n$ are positive weigts such that $\sum_{i=1}^{n} p_i = 1$. If the cdfs F_i are absolutely continuous with corresponding densities f_i, then X has a pdf $f(x) = \sum_{i=1}^{n} p_i f_i(x)$. The random variable X can be generated by the following:

Algorithm 2.3: Composition method

1. Generate discrete r.v. Y, $\mathbb{P}(Y = i) = p_i$
2. Generate $X \sim F_Y$ e.g. by inversion

Example 2.3 (Laplace distribution)**.** *Let $X \sim Lapl(\lambda)$ with pdf*

$$f(x) = \frac{\lambda}{2} e^{-\lambda|x|} = \frac{1}{2} \underbrace{\lambda e^{-\lambda x} \mathbb{1}_{\{x \geq 0\}}}_{\sim \mathrm{Exp}(1)} + \frac{1}{2} \underbrace{\lambda e^{\lambda x} \mathbb{1}_{\{x < 0\}}}_{\sim -\mathrm{Exp}(1)}.$$

Then, X can be generated by the composition method by first generating $B \sim \mathrm{Be}(\frac{1}{2})$, $Y \sim \mathrm{Exp}(\lambda)$ and then setting $X = Y$ if $B = 1$ and $X = -Y$ if $B = 0$, or, equivalently, $X = (2B - 1)Y$.

2.3 Alias method

A discrete random variable X taking the values $x_1 < x_2 < \cdots < x_n$ with non-uniform probabilities $p_i = \mathbb{P}(X = x_i)$ can be generated by the discrete inverse-transform method.

However, if n is large, the search for the interval $(F_{i-1}, F_i]$ such that $U \in (F_{i-1}, F_i]$, where $F_i = \sum_{j=1}^{i} p_j$, might be costly. In this case, an alternative approach consists in representing the cdf $F(x)$ as a mixture distribution $F(x) = \sum_{i=1}^{n} \frac{1}{n} G_i(x)$ such that each G_i is a two points distribution (Bernoulli) and apply the composition method.

With little abuse of notation, we describe the algorithm using the probability "density" function which, in this case, is a linear combination of concentrated masses (delta distributions) in the points $\{x_i\}$, i.e. $f(x) = \sum_{i=1}^{n} p_i \delta_{x_i}(x)$. We therefore aim at rewriting it as $f(x) = \sum_{i=1}^{n} \frac{1}{n} g_i(x)$ where each g_i, $i = 1, \dots, n$, has the form $g_i(x) = \alpha_i \delta_{x_{\ell_i}}(x) + (1-\alpha_i)\delta_{x_{k_i}}(x)$, with $\ell_i, k_i \in \{1, \dots, n\}$ and the distributions g_i are constructed iteratively.

- Choose ℓ_1 and k_1 such that $p_{\ell_1} < \frac{1}{n}$ and $p_{\ell_1} + p_{k_1} \geq \frac{1}{n}$ (such a choice always exists since $\{p_i\}$ are not uniform) and set $\alpha_1 = np_{\ell_1}$. Then

$$\begin{aligned} f(x) = f^{(0)}(x) &= p_{\ell_1}\delta_{x_{\ell_1}}(x) + p_{k_1}\delta_{x_{k_1}}(x) + \sum_{i \neq \ell_1, k_1} p_i \delta_{x_i}(x) \\ &= \frac{1}{n} g_1(x) + \frac{n-1}{n} f^{(1)}(x) \end{aligned}$$

  with

$$f^{(1)}(x) = \frac{n(p_{\ell_1} + p_{k_1}) - 1}{n-1} \delta_{k_1}(x) + \frac{n}{n-1} \sum_{i \neq \ell_1, k_1} p_i \delta_{x_i}(x).$$

  Notice that now $f^{(1)}(x)$ contains only point masses in $\{x_i, i \neq \ell_1\}$

- Iterate the procedure on $f^{(1)}, f^{(2)}, \dots$ until we reach the desired form.

We can now construct the following algorithm which does not require a search (however it requires to build in advance the table of distributions $\{g_i\}$)

Algorithm 2.4: Alias method

1 Generate $U \sim \mathcal{U}([0,1])$ and set $Y = \lceil nU \rceil$ `// hence` $Y \sim \mathcal{U}(\{1, 2, \dots, n\})$
2 Generate $X \sim G_Y$ `// hence` $X \sim Be(\alpha_Y)$ `with values` $\{x_{\ell_Y}, x_{k_Y}\}$

2.4 Acceptance-Rejection method

Consider a continuous random variable X with pdf f and cdf F. In cases where F is difficult to invert, the inverse-transform method is not viable. Another situation which may arise is when f is known only up to a multiplicative constant, i.e. $f(x) = \kappa \tilde{f}(x)$, with $\kappa = (\int_{\mathbb{R}} \tilde{f}(x)\, dx)^{-1}$ and we only know $\tilde{f}$ whereas κ is difficult or impossible to evaluate. In both cases, the acceptance-rejection method might represent a good alternative to generate X.

The idea is to find an auxiliary pdf g which is easy to sample from, and a constant $C \geq \kappa^{-1}$ such that $\tilde{f}(x) \leq Cg(x)$ for all $x \in \mathbb{R}$. Then, the acceptance-rejection algorithm reads:

Algorithm 2.5: Acceptance-Rejection (AR) algorithm

1 Generate $Y \sim g$
2 Generate $U \sim \mathcal{U}([0,1])$ independent of Y
3 If $U \leq \dfrac{\tilde{f}(Y)}{Cg(Y)}$ set $X = Y$, otherwise return to step 1

Lemma 2.1. *The acceptance-rejection algorithm 2.5 generates a random variable X with the desired pdf $f(x) = \kappa \tilde{f}(x)$ (even without knowing κ).*

Proof. Observe that the distribution of X is the distribution of Y conditional to the event $U \leq \dfrac{\tilde{f}(Y)}{Cg(Y)}$. Therefore $\mathbb{P}\left(X \leq x\right) = \mathbb{P}(Y \leq x \mid U \leq \frac{\tilde{f}(Y)}{Cg(Y)}) = \frac{\mathbb{P}\left(Y \leq x, U \leq \frac{\tilde{f}(Y)}{Cg(Y)}\right)}{\mathbb{P}\left(U \leq \frac{\tilde{f}(Y)}{Cg(Y)}\right)}$. Now,

$$\mathbb{P}\left(Y \leq x,\, U \leq \frac{\tilde{f}(Y)}{Cg(Y)}\right) = \int_{-\infty}^{x} \left(\int_0^{\frac{\tilde{f}(y)}{Cg(y)}} du\right) g(y)\, dy = \int_{-\infty}^{x} \frac{\tilde{f}(y)}{Cg(y)} g(y)\, dy = \frac{1}{C}\int_{-\infty}^{x} \tilde{f}(y)\, dy$$

and

$$\mathbb{P}\left(U \leq \frac{\tilde{f}(Y)}{Cg(Y)}\right) = \int_{\mathbb{R}} \frac{\tilde{f}(y)}{Cg(y)} g(y)\, dy = \frac{1}{C}\int_{\mathbb{R}} \tilde{f}(y)\, dy$$

so that

$$\mathbb{P}\left(X \leq x\right) = \frac{\int_{-\infty}^{x} \tilde{f}(y)\, dy}{\int_{\mathbb{R}} \tilde{f}(y)\, dy} = \int_{-\infty}^{x} f(y)\, dy = F(x).$$

□

The probability of acceptance in Algorithm 2.5 is $\mathbb{P}\left(U \leq \frac{\tilde{f}(Y)}{Cg(Y)}\right) = \frac{1}{\kappa C}$ and since the trials (Y,U) are independent, the number of trials required to obtain a successful pair (X,U) has a geometric distribution $\mathrm{Geom}(\frac{1}{\kappa C})$ with expected value κC. For the algorithm to be efficient, C should be as close as possible to κ^{-1}.

We now give a geometric interpretation of the acceptance rejection method, which is illustrated in Figure 2.3. Such interpretation is based on following lemma.

Lemma 2.2. *Consider a positive integrable function $\tilde{h} : \mathbb{R} \to \mathbb{R}_+$, the region $A_{\tilde{h}} = \{(x,u) : x \in \mathbb{R}, 0 \leq u \leq \tilde{h}(x)\}$, and the (normalized) probability density function $h(x) = \frac{\tilde{h}(x)}{\int \tilde{h}(x)\, dx}$ associated to $\tilde{h}$. A pair of random variables (X,U) is uniformly distributed in $A_{\tilde{h}}$ if and only it $X \sim h$ and $U \mid X \sim \mathcal{U}([0,\tilde{h}(X)])$.*

Proof. Assume first $(X,U) \sim \mathcal{U}(A_{\tilde{h}})$. Then, its probability density function is $f_{(X,U)}(x,u) = \frac{1}{|A_{\tilde{h}}|} = \frac{1}{\int_{\mathbb{R}} \tilde{h}(x)\, dx}$. It follows that the pdf of X is $f_X(x) = \int_0^{\tilde{h}(x)} f_{(X,U)}(x,u)\, du = \frac{\tilde{h}(x)}{|A_h|} = h(x)$ and the conditional probability density function of $U|X$ is $f_{U|X}(u|x) = \frac{f_{(X,U)}(x,u)}{f_X(x)} = \frac{1}{\tilde{h}(x)}$, hence $U|X \sim \mathcal{U}([0,\tilde{h}(X)])$.

Consider now the converse case, $X \sim h$ and $U \mid X \sim \mathcal{U}([0,\tilde{h}(X)])$. Then clearly $f_{(X,U)}(x,u) = f_{U|X}(u|x) f_X(x) = \frac{1}{\tilde{h}(x)} h(x) = \frac{1}{|A_h|}$, hence $(X,U) \sim \mathcal{U}(A_h)$. □

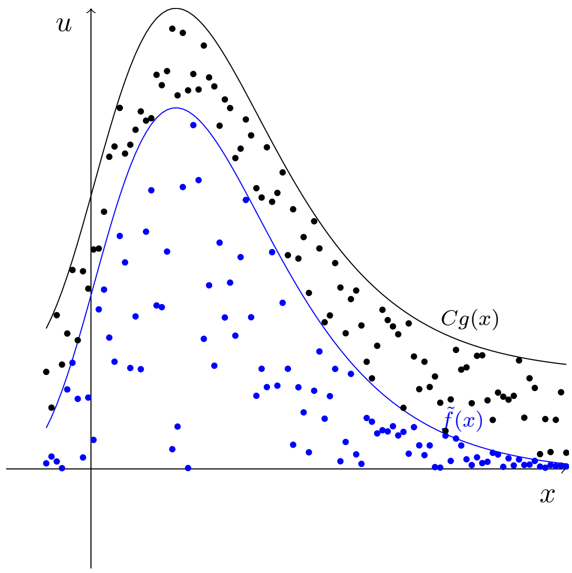


Figure 2.3: Graphical illustration of the Acceptance Rejection method: only the blue points are retained and their abscissas are distributed according to f.

This observation leads to the following geometrical interpretation of the AR algorithm: in Step 1-2, one draws samples (Y, U) uniformly in the region $A_{Cg} = \{(y, u) : 0 \leq u \leq Cg(y)\}$. In step 3, one retains only those samples that fall in the region $A_{\tilde{f}} = \{(x, u), 0 \leq u \leq \tilde{f}(x)\}$. Hence, the abscissas of the retained points have the desired density $\frac{\tilde{f}(x)}{\int \tilde{f}(x)\,dx} = f(x)$.

Example 2.4. *Let $Z \sim N(0, 1)$ and suppose we want to sample from $X = Z|(Z \geq 1)$, i.e. we want to sample the tail of a standard normal distribution for $Z \geq 1$. The pdf of X is $f(x) \propto e^{-x^2/2}\mathbb{1}_{\{x\geq 1\}}$. We could take as proposal distribution g an exponential* Exp(1) *translated in* 1*, i.e. $g(x) = e^{-(x-1)}\mathbb{1}_{\{x\geq 1\}}$ (see Figure 2.4). We have in this case $\tilde{f}(x) = e^{-x^2/2}\mathbb{1}_{\{x\geq 1\}}$ and $\tilde{f}(x) \leq g(x)\frac{1}{\sqrt{e}}$ for all $x \geq 1$, hence we can take $C = \frac{1}{\sqrt{e}}$). The AR Algorithm reads*

1. *Generate $Y = 1 + \mathrm{Exp}(1)$*
2. *Generate $U \sim \mathcal{U}(0, 1)$*
3. *If $U \leq e^{-Y^2/2+Y-1/2}$ set $X = Y$, otherwise return to step 1.*

The acceptance probability is $\sqrt{e}\int_1^\infty e^{-y^2/2}\,dy = \sqrt{2\pi e}(1 - \phi(1)) \approx 0.66$ with ϕ the cdf of a standard normal distribution. Notice that if we just sample from $N(0, 1)$ and reject all samples less than 1, we would have an acceptance rate ≈ 0.16.

2.4.1 Squeezing

In certain cases, the expression $\tilde{f}(x)$ might be complicated and costly to evaluate, whereas $g(x)$ has generally a simple expression. To minimize the number of evaluations of $\tilde{f}$, one

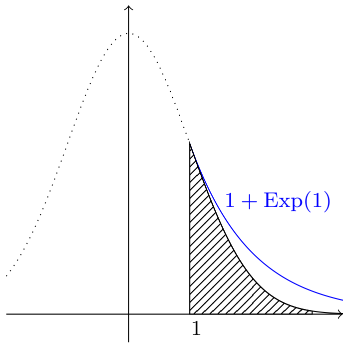


Figure 2.4: Sampling the tail of a Normal distribution by AR with an exponential proposal

could look for another auxiliary function $\hat{g}$, which is also inexpensive to evaluate, such that $\hat{g}(x) \leq \tilde{f}(x) \leq Cg(x)$ for all $x \in \mathbb{R}$ and modify the AR algorithm as follows:

Algorithm 2.6: AR algorithm with squeezing.

1 Generate $Y \sim g$
2 Generate $U \sim \mathcal{U}([0,1])$
3 If $U \leq \dfrac{\hat{g}(Y)}{Cg(Y)}$ set $X = Y$, otherwise, evaluate $\tilde{f}(Y)$
4 If $U \leq \frac{\tilde{f}(Y)}{Cg(Y)}$ set $X = Y$
5 else reject Y and go back to 1

2.4.2 Adaptive AR for log-concave densities

A particularly effective adaptive AR algorithm can be set up in the case where $\log \tilde{f}(x)$ is a *concave* function. We illustrate the procedure graphically in Figure 2.5.

Let $Z_r = \{z_1, \dots, z_r\}$ be an initial set of points. Thanks to the log-concavity of $\tilde{f}$, we have $e^{\hat{s}(x)} \leq \tilde{f}(x) \leq e^{s(x)}$ for all $x \in \mathbb{R}$, with $s(x)$ and $\hat{s}(x)$ as in the figure. Setting now $C = \int_{\mathbb{R}} e^{s(x)}\, dx$, $g(x) = C^{-1}e^{s(x)}$, $\hat{g}(x) = e^{\hat{s}(x)}$, we can apply the AR algorithm with squeezing. Notice that $g(x)$ is a piecewise exponential function and can be sampled effectively by the composition method. Moreover, once a new sample X has been generated, it can be added to the set $Z_r \to Z_{r+1} = Z_r \cup \{X\}$ so that the squeezing becomes more and more effective the more variables we generate.

2.5 Ad Hoc methods

The methods illustrated above are 'general purpose' methods, applicable to any distribution. However, for specific distributions such as Normal, Gamma, Possion, Binomial etc, there are often much more efficient methods for random variable generation, which exploit the special structure and probabilistic interpretation of the underlying distribution. (See [2, Chapter 4]). We mention only one possible algorithm to generate variables from the Normal distribution $N(0,1)$.

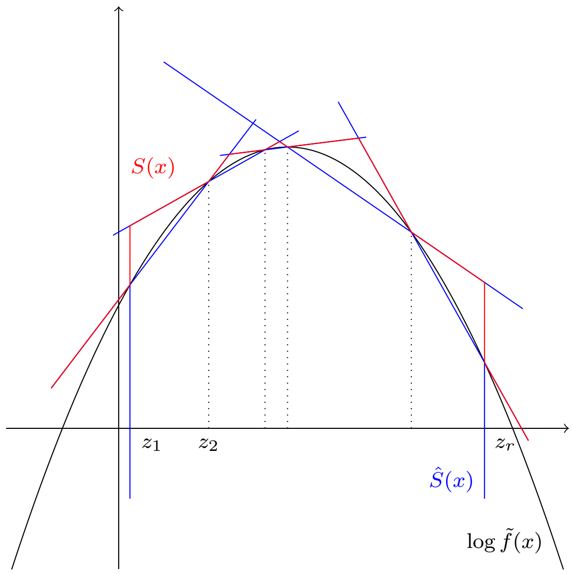


Figure 2.5: Graphical illustration of log concave density

2.5.1 Box-Muller method

Let $X, Y \sim N(0,1)$ be independent standard normal random variables, and (ρ, θ) their representation in polar coordinates. Since $X^2 + Y^2 \sim \chi_2^2 = \text{Exp}(\frac{1}{2})$ (χ_2^2 is a chi-square distribution with 2 degrees of freedom, which coincides with an exponential of parameter $\frac{1}{2}$), it follows that $\rho^2 \sim \text{Exp}(\frac{1}{2})$. Moreover, by the radial symmetry of the bivariate normal distribution $N(0, I_2)$, the distribution of (X, Y) given $\rho^2 = X^2 + Y^2$ is uniform in $[0, 2\pi)$. From these considerations, an algorithm to generate $(X, Y) \sim N(0, I_2)$ is:

Algorithm 2.7: Box-Muller method.

1 Generate $U \sim \mathcal{U}(0,1)$ and set $\rho = \sqrt{-2\log U}$ `// hence` $\rho^2 \sim \text{Exp}(\frac{1}{2})$
2 Generate $V \sim \mathcal{U}(0,1)$ and set $\Theta = 2\pi V$ `// hence` $\Theta \sim \mathcal{U}([0, 2\pi])$
3 Set $X = \rho\cos\Theta$, $Y = \rho\sin\Theta$.

2.6 Multivariate Random Variable Generation

We consider now the problem of generating from a multivariate distribution. Let $\boldsymbol{X} = (X_1, \dots, X_n)^\top \in \mathbb{R}^n$ be a vector of random variables with joint cumulative distribution function $F(\boldsymbol{z}) = F(z_1, \dots, z_n) = \mathbb{P}(X_1 \le z_1, \dots, X_n \le z_n)$ and probability density function $f(\boldsymbol{x}) = f(x_1, \dots, x_n)$, if it exists, such that

$$F(z_1, \dots, z_n) = \int_{-\infty}^{z_1} \cdots \int_{-\infty}^{z_n} f(x_1, \dots, x_n)\, dx_1 \dots dx_n.$$

The inverse transform method is not (directly) applicable in this case since the cumulative distribution function $F : \mathbb{R}^n \to \mathbb{R}$ is not invertible. The acceptance-rejection method, on the other hand, generalizes straightforwardly to the multivariate case. However, it is in general not an easy task to find an auxiliary function $g(\boldsymbol{x})$ and a constant $C > 1$ such that $f(\boldsymbol{x}) \leq Cg(\boldsymbol{x})$, $\forall \boldsymbol{x} \in \mathbb{R}^n$ and the method leads leads to reasonable acceptance rates.

In general, the problem of generating from a multivariate distribution can be very hard. We mention, hereafter, few cases where generation is relativey easy.

2.6.1 Independent components

The simplest case is when the components $X_1, \dots, X_n$ of $\boldsymbol{X}$ are independent, each with cdf $F_i : \mathbb{R} \to [0, 1]$, so that $F(\boldsymbol{z}) = F_1(z_1) \cdots F_n(z_n)$. (Similarly, if each X_i has a density f_i, there holds $f(\boldsymbol{z}) = f_1(z_1) \cdots f_n(z_n)$.) In this case, each component X_i can be generated independently of the others by using any of the techniques described for univariate functions.

Example 2.5. *Suppose we would like to draw a point $\boldsymbol{X} = (X_1, X_2)$ that is uniformly distributed in the unit cube $(0, 1)^2$. Since $F(\boldsymbol{z}) = \mathbb{P}(X_1 \leq z_1, X_2 \leq z_2) = z_1 z_2$, we conclude that X_1, X_2 are independent and $X_i \sim F_i(z) = z$, i.e. each X_i is uniformly distributed in (0,1). We can then draw $X_1, X_2 \sim \mathcal{U}(0, 1)$ independently and set $\boldsymbol{X} = (U_1, U_2)$.*

Example 2.6. *Suppose now that we would like to draw a point $\boldsymbol{X} = (X_1, X_2)$ that is uniformly distributed on the unit ball $\mathcal{B} = \{(x, y) : x^2 + y^2 \leq 1\}$. One possibility is to use an acceptance-rejection method. For instance, we could draw $\boldsymbol{Y}$ uniformly on the cube $(-1, 1)^2$ and accept it by setting $\boldsymbol{X} = \boldsymbol{Y}$ only if $\boldsymbol{Y} \in \mathcal{B}$. The acceptance rate is $\frac{\pi}{4} \approx 78.5\%$. (Try to do it now in dimension $n \gg 2$ and see what happens ...)*

Alternatively, we could try to generate directly a point $\boldsymbol{X}$ with the correct distribution, without the acceptance-rejection step. For this, let us consider a transformation in polar coordinates, $X_1 = R\cos\Theta$, $X_2 = R\sin\Theta$. Then, if $f_{(X_1,X_2)}$ denotes the joint density of (X_1, X_2) and $f_{(R,\Theta)}$ the joint density of (R, Θ), we have $f_{(X_1,X_2)}(x, y) = \frac{1}{\pi}$, $(x, y) \in \mathcal{B}$ and

$$f_{(R,\Theta)}(\rho, \theta) = f_{(X_1,X_2)}(\rho\cos\theta, \rho\sin\theta) \left| \frac{\partial(x_1, x_2)}{\partial(\rho, \theta)} \right| = \frac{\rho}{\pi} = 2\rho\frac{1}{2\pi}.$$

We see then that (R, Θ) are independent with $\Theta \sim \mathcal{U}(0, 2\pi)$ and R having pdf $f_R(\rho) = 2\rho$ and cdf $F_R(\rho) = \rho^2$, which can be easily inverted. Therefore, starting from $U_1, U_2 \sim \mathcal{U}(0, 1)$ independent, we set $R = \sqrt{U_1}$ and $\Theta = 2\pi U_2$ so that $\boldsymbol{X} = (R\cos\Theta, R\sin\Theta)$ is a uniformly distributed point in the unit circle.

2.6.2 Generation from conditional distributions

Another situation which may lead to a relatively easy generation algorithm is when the marginal and univariate conditional distributions of $\boldsymbol{X}$ are easily accessible. For instance, let us assume that the conditional density of $X_j | X_{1:j-1}$,

$$f_{X_j|X_{1:j-1}}(z_j|z_1, \dots, z_{j-1}) = \frac{\int_{\mathbb{R}^{n-j}} f(z_1, \dots, z_j, z_{j+1}, \dots, z_n) dz_{j+1} \dots dz_n}{\int_{\mathbb{R}^{n-j+1}} f(z_1, \dots, z_j, z_{j+1}, \dots, z_n) dz_j dz_{j+1} \dots dz_n}$$

with $X_{1:j}$ a shorthand notation for $(X_1,\dots,X_j)$, is known explicitely for any $j=1,\dots,n$. Assume, moreover, that we know how to generate a variable from the density $f_{X_j|X_{1:j-1}}(\cdot \mid z_{1:j-1})$, for any $z_{1:j-1}\in\mathbb{R}^{j-1}$. We can then generate $\boldsymbol{X}$ with the following iterative Algorithm:

Algorithm 2.8: Generation from conditional distributions.

1 Generate $X_1 \sim f_{X_1}(z)$
2 For $i=2,\dots,n$,
3 Generate $X_i \sim f_{X_i \mid X_{1:i-1}}(\cdot \mid X_1,\dots,X_{i-1})$

Again, the generation of X_i from $f_{X_i \mid X_{1:i-1}}(\cdot \mid X_{1:i-1})$ can be done using any of the techniques available for univariate variables.

Example 2.7 (Generating order statistics)**.** *Let $\boldsymbol{X}=(X_1,\dots,X_n)\sim\mathcal{U}((0,1)^n)$ and denote by $X_{(1)}\le X_{(2)}\le\cdots\le X_{(n)}$ the ordered sample (order statistics). To generate $X_{(1)},\dots,X_{(n)}$, one can simply generate $\boldsymbol{X}=(X_1,\dots,X_n)$ with $X_i \overset{iid}{\sim} \mathcal{U}(0,1)$ and then order the components of the vector. However, if n is large, the 'sort' operation might become costly so one may prefer to generate directly $X_{(1)},\dots,X_{(n)}$ from the distribution of the order statistics.*

Observe that $X_{(n)}=\max_{i=1,\dots,n}X_i\sim F_{X_{(n)}}(z)=z^n$ so that $X_{(n)}$ can be generated easily by inversion as $X_{(n)}=(U_n)^{1/n}$ with $U_n\sim\mathcal{U}(0,1)$. Moreover, it can be shown (exercise) that for all $j<n$,

$$\begin{aligned}F_{X_{(j)} \mid X_{(j+1)},\dots,X_{(n)}}(z \mid x_{j+1:n}) &= \mathbb{P}\left(X_{(j)}\le z \mid X_{(j+1)}=x_{j+1},\dots,X_{(n)}=x_n\right)\\ &= \mathbb{P}\left(X_{(j)}\le z \mid X_{(j+1)}=x_{j+1}\right)=\left(\frac{z}{x_{j+1}}\right)^j, \quad z\le x_{j+1},\end{aligned}$$

where $F_{X_{(j)} \mid X_{(j+1)},\dots,X_{(n)}}(z \mid x_{j+1:n})=\int_0^z f_{X_{(j)} \mid X_{(j+1)},\dots,X_{(n)}}(t \mid x_{j+1:n})dt$ is the cumulative conditional distribution, which can be easily inverted. Hence, we can generate $X_{(j)}$ as $X_{(j)}=(U_j)^{\frac{1}{j}}X_{(j+1)}$ with $U_j\sim\mathcal{U}(0,1)$ independent of the previously generated ones.

Chapter 3

Generation of Gaussian random variables and processes

3.1 Generation of multivariate Gaussian random variables

A multivariate Gaussian random variable $\boldsymbol{X} \sim N(\boldsymbol{\mu}, \Sigma)$ with mean $\boldsymbol{\mu} \in \mathbb{R}^n$ and covariance matrix $\Sigma \in \mathbb{R}^{n\times n}$ symmetric and positive definite has joint pdf

$$f(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^n \det \Sigma}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right), \quad \boldsymbol{x} \in \mathbb{R}^n$$

and characteristic function $\phi(\boldsymbol{t}) = \mathbb{E}\left[e^{i\boldsymbol{t}^\top \boldsymbol{X}}\right] = \exp\left(i\boldsymbol{t}^\top\boldsymbol{\mu} - \frac{1}{2}\boldsymbol{t}^\top\Sigma\boldsymbol{t}\right)$. Notice that the characteristic function is well defined also in the case of a singular covariance matrix Σ, whereas the pdf is not.

The standard algorithm to generate $\boldsymbol{X}$ relies on explicit factorization of the covariance matrix as $\Sigma = AA^\top$, with $A \in \mathbb{R}^{n\times n}$, which can always be done since Σ is symmetric and positive definition. There are two common ways to compute the factor A:

- *Cholesky factorization.* It is applicable if Σ is strictly positive definite (hence invertible) and leads to a lower triangular factor A.

  If Σ is nearly singular, a more stable procedure is given by the *pivoted* Cholesky factorisation, which can also be used in the the singular case.

- *Spectral decomposition.* It is based on diagonalization of the covariance matrix as $\Sigma = VDV^\top$ with $D = \mathrm{diag}(\lambda_1, \dots, \lambda_n)$ the matrix of eigenvalues and V the orthonormal matrix of eigenvectors. We set then $A = VD^{1/2}$, with $D^{1/2} = \mathrm{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n})$.

Using either factorization, $\boldsymbol{X}$ can be generated by the following algorithm.

Algorithm 3.1: Multivariate Gaussian generator

Given: $\boldsymbol{\mu} \in \mathbb{R}^n$ and $\Sigma = AA^\top \in \mathbb{R}^{n\times n}$ (spd)

1 Generate $\boldsymbol{Y} \sim N(\boldsymbol{0}, I_{n\times n})$ (i.e. $\boldsymbol{Y} = (Y_1, \dots, Y_n)$, $Y_i \overset{\text{iid}}{\sim} N(0,1)$)

2 Compute $\boldsymbol{X} = \boldsymbol{\mu} + A\boldsymbol{Y}$

It is easy to check that $\boldsymbol{X}$ has the correct distribution. Indeed, $\boldsymbol{X}$ is Gaussian being an affine transformation of a standard normal vector. Moreover, $\mathbb{E}[\boldsymbol{X}] = \boldsymbol{\mu}$ and

$$\operatorname{Cov}[\boldsymbol{X}] = \mathbb{E}\left[(\boldsymbol{X}-\boldsymbol{\mu})(\boldsymbol{X}-\boldsymbol{\mu})^\top\right] = \mathbb{E}\left[A\boldsymbol{Y}\boldsymbol{Y}^\top A^\top\right] = A\mathbb{E}\left[\boldsymbol{Y}\boldsymbol{Y}^\top\right]A^\top = \Sigma.$$

The algorithm can be easily modified in the case the *precision matrix* $\Lambda = \Sigma^{-1}$ is given, instead of Σ. (Of course we assume here that Σ^{-1} is invertible.)

Algorithm 3.2: Multivariate Gaussian generator from precision matrix

Given: $\boldsymbol{\mu} \in \mathbb{R}^n$ and $\Lambda = \Sigma^{-1}$

1 Compute the Cholesky factorisation $\Lambda = LL^T$
2 Generate $\boldsymbol{Z} \sim N(\boldsymbol{0}, I)$ // n independent standard normals
3 Solve the linear system $L^\top \boldsymbol{Y} = \boldsymbol{Z}$ // upper triangular linear system
4 Output $\boldsymbol{X} = \boldsymbol{\mu} + \boldsymbol{Y}$

Again it is easy to verify that $\boldsymbol{X}$ has the right distribution. Indeed $\mathbb{E}[\boldsymbol{Y}] = L^{-\top}\mathbb{E}[\boldsymbol{Z}] = \boldsymbol{0}$ which implies $\mathbb{E}[\boldsymbol{X}] = \boldsymbol{\mu}$ and

$$\mathbb{E}\left[(\boldsymbol{X}-\boldsymbol{\mu})(\boldsymbol{X}-\boldsymbol{\mu})^\top\right] = \mathbb{E}\left[\boldsymbol{Y}\boldsymbol{Y}^T\right] = \mathbb{E}\left[L^{-\top}\boldsymbol{Z}\boldsymbol{Z}^\top L^{-1}\right] = L^{-\top}L^{-1} = \Lambda^{-1} = \Sigma.$$

3.2 Generation from conditional Gaussian distribution

Consider a multivariate Gaussian random variable $\boldsymbol{X} \in \mathbb{R}^n$, $\boldsymbol{X} \sim N(\boldsymbol{\mu}, \Sigma)$, which we split into two components, $\boldsymbol{X} = (Y_1, \ldots, Y_k, Z_1, \ldots, Z_{n-k}) = (\boldsymbol{Y}, \boldsymbol{Z})$ that we suppose unobservable and observable, respectively. The mean $\boldsymbol{\mu}$ and covariance Σ also split accordingly as

$$\mathbb{E}[\boldsymbol{X}] = \begin{pmatrix} \boldsymbol{\mu}_{\boldsymbol{Y}} \\ \boldsymbol{\mu}_{\boldsymbol{Z}} \end{pmatrix}, \qquad \Sigma = \begin{pmatrix} \Sigma_{\boldsymbol{YY}} & \Sigma_{\boldsymbol{YZ}} \\ \Sigma_{\boldsymbol{YZ}}^\top & \Sigma_{\boldsymbol{ZZ}} \end{pmatrix}$$

with $\boldsymbol{\mu}_{\boldsymbol{Y}} = \mathbb{E}[\boldsymbol{Y}]$, $\boldsymbol{\mu}_{\boldsymbol{Z}} = \mathbb{E}[\boldsymbol{Z}]$, $\Sigma_{\boldsymbol{YZ}} = \mathbb{E}\left[(\boldsymbol{Y}-\boldsymbol{\mu}_{\boldsymbol{Y}})(\boldsymbol{Z}-\boldsymbol{\mu}_{\boldsymbol{Z}})^\top\right] = \Sigma_{\boldsymbol{ZY}}^\top$, and similarly for $\Sigma_{\boldsymbol{YY}}$, $\Sigma_{\boldsymbol{ZZ}}$.

We are interested to generate the conditional random variable $\boldsymbol{Y}|\boldsymbol{Z} = \boldsymbol{z}$. It is well known that the conditional distribution of $\boldsymbol{Y}$ given $\boldsymbol{Z} = \boldsymbol{z}$ is again a multivariate Gaussian $N(\boldsymbol{\mu}_{\boldsymbol{Y}\,|\,\boldsymbol{Z}}, \Sigma_{\boldsymbol{Y}\,|\,\boldsymbol{Z}})$ with

$$\boldsymbol{\mu}_{\boldsymbol{Y}\,|\,\boldsymbol{Z}} = \boldsymbol{\mu}_{\boldsymbol{Y}} + \Sigma_{\boldsymbol{YZ}}\Sigma_{\boldsymbol{ZZ}}^{-1}(\boldsymbol{z} - \boldsymbol{\mu}_{\boldsymbol{z}}) \tag{3.1}$$

$$\Sigma_{\boldsymbol{Y}\,|\,\boldsymbol{Z}} = \Sigma_{\boldsymbol{YY}} - \Sigma_{\boldsymbol{YZ}}\Sigma_{\boldsymbol{ZZ}}^{-1}\Sigma_{\boldsymbol{ZY}}. \tag{3.2}$$

To generate $\boldsymbol{Y} \mid \boldsymbol{Z} = \boldsymbol{z}$ one can, of course, factorize the covariance matrix $\Sigma_{\boldsymbol{Y}\,|\,\boldsymbol{Z}} = AA^\top$ (by Cholesky or spectral decomposition). This, however, can be expensive or cumbersome if the size of $\boldsymbol{Y}$ is big. In particular, there might be cases (typically in the generation of stationary Gaussian random fields) in which generating $\boldsymbol{X} = (\boldsymbol{Y}, \boldsymbol{Z})$ is easy, even for very large dimensions, but generating $\boldsymbol{Y} \mid \boldsymbol{Z} = \boldsymbol{z}$ e.g. by Cholesky factorization, would be very costly. Here is an alternative algorithm to do so.

Algorithm 3.3: Generation from conditional Gaussian distribution

Given: $\boldsymbol{\mu}, \Sigma$ and $\boldsymbol{z} \in \mathbb{R}^{n-k}$

Generate $\boldsymbol{X} \sim N(\boldsymbol{\mu}, \Sigma)$

Set $\boldsymbol{\mathcal{Y}} = (X_1, \ldots, X_k)$ and $\boldsymbol{\mathcal{Z}} = (X_{k+1}, \ldots, X_n)$

Output $\boldsymbol{Y} = \boldsymbol{\mathcal{Y}} + \Sigma_{\boldsymbol{YZ}} \Sigma_{\boldsymbol{ZZ}}^{-1} (\boldsymbol{z} - \boldsymbol{\mathcal{Z}})$

Again, we can easily verify that $\boldsymbol{Y}$ has the correct distribution. Indeed,

$$\mathbb{E}[\boldsymbol{Y}] = \mathbb{E}[\boldsymbol{\mathcal{Y}}] + \Sigma_{\boldsymbol{YZ}} \Sigma_{\boldsymbol{ZZ}}^{-1} (\boldsymbol{z} - \mathbb{E}[\boldsymbol{\mathcal{Z}}]) = \boldsymbol{\mu}_{\boldsymbol{Y}} + \Sigma_{\boldsymbol{YZ}} \Sigma_{\boldsymbol{ZZ}}^{-1} (\boldsymbol{z} - \boldsymbol{\mu}_{\boldsymbol{Z}}).$$

Moreover, setting $\boldsymbol{Y}' = \boldsymbol{Y} - \mathbb{E}[\boldsymbol{Y}] = (\boldsymbol{\mathcal{Y}} - \boldsymbol{\mu}_{\boldsymbol{Y}}) - \Sigma_{\boldsymbol{YZ}} \Sigma_{\boldsymbol{ZZ}}^{-1} (\boldsymbol{\mathcal{Z}} - \mu_{\boldsymbol{Z}})$, we have

$$\begin{aligned}
\operatorname{Cov}(\boldsymbol{Y}) = \mathbb{E}\left[\boldsymbol{Y}'\boldsymbol{Y}'^{\top}\right] &= \mathbb{E}\left[\boldsymbol{\mathcal{Y}}'\boldsymbol{\mathcal{Y}}'^{\top}\right] - \Sigma_{\boldsymbol{YZ}} \Sigma_{\boldsymbol{ZZ}}^{-1} \mathbb{E}\left[\boldsymbol{\mathcal{Z}}'\boldsymbol{\mathcal{Y}}'^{\top}\right] \\
&\quad - \mathbb{E}\left[\boldsymbol{\mathcal{Y}}'\boldsymbol{\mathcal{Z}}'^{T}\right] \Sigma_{\boldsymbol{ZZ}}^{-\top} \Sigma_{\boldsymbol{YZ}}^{\top} + \Sigma_{\boldsymbol{YZ}} \Sigma_{\boldsymbol{ZZ}}^{-1} \mathbb{E}\left[\boldsymbol{\mathcal{Z}}'\boldsymbol{\mathcal{Z}}'^{T}\right] \Sigma_{\boldsymbol{ZZ}}^{-\top} \Sigma_{\boldsymbol{YZ}}^{\top} \\
&= \Sigma_{\boldsymbol{YY}} - \Sigma_{\boldsymbol{YZ}} \Sigma_{\boldsymbol{ZZ}}^{-1} \Sigma_{\boldsymbol{ZY}}.
\end{aligned}$$

3.3 Gaussian process generation

Let $I \subset \mathbb{R}^d$. A collection of random variables $\{X_t,\ t \in I\}$ indexed by $t \in I$ is called a *stochastic process* when $d = 1$ (usually t denotes time) or a *random field* if $d \geq 1$ and t denotes the space variable.

Definition 3.1 (Gaussian process)**.** *A Gaussian process (or Gaussian random field) is a stochastic process (random field) for which all finite dimensional distributions are Gaussian, i.e. for all $n \in \mathbb{N}$ and $t_1, \ldots, t_n \in I$, the random vector $\boldsymbol{X} = (X_{t_1}, \ldots, X_{t_n})$ has a multivariate Gaussian distribution. Equivalently, any linear combination $Y_{\boldsymbol{b}} = \sum_{i=1}^n b_i X_{t_i}$ has a Gaussian distribution.*

Given a Gaussian process $\{X_t,\ t \in I\}$, we can define

Mean function : $\mu_X : I \to \mathbb{R}$, $\mu_X(t) = \mathbb{E}[X_t],\ t \in I$,

Covariance funct. : $C_X : I \times I \to \mathbb{R}$, $C_X(t,s) = \mathbb{E}[(X_t - \mu_X(t))(X_s - \mu_X(s))],\ t, s \in I$.

If we now take a set of points $t_1, \ldots, t_n \in I$ and consider the Gaussian random vector $\boldsymbol{X} = (X_{t_1}, \ldots, X_{t_n})$, it clearly holds $\boldsymbol{X} \sim N(\boldsymbol{\mu}, \Sigma)$, with $\boldsymbol{\mu} = (\mu_X(t_1), \ldots, \mu_X(t_n))$ and $\Sigma_{ij} = C_X(t_i, t_j)$. As such, the matrix Σ has to be symmetric and non negative definite. This poses restrictions to the class of functions that can be covariance functions of a stochastic process.

Definition 3.2. *A function $C : I \times I \to \mathbb{R}$ is* non-negative definite *if, for all n and $t_1, \ldots, t_n \in I$, the matrix $\Sigma \in \mathbb{R}^{n \times n}$, $\Sigma_{ij} = C(t_i, t_j)$ is non-negative definite.*

Proposition 3.1. *A Gaussian process $\{X_t, t \in I\}$ is uniquely determined by the mean function $\mu_X : I \to \mathbb{R}$ and a* symmetric *and non-negative definite covariance function $C_X : I \times I \to \mathbb{R}$.*

We use the notation $X \sim N(\mu_X, C_X)$ to denote a Gaussian process $\{X_t, t \in I\}$ with mean function μ_X and covariance function C_X.

A Gaussian process $X \sim N(\mu_X, C_X)$ can be generated exactly in a set of points $t_1, \ldots, t_n \in I$ by generating the corresponding random vector $\boldsymbol{X} \sim N(\boldsymbol{\mu}, \Sigma)$, with $\boldsymbol{\mu} = (\mu_X(t_1), \ldots, \mu_X(t_n))$ and $\Sigma_{ij} = C_X(t_i, t_j)$. This can be done by either Cholesky or spectral factorization of the matrix Σ.

Similarly, assume that we have generated already $\boldsymbol{Z} = (X_{t_1}, \ldots, X_{t_n})$ and we want to generate new values $\boldsymbol{Y} = (X_{t_{n+1}}, \ldots, X_{t_m})$ conditional to the previously generated ones. This can be done by the Algorithm 3.3 illustrated in the previous section. Figure 3.1 gives a graphical interpretation of the procedure.

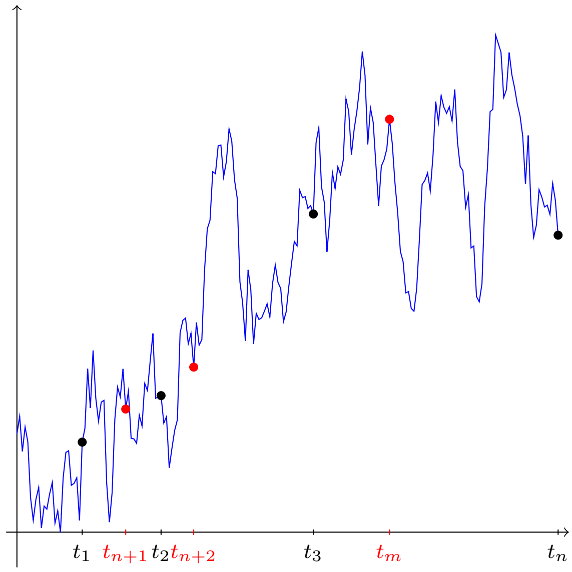


Figure 3.1: Conditioned Gaussian Process.

3.3.1 Wiener process (Brownian motion)

Definition 3.3. *The Wiener process is a stochastic process $\{W_t, t \geq 0\}$ with the following properties:*

- *Independent increments: for all $0 < t_1 < t_2 \leq t_3 < t_4$, $(W_{t_2} - W_{t_1})$ and $(W_{t_4} - W_{t_3})$ are independent random variables*
- *Gaussian stationary increments: for all $0 \leq t_1 \leq t_2$, $W_{t_2} - W_{t_1} \sim N(0, t_2 - t_1)$*
- *Path continuity: $\{W_t\}$ has continuous paths with $W_0 = 0$.*

The Wiener process is a Gaussian process with mean function $\mu_W(t) = 0$ and covariance function $\mathrm{Cov}_W(s,t) = \min\{s,t\}$. Indeed, if $t \geq s$, $\mathrm{Cov}_W(s,t) = \mathbb{E}[W_s W_t] = \mathbb{E}[W_s(W_t - W_s)] + \mathbb{E}\left[W_s^2\right] = s$. Similarly, if $t \leq s$ then $\mathrm{Cov}_W(s,t) = t$.

To generate $\{W_t, \, t \geq 0\}$ on a set of points $(t_1, \ldots, t_n)$, one could either compute a Cholesky/spectral decomposition of the covariance matrix $\Sigma_{ij} = \min\{t_i, t_j\}$, or, more

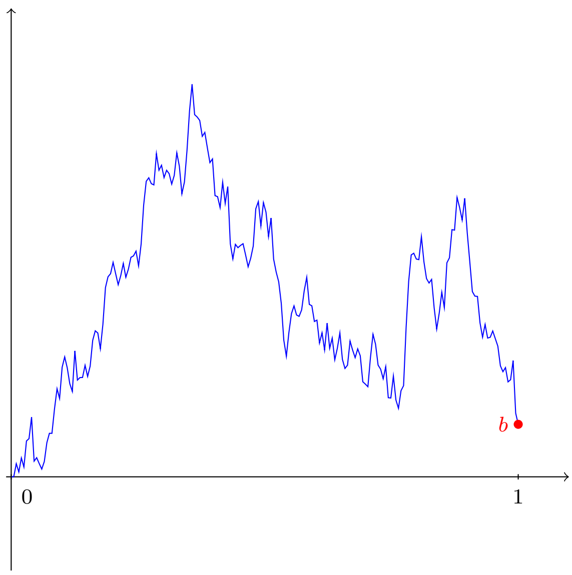


Figure 3.2: Brownian bridge.

efficiently, rely on the property of independent Gaussian increments as the following Algorithm shows.

Algorithm 3.4: Wiener process generation

```
Set t_0 = 0 and W_{t_0} = 0
for k = 1, ..., n do
    Generate ΔW_k ~ N(0, t_k − t_{k−1})
    Set W_{t_k} = W_{t_{k−1}} + ΔW_k
end
```

A Brownian motion with drift ν and diffusion coefficient σ^2, is the solution of the stochastic differential equation

$$dB_t = \nu\, dt + \sigma dW_t, \quad B_0 = 0$$

that is, $B_t = \nu t + \sigma W_t,\ t \geq 0$. Hence, it can be easily generated on a set of points $(t_1, \dots, t_n)$ as an affine transformation of the Wiener process.

3.3.2 Brownian bridge

A Brownian bridge process $\{X_t, t \in [0,1]\}$ is a Wiener process $\{W_t, t \in [0,1]\}$ conditioned upon $W_1 = b$. See figure 3.2 for a realization of a Brownian bridge.

The conditional mean and covariance function of a Brownian bridge can be calculated using the standard formulas for conditioned multivariate Gaussian variables. Indeed, let us first calculate the conditinal mean. For that, we set $Y = W_t$, $t \in (0,1)$ and $Z = W_1$, for which we have $\Sigma_{YY} = \Sigma_{YZ} = t$ and $\Sigma_{ZZ} = 1$. Therefore, using formula (3.1) for the conditional mean, we conclude that

$$\mu_X(t) = \mathbb{E}\left[X_t\right] = \mathbb{E}\left[W_t \mid W_1 = b\right] = \mu_W(t) + \Sigma_{YZ}\Sigma_{ZZ}^{-1}(b - \mu_W(1)) = tb.$$

An analogous procedure can be followed to compute the conditional covariance. Take this time $Y = (W_s, W_t)$, $s, t \in (0, 1)$ and $Z = W_1$, so that

$$\Sigma_{YY} = \begin{pmatrix} s & \min\{s,t\} \\ \min\{s,t\} & t \end{pmatrix}, \quad \Sigma_{YZ} = \begin{pmatrix} s \\ t \end{pmatrix}, \quad \Sigma_{ZZ} = 1.$$

Therefore

$$\begin{aligned}\Sigma_{Y\mid Z} &= \Sigma_{YY} - \Sigma_{YZ}\Sigma_{ZZ}^{-1}\Sigma_{ZY} \\ &= \begin{pmatrix} s & \min\{s,t\} \\ \min\{s,t\} & t \end{pmatrix} - \begin{pmatrix} s \\ t \end{pmatrix}(s,t) = \begin{pmatrix} s - s^2 & \min\{s,t\} - st \\ \min\{s,t\} - st & t - t^2 \end{pmatrix}.\end{aligned}$$

and

$$\mathrm{Cov}_X(s,t) = \mathrm{Cov}(W_s, W_t | W_1 = b) = (\Sigma_{Y|Z})_{12} = \min\{s,t\} - st.$$

To generate a Brownian bridge in a set of points $0 < t_1 < \ldots < t_n < t_{n+1} = 1$ one can first generate $(W_{t_1}, \ldots, W_{t_n}, W_{t_{n+1}})$ from a standard Wiener process and then use Algorithm 3.3 with $\boldsymbol{\mathcal{Y}} = (W_{t_1}, \ldots, W_{t_n})$ and $\mathcal{Z} = W_{t_{n+1}}$. This leads to the following procedure.

Algorithm 3.5: Brownian bridge generation.

Given: $0 < t_1 < \cdots < t_n < t_{n+1} = 1$ and b
1. Generate W_{t_i}, $i = 1, \ldots, n+1$ from standard Wiener process
2. Output $X_{t_i} = W_{t_i} + t_i(b - W_{t_{n+1}})$, $i = 1, \ldots, n$.

3.4 Stationary Gaussian processes / random fields

Definition 3.4. *A Gaussian process $\{X_t, t \in \mathbb{R}\}$ is weakly stationary if $C_X(t,s)$ depends only on $(s-t)$ and is (strongly) stationary if it is weakly stationary and $\mu_X(t)$ does not depend on t.*

A weakly stationary Gaussian process can be generated very efficiently on a uniform grid $\{t_j = t_0 + jh, j = 0, \ldots, n\}$ with the use of FFT. This avoids the costly step of computing the Cholesky or spectral decomposition of the covariance matrix. We denote by $\boldsymbol{X} = (X_{t_0}, \ldots, X_{t_n})$ the discrete Gaussian process on the uniform grid. Since $\{t_j, j = 0, \ldots, n\}$ is a uniform grid, it follows that the corresponding covariance matrix

$$\Sigma_{ij} = C_X(t_i, t_j) = C_X(0, (j-i)h)$$

depends only on $j - i$, hence is a symmetric Toeplitz matrix and we have to store only the vector $(\sigma_0, \ldots, \sigma_n)$ with $\sigma_i = C_X(t_0, t_0 + ih)$:

$$\Sigma = \begin{pmatrix} \sigma_0 & \sigma_1 & \sigma_2 & \cdots & & \sigma_n \\ \sigma_1 & \sigma_0 & \sigma_1 & \cdots & & \sigma_{n-1} \\ \sigma_2 & \sigma_1 & \ddots & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & & \vdots \\ \vdots & \vdots & & & \ddots & \vdots \\ \sigma_n & \sigma_{n-1} & \cdots & & \sigma_1 & \sigma_0 \end{pmatrix}$$

Consider now the following circulant embedding of Σ:

$$\tilde{\Sigma} = \left(\begin{array}{cccccc|cccc}
\sigma_0 & \sigma_1 & \sigma_2 & \dots & \sigma_{n-1} & \sigma_n & \sigma_{n-1} & \sigma_{n-2} & \dots & \sigma_1 \\
\sigma_1 & \sigma_0 & \sigma_1 & \dots & & \sigma_{n-1} & \sigma_n & \sigma_{n-1} & \dots & \sigma_2 \\
\sigma_2 & \sigma_1 & \ddots & \ddots & & \vdots & \vdots & \ddots & \ddots & \vdots \\
\vdots & \vdots & \ddots & \ddots & & \vdots & \vdots & & \ddots & \sigma_{n-1} \\
\vdots & \vdots & & & \ddots & \sigma_1 & \sigma_2 & & & \sigma_n \\
\sigma_n & \sigma_{n-1} & \dots & & \sigma_1 & \sigma_0 & \sigma_1 & \dots & \dots & \sigma_{n-1} \\
\hline
\sigma_{n-1} & \sigma_n & \sigma_{n-1} & \dots & \dots & \sigma_1 & \sigma_0 & \sigma_1 & \dots & \sigma_{n-2} \\
\sigma_{n-2} & \sigma_{n-1} & \ddots & \ddots & & \sigma_0 & \sigma_1 & \sigma_0 & \dots & \vdots \\
\vdots & & \ddots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\
\sigma_1 & \dots & \dots & \sigma_{n-1} & \sigma_n & \sigma_{n-1} & \sigma_{n-2} & & \sigma_1 & \sigma_0
\end{array}\right) \in \mathbb{R}^{2n \times 2n}$$

and the generator vector $\boldsymbol{\alpha} = (\sigma_0, \sigma_1, \dots, \sigma_n, \sigma_{n-1}, \dots, \sigma_1)$ given by the first column. We write compactly $\tilde{\Sigma} = \mathrm{circ}(\boldsymbol{\alpha})$ and assume that $\tilde{\Sigma}$ is also non-negative definite.

Lemma 3.2. *Let $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_{2n}) \in \mathbb{R}^{2n}$ and $\tilde{\Sigma} = circ(\boldsymbol{\alpha})$. Then the vectors $\boldsymbol{v}^{(\ell)}$, $\ell = 1, \dots, 2n$, $v_k^{(\ell)} = e^{2\pi i(\ell-1)(k-1)/2n}$ are eigenvectors of $\tilde{\Sigma}$ with corresponding eigenvalues $\lambda_\ell = \sum_{k=1}^{2n} \alpha_k e^{-2\pi i(\ell-1)(k-1)/2\pi}$, which are real and non-negative if $\tilde{\Sigma}$ is semi positive definite.*

Proof. It is enough to verify that $\sum_{k=1}^{2n} \tilde{\Sigma}_{jk} v_k^{(\ell)} = \lambda_\ell v_j^{(\ell)}$, for all $j = 1, \dots, 2n$. Notice that $\tilde{\Sigma}$ can be written as $\tilde{\Sigma}_{jk} = \alpha_{\{(2n+j-k+1) \mod 2n\}}$, where we set $\alpha_0 = \alpha_{2n}$. Then

$$\begin{aligned}
\sum_{k=1}^{2n} \tilde{\Sigma}_{jk} v_k^{(\ell)} &= \sum_{k=1}^{2n} \alpha_{\{(2n+j-k+1) \mod 2n\}} e^{2\pi i(\ell-1)(k-1)/2n} \\
&= \sum_{k=1}^{2n} \alpha_{\{(2n+j-k+1) \mod 2n\}} e^{2\pi i(\ell-1)(k-j-2n)/2n} e^{2\pi i(\ell-1)(2n+j-1)/2n} \\
&= \left(\sum_{k=1}^{2n} \alpha_k e^{-2\pi i(\ell-1)(k-1)/2n} \right) v_j^{(\ell)}.
\end{aligned}$$

$\square$

It follows from this Lemma that the matrix $\tilde{\Sigma}$ can be diagonalized as $\tilde{\Sigma} F^* = F^* \Lambda$ where $F_{k\ell} = e^{-2\pi i(\ell-1)(k-1)/2n}$ corresponds to the FFT matrix and $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_{2n})$. Moreover, the vector of eigenvalues corresponds to $\boldsymbol{\lambda} = F\boldsymbol{\alpha} = \mathrm{FFT}(\boldsymbol{\alpha})$. Observe that $F^*F = FF^* = 2nI_{2n}$ so that $\tilde{\Sigma} = \frac{1}{2n} F^* \Lambda F$ and can be factorized as $\tilde{\Sigma} = AA^*$ with $A = \frac{1}{\sqrt{2n}} F^* \Lambda^{1/2}$. We consider now a vector $\boldsymbol{Y} = (Y_1, \dots, Y_{2n})$ of *complex standard normal r.v.s*, i.e. $\boldsymbol{Y} = \boldsymbol{Y}_R + i\boldsymbol{Y}_I$ with $\boldsymbol{Y}_R, \boldsymbol{Y}_I \overset{\text{iid}}{\sim} N(0, I_{2n})$, and set

$$\tilde{\boldsymbol{X}} = A\boldsymbol{Y} = \frac{1}{\sqrt{2n}} F^* \Lambda^{1/2} = \mathrm{iFFT}(\sqrt{2\pi}\Lambda^{1/2}\boldsymbol{Y}),$$

where the iFFT matrix is given by $\frac{1}{2n} F^*$. The following holds:

- $\mathbb{E}\left[\boldsymbol{Y}\boldsymbol{Y}^*\right] = 2I_{2n}, \quad \mathbb{E}\left[\boldsymbol{Y}\boldsymbol{Y}^\top\right] = \mathbb{E}\left[\bar{\boldsymbol{Y}}\bar{\boldsymbol{Y}}^\top\right] = 0,$
- $\mathbb{E}\left[\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^*\right] = \mathbb{E}\left[A\boldsymbol{Y}\boldsymbol{Y}^*A^*\right] = 2\tilde{\Sigma}, \quad \mathbb{E}\left[\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^\top\right] = 0,$
- $\mathbb{E}\left[\mathrm{Re}(\tilde{\boldsymbol{X}})\,\mathrm{Re}(\tilde{\boldsymbol{X}})^\top\right] = \mathbb{E}\left[\frac{\tilde{\boldsymbol{X}}+\bar{\tilde{\boldsymbol{X}}}}{2}\left(\frac{\tilde{\boldsymbol{X}}+\bar{\tilde{\boldsymbol{X}}}}{2}\right)^\top\right] = \tilde{\Sigma} = \mathbb{E}\left[\mathrm{Im}(\tilde{\boldsymbol{X}})\,\mathrm{Im}(\tilde{\boldsymbol{X}})^\top\right],$
- $\mathbb{E}\left[\mathrm{Re}(\tilde{\boldsymbol{X}})\,\mathrm{Im}(\tilde{\boldsymbol{X}})^\top\right] = \mathbb{E}\left[\frac{\tilde{\boldsymbol{X}}+\bar{\tilde{\boldsymbol{X}}}}{2}\left(\frac{\tilde{\boldsymbol{X}}-\bar{\tilde{\boldsymbol{X}}}}{2i}\right)^\top\right] = 0.$

Hence $\mathrm{Re}(\tilde{\boldsymbol{X}})$, $\mathrm{Im}(\tilde{\boldsymbol{X}}) \sim N(0,\tilde{\Sigma})$ and are independent and $\mathrm{Re}(\tilde{\boldsymbol{X}}_{1:n+1})$, $\mathrm{Im}(\tilde{\boldsymbol{X}}_{1:n+1}) \overset{\text{iid}}{\sim} N(0,\Sigma)$. This suggests the following algorithm to generate $\boldsymbol{X} \sim N(\boldsymbol{\mu},\Sigma)$.

Algorithm 3.6: Circulant embedding.

Given: $\mu \in \mathbb{R}^n$ and $\Sigma = \begin{pmatrix} \sigma_0 & \sigma_1 & \dots & \sigma_n \\ \sigma_1 & \sigma_0 & \dots & \sigma_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_n & \dots & \dots & \sigma_0 \end{pmatrix} \in \mathbb{R}^{n+1\times n+1}$

1. Generate the vector $\boldsymbol{\alpha} = (\sigma_0, \sigma_1, \dots, \sigma_n, \sigma_{n-1}, \dots, \sigma_1) \in \mathbb{R}^{2n}$
2. Compute $\boldsymbol{\lambda} = \mathrm{FFT}(\boldsymbol{\alpha})$
3. Generate $\boldsymbol{Y} = \boldsymbol{Y}_R + i\boldsymbol{Y}_I$ with $\boldsymbol{Y}_R, \boldsymbol{Y}_I \overset{\text{iid}}{\sim} N(0, I_{2n})$
4. Compute $\tilde{\boldsymbol{X}} = \mathrm{iFFT}(\sqrt{2n}\,\mathrm{diag}(\sqrt{\boldsymbol{\lambda}})\boldsymbol{Y})$
5. Output $\boldsymbol{X}^{(1)} = \boldsymbol{\mu} + \mathrm{Re}(\tilde{\boldsymbol{X}}_{1:n+1})$ and $\boldsymbol{X}^{(2)} = \boldsymbol{\mu} + \mathrm{Im}(\tilde{\boldsymbol{X}}_{1:n+1})$

One may encounter the problem that the matrix $\tilde{\Sigma}$ might not be semi positive definite, even if Σ is. In such a case, one could try to enlarge the circulant embedding

$$\boldsymbol{\alpha} = (\sigma_0, \sigma_1, \dots, \sigma_n, \sigma^*_{n+1}, \dots, \sigma^*_m, \sigma^*_{m-1}, \dots, \sigma^*_{n+1}, \sigma_n, \dots, \sigma_1)$$

where $m > n$ and σ^*_j, $j = n+1, \dots, m$ are chosen such that $\tilde{\Sigma} = \mathrm{circ}(\boldsymbol{\alpha})$ is semi positive definite. A typical choice is to take $\sigma^*_j = \sigma_j = C_X(0, jh)$ and m large enough.

Chapter 4

Generation of Markov processes

4.1 Discrete time / discrete state Markov chains

Let us consider a stochastic process $\{X_n,\, n \in \mathbb{N}_0\}$ defined on the countable set $\mathbb{N}_0 = \{0, 1, \dots\}$ and taking values in a countable set $\mathcal{X} = \{y_1, y_2, \dots\}$ i.e. $X_n \in \mathcal{X}$ for all $n \in \mathbb{N}_0$.

Definition 4.1. *A stochastic process $\{X_n \in \mathcal{X},\, n \in \mathbb{N}_0\}$ is a Markov chain if it satisfies the Markov property*

$$\mathbb{P}\left(X_{n+1} = y_{n+1} \mid X_n = y_n, X_{n-1} = y_{n-1}, \dots, X_0 = y_0\right) = \mathbb{P}\left(X_{n+1} = y_{n+1} \mid X_n = y_n\right)$$

with $y_0, \dots, y_{n+1} \in \mathcal{X}$.

The process is therefore entirely defined by the distribution λ of the initial state X_0 and the transition matrices

$$P(n) = (P_{ij}(n))_{ij}, \quad \text{with } P_{ij}(n) = \mathbb{P}\left(X_n = y_j \mid X_{n-1} = y_i\right).$$

which are, in particular, *stochastic matrices* i.e. they satisfy

$$\sum_j P_{ij}(n) = 1, \quad \forall i = 1, 2, \dots, \quad \forall n \in \mathbb{N}_0$$

A Markov chain is time-homogeneous if $P(n)$ does not depend on n. Generating a discrete time / discrete state Markov chain is rather straightforward.

Algorithm 4.1: Generation of discrete time / discrete space Markov process.

Given: λ and $P(n)$, $n \in \mathbb{N}_0$

Generate $X_0 \sim \lambda$

For $n = 1, 2, \dots,$

    Generate $X_n \sim P_{X_{n-1},\cdot}(n)$ `// pmf given by` X_{n-1}`-th row of` $P(n)$

Exercise 4.1 (Random walk on a lattice)**.** *A random walk on the integers* $\{X_n \in \mathbb{Z},\ n \in \mathbb{N}_0\}$*, starting at* $X_0 = 0$ *is a Markov chain defined by the following transition probabilities*

$$\begin{aligned}
&\mathbb{P}\left(X_{n+1} = j \mid X_n = j-1\right) = \mathbb{P}\left(X_{n+1} = j \mid X_n = j+1\right) = a \in (0,1),\\
&\mathbb{P}\left(X_{n+1} = j \mid X_n = j\right) = 1 - 2a,\\
&\mathbb{P}\left(X_{n+1} = j \mid X_n = i\right) = 0, \quad i \neq j, j-1, j+1.
\end{aligned}$$

Figure 4.1 shows a graph representation of the Markov chain. An arrow between two states denotes a connection, i.e. a non zero probability of moving from the base to the head of the arrow.

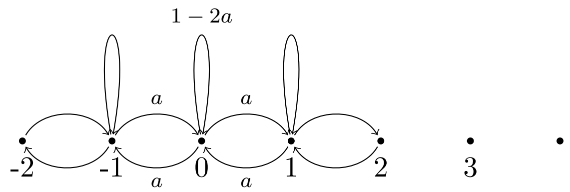


Figure 4.1: Random walk on lattice.

4.2 Discrete time / continuous state Markov chains

Consider now a stochastic process $\{X_n,\ n \in \mathbb{N}_0\}$ defined on $\mathbb{N}_0 = \{0, 1, \dots\}$ and taking values on a continuous set $\mathcal{X} \subset \mathbb{R}^d$. We denote by $\mathcal{B}(\mathcal{X})$ the Borel σ-algebra on $\mathcal{X}$ so that $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ is a measurable space.

Definition 4.2. *A Markov transition kernel on* $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ *is a function* $P : \mathcal{X} \times \mathcal{B}(\mathcal{X}) \to [0, 1]$ *such that*

- *for all* $y \in \mathcal{X}$*,* $P(y, \cdot)$ *is a probability measure on* $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$*;*
- *for all* $A \in \mathcal{B}(\mathcal{X})$*,* $P(\cdot, A)$ *is a measurable function on* $\mathcal{X}$*.*

Often, the transition kernel is defined starting from a density function $p : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ such that for all $x \in \mathcal{X}$, $A \in \mathcal{B}(\mathcal{X})$, $P(x, A) = \int_A p(x, y)\, dy$.

Definition 4.3. *Given a Markov transition kernel* $P : \mathcal{X} \times \mathcal{B}(\mathcal{X}) \to [0, 1]$*, a stochastic process* $\{X_n,\ n \in \mathbb{N}_0\}$ *with values in* $\mathcal{X}$ *is a homogeneous Markov chain with kernel* P *and initial distribution* $X_0 \sim \lambda$*, denoted* $\{X_n\} \sim$ *Markov*(λ, P)*, if for any* $n \in \mathbb{N}$*,* $A \in \mathcal{B}(\mathcal{X})$*,*

$$\mathbb{P}\left(X_{n+1} \in A \mid X_n = y_n, \dots, X_0 = y_0\right) = \mathbb{P}\left(X_{n+1} \in A \mid X_n = y_n\right) = P(y_n, A).$$

Again, generating a discrete time / continuous state Markov chain is rather straightforward, provided we know how to generate random variables from the probability measure $P(y, \cdot)$ (resp. probability density function $p(y, \cdot)$) for all $y \in \mathcal{X}$.

Algorithm 4.2: Generation of discrete time / continuous space Markov process.

Given: λ and P

Generate $X_0 \sim \lambda$

For $n = 1, 2, \dots,$

  Generate $X_n \sim P(X_{n-1}, \cdot)$

Exercise 4.2 (Random walk in 2D)**.** *Let $\mathcal{X} = \mathbb{R}^2$ and consider the stochastic process $\{\boldsymbol{X}_n \in \mathcal{X},\ n \in \mathbb{N}_0\}$ starting at $\boldsymbol{X}_0 = (0,0)$, defined by*

$$\boldsymbol{X}_{n+1} = \boldsymbol{X}_n + \boldsymbol{\xi}_n, \qquad \boldsymbol{\xi}_n \overset{iid}{\sim} N(0, \sigma^2 I_2).$$

This is clearly a homogeneous discrete time / continuous state Markov chain with transition kernel

$$P(\boldsymbol{y}, A) = \mathbb{P}\left(\boldsymbol{X}_{n+1} \in A \mid \boldsymbol{X}_n = \boldsymbol{y}\right) = \mathbb{P}\left(\boldsymbol{\xi_n} + \boldsymbol{y} \in A\right) = \frac{1}{2\pi\sigma^2} \int_A e^{-\frac{\|\boldsymbol{\xi}-\boldsymbol{y}\|^2}{2\sigma^2}}\, d\boldsymbol{\xi}$$

and transition density function $p(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(y_1-x_1)^2+(y_2-x_2)^2}{2\sigma^2}\right)$.

4.3 Continuous time / discrete state Markov chains

Let $\mathcal{X} = \{y_1, y_2, \dots\}$ be a discrete (finite or countable) set and $\{X_t,\ t \geq 0\}$ a stochastic process taking values in $\mathcal{X}$. The process is said to be right continuous if each path is so, i.e. for any realization ω,

$$\lim_{h \to 0^+} X_{t+h}(\omega) = X_t(\omega).$$

Since the process takes only discrete values, the right continuity property implies that if $X_t = y_i$ at some t, it will stay in state y_i for a certain amount of time, i.e. there exists a (random) $\varepsilon > 0$ s.t. $X_s = y_i$, for all $t \leq s < t + \varepsilon$. We denote J_n the n-th jump time

$$J_0 = 0, \qquad J_n = \inf\{t \geq J_{n-1} : X_t \neq X_{J_{n-1}}\}, \quad n > 0$$

and S_n the n-th holding time

$$S_n = \begin{cases} J_n - J_{n-1}, & \text{if } J_{n-1} < \infty, \quad n = 1, 2, \dots \\ \infty, & \text{otherwise.} \end{cases}$$

The discrete time process $\{Y_n = X_{J_n},\ n \in \mathbb{N}_0\}$ is called the *jump process* (or jump chain) of $\{X_t, t \geq 0\}$. The process is therefore completely characterized by the sequence $\{J_n\}_n$ of jump times (equivalently the sequence $\{S_n\}_n$ of holding times) as well as the sequence $\{Y_n\}_n$ of visited states, i.e. the jump chain. Figure 4.2 gives an illustration of a continuous time / discrete state Markov process.

The (first) explosion time T^* is defined as $T^* = \sup_n J_n = \sum_{n=1}^{\infty} S_n$. If $T^* < +\infty$, we consider only the process $\{X_t,\ t \in [0, T^*)\}$ or, equivalently, we set $X_t = \infty$ for $t \geq T^*$. This is called the *minimal process*.

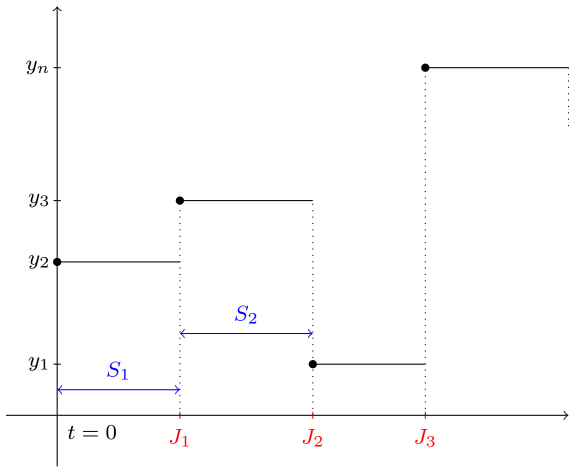


Figure 4.2: Continuous time / discrete state Markov process and associated jump and holding times.

4.4 Poisson process

The Poisson process is the simplest example of a continuous time / discrete state *Markov* process.

Definition 4.4. *A Poisson process* $\{N_t \in \mathbb{N}_0, t \geq 0\}$ *with initial state* $N_0 = 0$ *and parameter* $0 < \lambda < \infty$*, is a non decreasing, right-continuous, integer valued process which satisfies the following properties.*

1. *Independent increments: for all* $0 < t_1 < t_2 \leq t_3 < t_4$*,* $N_{t_2} - N_{t_1}$ *and* $N_{t_4} - N_{t_3}$ *are independent;*

2. *Poisson stationary increments: for all* $0 < s < t$*,* $N_t - N_s \sim \mathrm{Pois}(\lambda(t-s))$ *i.e.*

$$\mathbb{P}\left(N_t - N_s = j\right) = \frac{(\lambda(t-s))^j}{j!} e^{-\lambda(t-s)}.$$

It follows, in particular, that $N_t \sim \mathrm{Pois}(\lambda t)$. Moreover, N_t satisfies the Markov property: for any $s \geq 0$, $\tilde{N}_t = N_{s+t} - N_s$, $t \geq 0$ is also a Poisson process of rate λ, independent of $\{N_t,\ t \leq s\}$, as well as the strong Markov property where s is replaced by a stopping time T. (T is a stopping time if the event $\{T \leq t\}$ is measurable with respect to the σ-algebra $\mathcal{F}_t$ generated by $\{N_s, s \leq t\}$). The following are two equivalent characterizations of a Poisson process:

a. For any $t > 0$ and $h \to 0^+$, uniformly in t it holds

$$\begin{aligned}
\mathbb{P}\left(N_{t+h} - N_t = 0\right) &= 1 - \lambda h + o(h),\\
\mathbb{P}\left(N_{t+h} - N_t = 1\right) &= \lambda h + o(h),\\
\mathbb{P}\left(N_{t+h} - N_t > 1\right) &= o(h).
\end{aligned}$$

The last condition is actually a consequence of the first two.

b. The holding times $S_1, S_2, \dots$ are independent exponential random variables $\mathrm{Exp}(\lambda)$ and the jump chain is $Y_n = N_{J_n} = n$.

The first property follows immediately from the Poisson distribution of the increments. For the second proprety, observe that $\mathbb{P}(S_1 > t) = \mathbb{P}(N_t = 0) = e^{-\lambda t}$ hence $S_1 \sim \mathrm{Exp}(\lambda)$. Similarly, $\mathbb{P}(S_{n+1} > t) = \mathbb{P}(N_{J_n+t} - N_{J_n} = 0) = e^{-\lambda t}$ so $S_{n+1} \sim \mathrm{Exp}(\lambda)$ and independent of $S_1, \dots, S_n$ by the property of independent increments of N_t. The second property suggests an easy algorithm to generate a Poisson process with parameter λ.

Algorithm 4.3: Homogeneous Poisson process I.

Set $N_0 = 0,\ J_0 = 0,\ Y_0 = 0$
For $n = 1, 2, \dots,$
    Generate $S_n \sim \mathrm{Exp}(\lambda)$ and set $J_n = J_{n-1} + S_n$
    Set $N_t = N_{J_{n-1}},\ t \in [J_{n-1}, J_n)$ and $N_{J_n} = N_{J_{n-1}} + 1$.

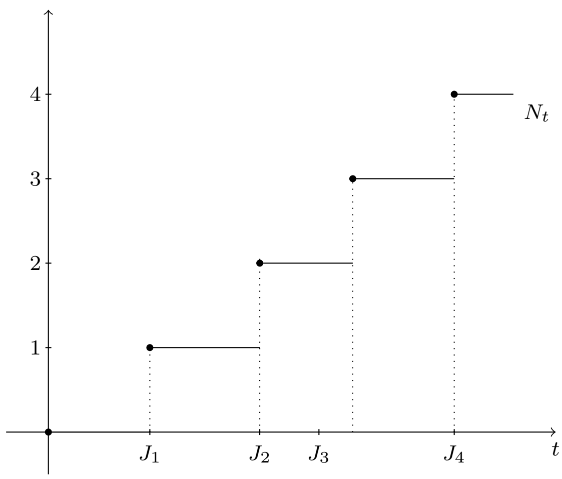


Figure 4.3: Homogeneous Poisson process.

Figure 4.3 shows a realization of a Poisson process. Another useful property of the Poisson process is the following.

c. Conditional on $N_t = n$, the n jump times are uniformly distributed in $(0, t)$, i.e. $J_1, \dots, J_n$ have the same distribution of the order statistics $U_{(1)}, \dots U_{(n)}$ with $U_i \overset{\text{iid}}{\sim} \mathcal{U}(0, t)$.

Property c. suggests an alternative algorithm to generate a Poisson process of rate λ on $[0, T]$.

Algorithm 4.4: Homogeneous Poisson process II.

Generate $N_T \sim \mathrm{Pois}(\lambda T)$
Generate $U_1, \dots, U_{N_T} \overset{\text{iid}}{\sim} \mathcal{U}(0, T)$
Order the sample $U_{(1)} < \dots < U_{(N_T)}$
Set $J_0 = 0,\ J_n = U_{(n)}$, and $N_t = n,\ t \in [J_n, J_{n+1}),\ n = 1, \dots, N_T$

Finally we mention that a Poisson process $\{N_t,\ t \geq 0\}$ can also be thought of a random counting measure. For a given interval $A = (t_1, t_2)$, $\mu(A) = \sum_{k=1}^{\infty} \mathbb{1}_{\{J_k \in A\}}$ counts the number of jumps that occured in A (which is clearly a random number). Thus $d\mu(t) = \sum_{k=1}^{\infty} \delta_{J_k}(t)$ and it holds $N_t = N_0 + \int_0^t d\mu(t)$.

4.5 Non-homogeneous Poisson process

A non-homogeneous Poisson process with rate $\lambda(t)$ varying over time can be defined by extending the property b. of a Poisson process.

Definition 4.5. $\{N_t,\ t \geq 0, N_0 = 0\}$ *is a non-homogeneous Poisson process with rate* $\lambda : [0, \infty) \to \mathbb{R}_+$ *if it is a right-continuous process with independent increments, such that*

$$\begin{aligned} &\mathbb{P}\left(N_{t+h} - N_t = 0\right) = 1 - \lambda(t)h + o(h), \\ &\mathbb{P}\left(N_{t+h} - N_t = 1\right) = \lambda(t)h + o(h). \end{aligned}$$

Therefore, the non-homogeneous rate $\lambda(t)$ can be characterized by the following limits

$$\lambda(t) = \lim_{h \to 0^+} \frac{1 - \mathbb{P}\left(N_{t+h} - N_t = 0\right)}{h} = \lim_{h \to 0^+} \frac{\mathbb{P}\left(N_{t+h} - N_t = 1\right)}{h}.$$

To be able to generate a non-homogeneous Poisson process we need to derive the distribution of the holding times. This is shown in the next lemma.

Lemma 4.1. *Let* $\{N_t,\ t \geq 0, N_0 = 0\}$ *is a non-homogeneous Poisson process with rate* $\lambda : [0, \infty) \to \mathbb{R}_+$ *and denote by* F *the cdf of the* $n+1$ *holding time* $F_{n+1}(t) = \mathbb{P}\left(S_{n+1} \leq t\right)$. *It holds*

$$F_{n+1}(t) = 1 - \exp\left\{ - \int_{J_n}^{J_n + t} \lambda(s)\, ds \right\}.$$

Proof. We have

$$\begin{aligned} F'_{n+1}(t) &= \lim_{h \to 0} \frac{F_{n+1}(t+h) - F_{n+1}(t)}{h} \\ &= \lim_{h \to 0} \frac{\mathbb{P}\left(t < S_{n+1} \leq t + h\right)}{h} = \lim_{h \to 0} \frac{\mathbb{P}\left(S_{n+1} \leq t + h \mid S_{n+1} > t\right)}{h}(1 - F_{n+1}(t)) \\ &= \lim_{h \to 0} \frac{\mathbb{P}\left(N_{J_n+t+h} > n \mid N_{J_n+t} = n\right)}{h}(1 - F_{n+1}(t)) \\ &= \lim_{h \to 0} \frac{1 - \mathbb{P}\left(N_{J_n+t+h} = n \mid N_{J_n+t} = n\right)}{h}(1 - F_{n+1}(t)) \\ &= \lambda(J_n + t)(1 - F_{n+1}(t)). \end{aligned}$$

Solving this differential equation with initial condition $F_{n+1}(0) = 0$ leads to the desired result. □

Hence, a non-homogeneous Poisson process with rate function $\lambda(t)$ can be generated by the following Algorithm.

Algorithm 4.5: Non-homogeneous Poisson process.

Set $N_0 = 0$, $J_0 = 0$, $Y_0 = 0$
For $n = 1, 2, \ldots$
    Generate $S_n \sim F_n(t) = 1 - \exp\left\{-\int_{J_{n-1}}^{J_{n-1}+t} \lambda(s)\, ds\right\}$
    Set $J_n = J_{n-1} + S_n$,
    Set $N_t = N_{J_{n-1}}$, $t \in [J_{n-1}, J_n)$,
    Set $N_{J_n} = N_{J_{n-1}} + 1$

If we define the function $\Lambda(t) = \int_0^t \lambda(s)\, ds$, and let $\tilde{N}_t$ be a homogeneous Poisson process with rate 1, it can also be shown (exercise) that the non homogeneous Poisson process N_t with rate function $\lambda(t)$ can be obtained as $N_t = \tilde{N}_t \circ \Lambda = \tilde{N}_{\Lambda(t)}$.

4.6 Compound Poisson process

A compound Poisson process $\{X_t,\ t \geq 0, X_0 = 0\}$ is a Poisson process with variable jump intensity. Let $\nu(dy)$ be a probability measure on $\mathbb{R}$ and $\{N_t,\ t \geq 0\}$ a homogeneous Poisson process with rate $\lambda > 0$. Then, the compound Poisson process with jump measure $\lambda\nu(dy)dt$ is given by

$$X_t = \sum_{i=1}^{N_t} Z_i, \quad Z_i \overset{\text{iid}}{\sim} \nu.$$

Algorithm 4.6: Compound Poisson process.

Set $N_0 = 0$, $J_0 = 0$, $Y_0 = 0$
For $n = 1, 2, \ldots$
    Generate $S_n \sim \mathrm{Exp}(\lambda)$ and set $J_n = J_{n-1} + S_n$,
    Generate $Z_n \sim \nu$,
    Set $X_t = X_{J_{n-1}}$, $t \in [J_{n-1}, J_n)$ and $X_{J_n} = X_{J_{n-1}} + Z_n$

4.7 General continuous time / discrete space Markov process

Let $\mathcal{X} = \{y_0, y_1, \ldots\}$ be a discrete (finite or countable) set and let $\mu = \{\mu_i\}_i$ be a probability mass function on $\mathcal{X}$, i.e. $\mu_i \geq 0$, $\forall i$ and $\sum_i \mu_i = 1$. A continuous time Markov chain $\{X_t \in \mathcal{X},\ t \geq 0\}$ with initial state $X_0 \sim \mu$, is fully characterized by the transition probabilities

$$q_{ij}(t) = \lim_{h \to 0^+} \frac{\mathbb{P}\left(X_{t+h} = j \mid X_t = i\right)}{h}$$
$$q_i(t) = \lim_{h \to 0^+} \frac{1 - \mathbb{P}\left(X_{t+h} = i \mid X_t = i\right)}{h}$$

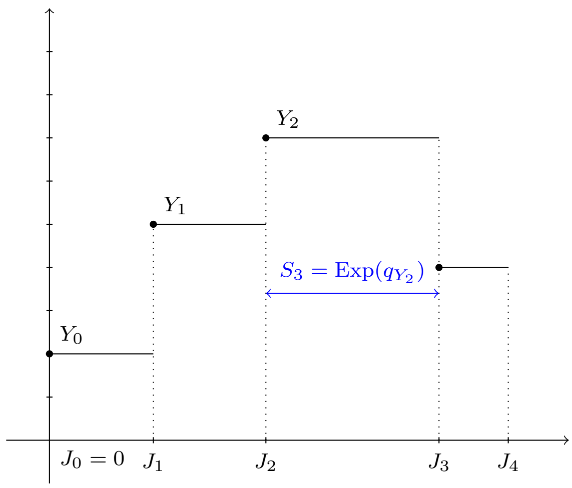


Figure 4.4: Homogeneous continuous time Markov process.

If $q_i(t)$ and $q_{ij}(t)$ do not depend on t, the Markov chain is homogeneous. The (possibly infinite) matrix $Q = (Q_{ij})_{ij}$ given by

$$Q_{ij} = \begin{cases} q_{ij} & i \neq j \\ -q_i & i = j \end{cases}$$

is called the generator of the Markov chain. We assume here that Q is *stable*, i.e. $q_i < \infty$ for all i and *conservative*, i.e. $\sum_{j \neq i} q_{ij} = q_i$.

Definition 4.6. *A homogeneous continuous time Markov chain $\{X_t \in \mathcal{X},\ t \geq 0\}$ with initial state $X_0 \sim \mu$ and (stable and conservative) generator matrix Q, is a right-continuous, piecewise constant process denoted* Markov (μ, Q) *s.t.*

- *the jump process $\{Y_n = X_{J_n},\ n \in \mathbb{N}_0\}$ is a discrete time Markov chain with transition probability*

$$\begin{aligned} &\pi_{ij} = \frac{q_{ij}}{q_i}, \quad i \neq j, \quad \pi_{ii} = 0, && \textit{if } q_i \neq 0 \\ &\pi_{ij} = 0, \quad i \neq j, \quad \pi_{ii} = 1, && \textit{if } q_i = 0. \end{aligned}$$

- *conditional on $Y_0, Y_1, \ldots, Y_{n-1}$, the holding times $S_1, \ldots, S_n$ are independent random variables, $S_i \sim \mathrm{Exp}(q_{Y_{i-1}}),\quad i = 1, \ldots, n$.*

Notice that in this case, the holding time S_n depends on the current state of the chain $S_n \sim \mathrm{Exp}(q_{X_{J_{n-1}}})$ and the chain can jump to any other state j with transition probability $\pi_{X_{J_{n-1}},j}$. An algorithm to generate such a process Markov (μ, Q) is given next.

Algorithm 4.7: Markov (μ, Q).

Generate $X_0 \sim \mu$ and set $J_0 = 0$, $Y_0 = X_0$
For $n = 1, 2, \dots$
  Generate $S_n \sim \text{Exp}(-Q_{Y_n Y_n})$ and set $J_n = J_{n-1} + S_n$,
  Generate $Y_{n+1} \sim \pi_{Y_n, \cdot}$
  Set $X_t = Y_n$, $t \in [J_{n-1}, J_n)$, and $X_{J_{n+1}} = Y_{n+1}$

Exercise 4.3. *The Poisson process* $\{N_t,\ t \geq 0, N_0 = 0\}$ *with rate* $\lambda > 0$ *is a continuous time Markov chain* Markov (δ_0, Q) *with Q-matrix*

$$Q = \begin{bmatrix} -\lambda & \lambda & 0 & \dots \\ 0 & -\lambda & \lambda & \dots \\ 0 & \ddots & \ddots & \ddots \end{bmatrix}$$

since

$$\begin{aligned} Q_{ii} = -\lambda &= \lim_{h \to 0^+} \frac{\mathbb{P}\left(N_{t+h} = i \mid N_t = i\right) - 1}{h} \\ Q_{i,i+1} = \lambda &= \lim_{h \to 0^+} \frac{\mathbb{P}\left(N_{t+h} = i + 1 \mid N_t = i\right)}{h} \\ Q_{i,j} = 0, &\quad j \neq i, i+1. \end{aligned}$$

Exercise 4.4 (Birth process)**.** *Let* $X_t \in \mathbb{N}$ *be the size of a population at time* t*. Births of new individuals arrive after exponential time with rate* λX_t *proportional to the actual size of the population. Hence, the birth process is characterised by the Q-matrix*

$$Q = \begin{bmatrix} -\lambda & \lambda & & & \\ & -2\lambda & 2\lambda & & \\ & & -3\lambda & 3\lambda & \\ & & & \ddots & \ddots \end{bmatrix}.$$

Chapter 5

Monte Carlo method

Let us consider a random variable Z that is the output quantity of a stochastic model and the goal of computing its expectation: $\mu = \mathbb{E}[Z]$. We assume that the probability distribution of Z is not known analytically, but Z can be simulated.

Example 5.1. *Consider a continuous time stochastic process $\{X_t, t \geq 0\}$ with values in a subset $\mathcal{X} \subset \mathbb{R}^d$ and the goal of computing the expectation of X_T at a given time $T \geq 0$, i.e. $Z = X_T$, or the expectation of a stopping time $Z = \inf\{t \geq 0 : X_t \in A \subset \mathcal{X}\}$ for some measurable set A.*

The Monte Carlo method consists simply in generating N i.i.d. replicas $Z^{(1)}, \ldots, Z^{(N)}$ of Z and estimating $\mu = \mathbb{E}[Z]$ by a *sample mean estimator*

$$\hat{\mu}_N = \frac{1}{N} \sum_{i=1}^{N} Z^{(i)}.$$

We assume here that $\mathbb{V}\mathrm{ar}(Z) = \sigma^2 < +\infty$.

5.1 Properties of the Monte Carlo estimator and confidence intervals

The sample mean estimator $\hat{\mu}_N$, which we will call also the *Monte Carlo estimator*, has the following properties.

1. $\hat{\mu}_N$ is *unbiased*, i.e. $\mathbb{E}[\hat{\mu}_N] = \mu$.

   The expectation here is taken with respect to the distribution of the sample $(Z^{(1)}, \ldots, Z^{(N)})$.

2. $\mathbb{V}\mathrm{ar}(\hat{\mu}_N) = \frac{\sigma^2}{N}$.

Indeed,

$$\begin{aligned}
\mathbb{V}\mathrm{ar}\left(\hat{\mu}_N\right) &= \mathbb{E}\left[(\hat{\mu}_N - \mathbb{E}\left[\hat{\mu}_N\right])^2\right] = \mathbb{E}\left[\left(\frac{1}{N}\sum_{i=1}^{N}(Z^{(i)} - \mu)\right)^2\right] \\
&= \frac{1}{N^2}\sum_{i,j=1}^{N}\mathbb{E}\left[(Z^{(i)} - \mu)(Z^{(j)} - \mu)\right] \\
&= \frac{1}{N^2}\sum_{i=1}^{N}\underbrace{\mathbb{E}\left[(Z^{(i)} - \mu)^2\right]}_{=\sigma^2\ \forall i \text{ since } Z^{(i)} \text{ are iid}} + \frac{1}{N^2}\sum_{i\neq j}\underbrace{\mathbb{E}\left[(Z^{(i)} - \mu)(Z^{(j)} - \mu)\right]}_{=0 \text{ since } Z^{(i)}, Z^{(j)} \text{ are indept.}} = \frac{\sigma^2}{N}.
\end{aligned}$$

3. *Almost sure convergence*: $\hat{\mu}_N \xrightarrow{N\to\infty} \mu$ a.s.

   This comes from the Strong Law of Large Numbers (SLLN), since $\mathbb{E}\left[Z\right] = \mu < \infty$.

4. *Asymptotic normality*

$$\frac{\sqrt{N}(\hat{\mu}_N - \mu)}{\sigma} \xrightarrow{\mathrm{d}} N(0, 1)$$

   where $\xrightarrow{\mathrm{d}}$ means convergence in distribution. This comes from the Central Limit Theorem (CLT), since $\mathbb{V}\mathrm{ar}\left(Z\right) < +\infty$.

Using the CLT, we can construct asymptotic $1 - \alpha$ confidence interval (i.e. an interval with coverage probability $1 - \alpha$)

$$I_{\alpha,N} = \left[\hat{\mu}_N - c_{1-\alpha/2}\frac{\sigma}{\sqrt{N}},\ \hat{\mu}_N + c_{1-\alpha/2}\frac{\sigma}{\sqrt{N}}\right]$$

with c_α the α-quantile of the normal distribution satisfying $\Phi(c_\alpha) = \alpha$ and Φ the cdf of a standard normal random variable. This means that $\mathbb{P}\left(\mu_N \in I_{\alpha,N}\right) \xrightarrow{N\to\infty} 1 - \alpha$. See Figure (5.1) for an illustration. Equivalently, the error $|\mu - \hat{\mu}_N|$ satisfies

$$|\mu - \hat{\mu}_N| \leq c_{1-\alpha/2}\frac{\sigma}{\sqrt{N}} \qquad \text{with probability } 1 - \alpha\text{, asymptotically.}$$

The CLT shows that the Monte Carlo error is of order $N^{-1/2}$, which is a very slow convergence rate (to reduce the error by a factor of 10, one has to multiply N by a factor of 100) and is peculiar of Monte Carlo estimates, generally not improvable. On the other hand, it holds under quite weak assumptions ($\mathbb{V}\mathrm{ar}\left(Z\right) < +\infty$).

The previous error estimate and confidence interval is not practical as it involves the, usually unknown, variance $\sigma^2 = \mathbb{V}\mathrm{ar}\left(Z\right)$. We can replace it by the *sample variance estimator* computed using the same sample $(Z^{(1)}, \ldots, Z^{(N)})$,

$$\hat{\sigma}_N^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(Z^{(i)} - \hat{\mu}_N\right)^2.$$

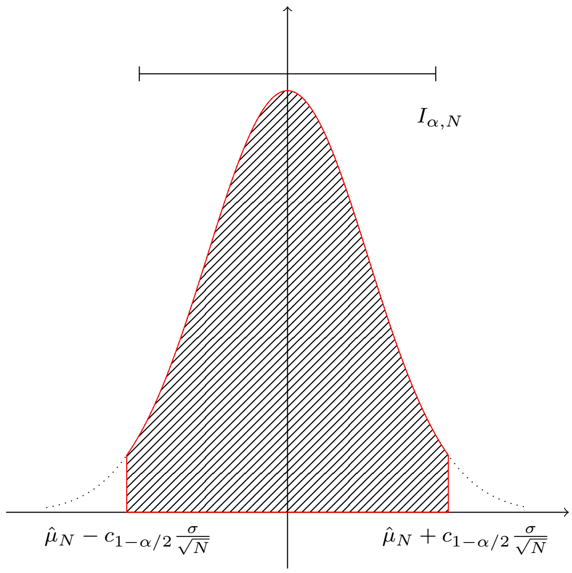


Figure 5.1: Asymptotic confidence interval for the sample mean estimator.

which is also an unbiased estimator and converges almost surely to σ^2. It follows that $\frac{\sigma}{\hat{\sigma}_N} \to 1$ a.s. and

$$\frac{\sqrt{N}(\hat{\mu}_N - \mu)}{\hat{\sigma}_N} = \underbrace{\frac{\sigma}{\hat{\sigma}_N}}_{\to 1 \text{ a.s.}} \underbrace{\frac{\sqrt{N}(\hat{\mu}_N - \mu)}{\sigma}}_{\xrightarrow{\mathrm{d}} N(0,1)} \xrightarrow{\mathrm{d}} N(0,1).$$

Then, a *computable* asymptotic confidence interval is given by

$$\hat{I}_{\alpha,N} = \left[\hat{\mu}_N - c_{1-\alpha/2}\frac{\hat{\sigma}_N}{\sqrt{N}},\ \hat{\mu}_N + c_{1-\alpha/2}\frac{\hat{\sigma}_N}{\sqrt{N}}\right] \tag{5.1}$$

which leads to $\mathbb{P}\left(\mu \in \hat{I}_{\alpha,N}\right) \xrightarrow{N\to\infty} 1 - \alpha$.

5.2 Implementation aspects

As an output of a Monte Carlo simulation, one should *always* provide, beside the point estimate $\hat{\mu}_N$, also an estimate of the error, quantified by e.g. the $1-\alpha$ asymptotic confidence interval $\hat{I}_{\alpha,N}$.

In practice, one would also like to choose N so as to achieve a prescribed tolerance *tol*. Again, this could be for instance in term of the length of the $1-\alpha$ confidence interval:

$$\textit{choose } N\textit{:} \qquad |\hat{I}_{\alpha,N}| \le 2\,tol.$$

This can be done by a two (or more) stages procedure as illustrated by the Algorithm 5.1.

Algorithm 5.1: Two stages Monte Carlo.

1 Do a pilot run with $\bar{N}$ replicas $(Z^{(i)}, \dots, Z^{(\bar{N})})$ and compute

$$\hat{\mu}_{\bar{N}} = \frac{1}{N}\sum_{i=1}^{\bar{N}} Z^{(i)}, \qquad \hat{\sigma}^2_{\bar{N}} = \frac{1}{\bar{N}-1}\sum_{i=1}^{\bar{N}}(Z^{(i)} - \hat{\mu}_{\bar{N}})^2$$

2 Based on the previously estimated variance, fix

$$N = \frac{c^2_{1-\alpha/2}\hat{\sigma}^2_{\bar{N}}}{tol^2}.$$

3 Generate a new sample $(Z^{(1)}, \dots, Z^{(N)})$ and compute $\hat{\mu}_N$ and $\hat{\sigma}^2_N$

4 **if** $\hat{\sigma}^2_N > \hat{\sigma}^2_{\bar{N}}$ **then**

5   Set $\bar{N} = N$ and go back to 2.

6 **else**

7   Output $\hat{\mu}_N$ and $\hat{I}_{\alpha,N}$.

8 **end**

Alternatively, one can adopt a *sequential* procedure as illustrated by Algorithm 5.2.

Algorithm 5.2: Sequential Monte Carlo.

1 Do a pilot run with $\bar{N}$ replicas $(Z^{(i)}, \dots, Z^{(\bar{N})})$ and compute $\hat{\sigma}^2_{\bar{N}} = \dfrac{1}{\bar{N}-1}\sum_{i=1}^{\bar{N}}(Z^{(i)} - \hat{\mu}_{\bar{N}})^2$.

2 Set $N = \bar{N}$, $\hat{\mu}_N = \hat{\mu}_{\bar{N}}$, $\hat{\sigma}_N = \hat{\sigma}_{\bar{N}}$.

3 **while** $\dfrac{\hat{\sigma}_N c_{1-\alpha/2}}{\sqrt{N}} > tol$ **do**

4   set $N = N + 1$

5   generate $Z^{(N)}$ independent of $Z^{(i)}$, $i < N$

6   recompute $\hat{\mu}_N$, $\hat{\sigma}^2_N$

7 **end**

An efficient implementation of Algorithm 5.2 requires stable update formulas for $\hat{\mu}_N$ and $\hat{\sigma}_N$. Two such formulas are the following:

$$\hat{\mu}_{N+1} = \frac{N}{N+1}\hat{\mu}_N + \frac{1}{N+1}Z^{(N+1)}$$

$$\hat{\sigma}^2_{N+1} = \frac{N-1}{N}\hat{\sigma}^2_N + \frac{1}{N+1}\left(Z^{(N+1)} - \hat{\mu}_N\right)^2.$$

If N denotes the sample size at the end of the while loop, which is a random variable, and $\hat{\mu}_N$ the corresponding sample mean estimator, it can be shown [1] under the sole assumption that $\mathbb{V}\mathrm{ar}\,(Z) < +\infty$, that

$$\lim_{tol\to 0} \mathbb{P}\left(|\hat{\mu}_N - \mu| \le tol\right) = 1 - \alpha \qquad \text{(asymptotic consistency)}$$

and

$$\frac{Ntol^2}{\sigma^2 c_{1-\alpha/2}^2} \xrightarrow{\text{a.s.}} 1 \quad \text{as } tol \to 0.$$

The drawback of this algorithm is that if $\bar{N}$ is chosen too small so that the estimator $\hat{\sigma}_N$ is unreliable, this may cause the algorithm to terminate too early, leading to a poor estimation of $\mathbb{E}[Z]$.

5.3 Non asymptotic error bounds

The confidence interval (5.1) for the sample mean estimator, derived in Section 5.1 is based on the CLT and is only valid asymptotically. Sometimes, if the distribution of the random variable Z is far from being Gaussian and the sample size is small, the distribution of the rescaled sample mean estimator $\frac{\sqrt{N}(\hat{\mu}_N-\mu)}{\hat{\sigma}_N}$ may still be far from the asymptotic Normal one and the corresponding confidence interval $\hat{I}_{\alpha,N}$ will be unreliable. Other more robust confidence intervals could be used instead, in this case, which however often lead to very conservative bounds. We mention:

- Bound based on Chebyshev inequality $\mathbb{P}(|Z - \mathbb{E}[Z]| > a) \leq \frac{\mathbb{V}\mathrm{ar}(Z)}{a^2}$ which implies

$$\mathbb{P}\left(|\hat{\mu}_N - \mu| \leq \frac{\sigma}{\sqrt{N\alpha}}\right) \geq 1 - \alpha$$

  This should be compared with the CLT result $\mathbb{P}\left(|\hat{\mu}_N - \mu| \leq \frac{\sigma c_{1-\alpha/2}}{\sqrt{N}}\right) \geq 1 - \alpha$. For α small, one has typically $c_{1-\alpha/2} \ll \frac{1}{\sqrt{\alpha}}$. E.g. for $\alpha = 0.05$ we have $c_{.975} = 1.96$ whereas $\frac{1}{\sqrt{\alpha}} = 4.47$ and for $\alpha = 0.01$ we have $c_{.995} = 2.576$ whereas $\frac{1}{\sqrt{\alpha}} = 10$.

- Bound based on Berry-Essén (for random iid variables with bounded third moment)

$$\sup_x \left| \mathbb{P}\left(\frac{\sqrt{N}(\hat{\mu}_N - \mu)}{\sigma} \leq x\right) - \Phi(x) \right| \leq k \frac{\mathbb{E}\left[|Z-\mu|^3\right]}{\sqrt{N}\sigma^3}, \quad (k \approx 0.4748)$$

  which quantifies how far the distribution of $\frac{\sqrt{N}(\hat{\mu}_N-\mu)}{\sigma}$ is from the standard Normal cdf based on 3rd moment estimates. Given an estimate $\hat{\gamma}_N \approx \mathbb{E}\left[|Z-\mu|^3\right]^{\frac{1}{3}}$ one can then solve the problem:

$$\textit{find } \hat{x}_\alpha : \qquad \Phi(\hat{x}_\alpha) - \frac{k\hat{\gamma}_N^3}{\sqrt{N}\hat{\sigma}_N^3} = 1 - \frac{\alpha}{2}$$

  and output the confidence interval $\hat{I}_{\alpha,N} = [\hat{\mu}_N - \hat{x}_\alpha \frac{\hat{\sigma}_N}{\sqrt{N}},\ \hat{\mu}_N + \hat{x}_\alpha \frac{\hat{\sigma}_N}{\sqrt{N}}]$.

5.4 Vector valued output

The Monte Carlo method estends trivially to a vector valued output $\boldsymbol{Z} = (Z_1, \ldots, Z_m)$ and the estimation of its expected value $\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{Z}]$. In this case, we generate iid replicas $\boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(N)}$ and set $\hat{\boldsymbol{\mu}}_N = \frac{1}{N}\sum_{i=1}^N \boldsymbol{Z}^{(i)}$.

We can also estimate from the same sample the covariance matrix $\hat{C}_N = \frac{1}{N-1}\sum_{i=1}^N (\boldsymbol{Z}^{(i)} - \hat{\boldsymbol{\mu}}_N)(\boldsymbol{Z}^{(i)} - \hat{\boldsymbol{\mu}}_N)^\top$. The considerations on asymptotic confidence intervals based on the CLT extend to the case of a vector valued output as well. Indeed we have $\hat{\boldsymbol{\mu}}_N \xrightarrow{\text{a.s.}} \boldsymbol{\mu}$, $\hat{C}_N \xrightarrow{\text{a.s.}} C$ and

$$N(\hat{\boldsymbol{\mu}}_N - \boldsymbol{\mu})^\top \hat{C}_N^{-1}(\hat{\boldsymbol{\mu}}_N - \boldsymbol{\mu}) \xrightarrow{\text{d}} \chi^2_m$$

where χ^2_m denotes the χ^2 distribution with m degrees of freedom. Based on this asymptotic result, a computable $1-\alpha$ asymptotic confidence region is

$$\hat{I}_{\alpha,N} = \{\boldsymbol{y} \in \mathbb{R}^m : (\hat{\boldsymbol{\mu}}_N - \boldsymbol{y})^\top \hat{C}_N^{-1}(\hat{\boldsymbol{\mu}}_N - \boldsymbol{y}) \leq \frac{\chi^2_{m;1-\alpha}}{N}\}$$

where $\chi^2_{m;1-\alpha}$ is the $1-\alpha$ quantile of the χ^2_m distribution, so that $\mathbb{P}\left(\boldsymbol{\mu} \in \hat{I}_\alpha\right) \xrightarrow{N\to\infty} 1-\alpha$.

5.5 Smooth functions of expectations and delta method

Consider, as in the previous section, a vector of output quantities $\boldsymbol{Z} = (Z_1, \ldots, Z_m)$ of a stochastic model. However, now, we wish to compute a nonlinear function of the expectation of $\boldsymbol{Z}$, i.e.

$$\zeta = f(\mathbb{E}[Z_1], \ldots, \mathbb{E}[Z_m])$$

with $f : \mathbb{R}^m \to \mathbb{R}$ smooth. If we denote $\mu_i = \mathbb{E}[Z_i]$ and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_m) \in \mathbb{R}^m$, the natural Monte Carlo estimator for ζ is

$$\hat{\zeta}_N = f(\hat{\mu}_{1,N}, \ldots, \hat{\mu}_{m,N}) \qquad \text{with} \quad \hat{\mu}_{i,N} = \frac{1}{N}\sum_{k=1}^N Z_i^{(k)}$$

with $\boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(N)}$ iid replicas of $\boldsymbol{Z}$. If f is continuous at $\boldsymbol{\mu}$, then $\hat{\zeta}_N \xrightarrow{\text{a.s.}} \zeta$ i.e. $\hat{\zeta}_N$ is a consistent estimator of ζ.

The question is now how to estimate the error on $\hat{\zeta}_N$ and provide a confidence interval. One way of doing this is provided by the so calles *delta method*, based on a first order Taylor expansion of f around $\boldsymbol{\mu}$:

$$\hat{\zeta}_N - \zeta = f(\hat{\boldsymbol{\mu}}_N) - f(\boldsymbol{\mu}) = \nabla f(\boldsymbol{\mu})(\hat{\boldsymbol{\mu}}_N - \boldsymbol{\mu}) + o(\|\hat{\boldsymbol{\mu}}_N - \boldsymbol{\mu}\|).$$

where we have used the convention that ∇f is a row vector. Let $C = \mathrm{Cov}(\boldsymbol{Z}) = \mathbb{E}\left[(\boldsymbol{Z} - \boldsymbol{\mu})(\boldsymbol{Z} - \boldsymbol{\mu})^\top\right]$. Then

$$\sqrt{N}(\hat{\zeta}_N - \zeta) \xrightarrow{\text{d}} N(0, \nabla f(\boldsymbol{\mu}) C \nabla f(\boldsymbol{\mu})^\top).$$

A computable $1-\alpha$ confidence interval can then be constructed by replacing C with $\hat{C}_N = \frac{1}{N-1}\sum_{i=1}^N (\boldsymbol{Z}^{(i)} - \hat{\mu})(\boldsymbol{Z}^{(i)} - \hat{\mu})^\top$ and $\nabla f(\boldsymbol{\mu})$ with $\nabla f(\hat{\boldsymbol{\mu}}_N)$ as

$$\hat{I}_{\alpha,N} = [\hat{\zeta}_N - \Delta_N,\ \hat{\zeta}_N + \Delta_N], \qquad \Delta_N = \frac{c_{1-\alpha/2}}{\sqrt{N}}\sqrt{\nabla f(\hat{\boldsymbol{\mu}}_N)\hat{C}_N \nabla f(\hat{\boldsymbol{\mu}}_N)^\top}.$$

Obverse that the estimator $\hat{\zeta}_N$ is *biased* in general.

Example 5.2. *Let Z be a scalar random variable, output of a stochastic model. Suppose we want to estimate the coefficient of variation of Z*

$$\zeta = \frac{\sigma(Z)}{\mu(Z)} = \frac{\sqrt{\mathbb{E}[Z^2] - \mathbb{E}[Z]^2}}{\mathbb{E}[Z]}.$$

Setting $\boldsymbol{Z} = (Z_1, Z_2) = (Z, Z^2)$ and $f(x,y) = \sqrt{\frac{y}{x^2} - 1}$, then $\zeta = f(\mathbb{E}[Z_1], \mathbb{E}[Z_2])$. If $\hat{\zeta}_N$ denotes a Monte Carlo estimator for ζ, the delta method can be used to produce a $1-\alpha$ asymptotic confidence interval. Explicit calculations are left as an exercise.

5.6 Monte Carlo to compute integrals

Consider a simple stochastic model $Z = \psi(X_1, \dots, X_d)$ with $\psi : \mathbb{R}^d \to \mathbb{R}$ bounded and $\boldsymbol{X} = (X_1, \dots X_d)$ a random vector with joint probability density function $f : \mathbb{R}^d \to \mathbb{R}_+$. Then,

$$\mathbb{E}[Z] = \int_{\mathbb{R}^d} \psi(x_1, \dots, x_d) f(x_1, \dots, x_d)\, dx_1 \dots dx_d = \int_{\mathbb{R}^d} \psi(\boldsymbol{x}) f(\boldsymbol{x})\, d\boldsymbol{x}.$$

A Monte Carlo approximation of $\mu = \mathbb{E}[Z]$ consists of:

- generating N iid replicas of $X^{(i)} \sim f$
- computing $\hat{\mu}_N = \frac{1}{N}\sum_{i=1}^N \psi(\boldsymbol{X}^{(i)}) \approx \int_{\mathbb{R}^d} \psi(\boldsymbol{x}) f(\boldsymbol{x})\, d\boldsymbol{x}$.

Hence, the formula $\frac{1}{N}\sum_{i=1}^N \psi(\boldsymbol{X}^{(i)})$ can be seen as a quadrature formula to approximate the integral $\int_{\mathbb{R}^d} \psi(\boldsymbol{x}) f(\boldsymbol{x})\, d\boldsymbol{x}$.

Conversely let us consider the problem of computing an integral $I = \int_{\mathbb{R}^d} \psi(\boldsymbol{x}) w(\boldsymbol{x})\, d\boldsymbol{x}$ where $w : \mathbb{R}^d \to \mathbb{R}_+$ a non negative weight such that $\int_{\mathbb{R}^d} w = 1$. Then we can estimate the integral by a Monte Carlo formula

$$\hat{I} = \frac{1}{N}\sum_{i=1}^N \psi(\boldsymbol{X}^{(i)})$$

with $\boldsymbol{X}^{(i)} \overset{\text{iid}}{\sim} w$.

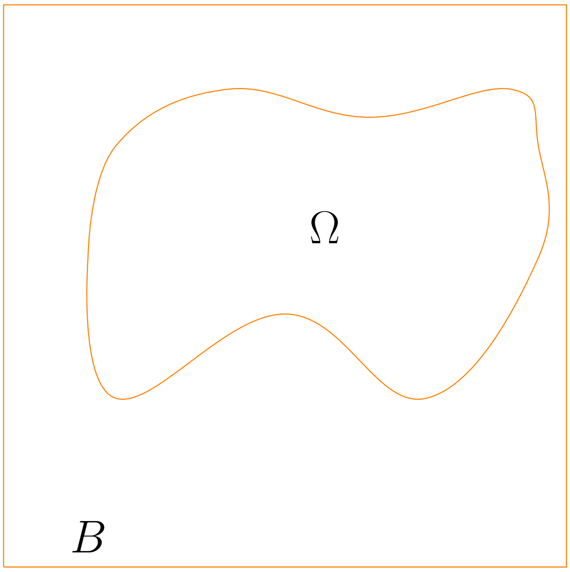


Figure 5.2: Monte Carlo to estimate the volume of Ω.

Example 5.3. *Let $\Omega \subset \mathbb{R}^d$ be a bounded domain. We want to compute its volume $|\Omega|$. Let B be a rectangular domain containing Ω. Then $I = |\Omega| = |B| \int_B \mathbb{1}_\Omega(\boldsymbol{x}) \frac{1}{|B|}\, d\boldsymbol{x}$ and its Monte Carlo approximation is*

$$\hat{I} = \frac{|B|}{N} \sum_{i=1}^{N} \mathbb{1}_\Omega(\boldsymbol{X}^{(i)}) = \frac{\#\{\boldsymbol{X}^{(i)} \in \Omega\}}{N} |B|,$$

with $\boldsymbol{X}^{(i)} \overset{iid}{\sim} \mathcal{U}(B)$ i.e. we draw independently uniform points in B and count how many fall in Ω. See Figure 5.2 for a graphical illustration.

The error in the Monte Carlo approximation is

$$|I - \hat{I}| \leq c_{1-\alpha/2} \frac{\sigma}{\sqrt{N}}$$

with probability $1 - \alpha$ asymptotically, where

$$\sigma^2 = \int_{\mathbb{R}^d} \psi^2(\boldsymbol{x}) w(\boldsymbol{x})\, d\boldsymbol{x} - I^2 \leq \int_{\mathbb{R}^d} \psi^2(\boldsymbol{x}) w(\boldsymbol{x})\, d\boldsymbol{x}.$$

Hence, the rate of convergence is $O(N^{-1/2})$ and is achieved under the sole condition $\int_{\mathbb{R}^d} \psi^2(\boldsymbol{x}) w(\boldsymbol{x})\, d\boldsymbol{x} < +\infty$. Observe, in particular, that this rate is *independent of the dimension d*! (assuming that the variance σ^2 remains bounded when we increase the dimension of the problem). Although Monte Carlo has a very poor convergence rate $O(N^{-1/2})$, its use is still very appealing for high dimensional problems.

As a term of comparison, consider the problem of computing an integral $I = \int_{[0,1]^d} \psi(\boldsymbol{x})\, d\boldsymbol{x}$ on the unit hypercube by a tensor quadrature formula, e.g. tensor mid-point rule

$$I^{mp} = \sum_{i=1}^{N} \psi(\boldsymbol{X}^{(i)}) h^d = \frac{1}{N} \sum_{i=1}^{N} \psi(\boldsymbol{X}^{(i)})$$

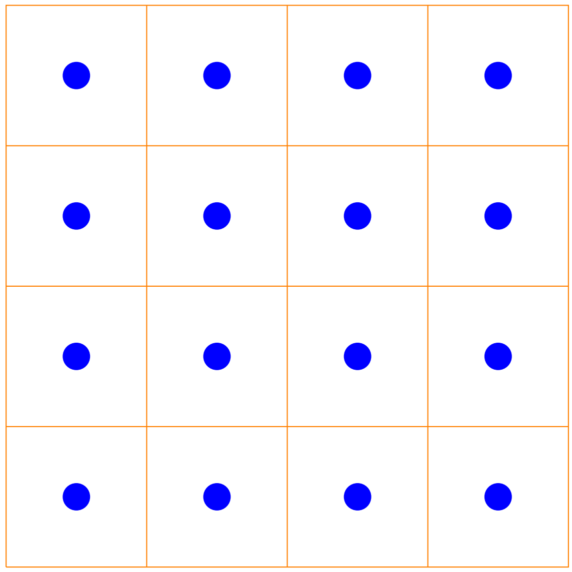


Figure 5.3: Uniform grid to estimate the integral of ψ.

where $\boldsymbol{X}^{(i)}$ are the centres of the cells and $h = N^{-1/d}$ is the length of each cell.

The error of the quadrature formula can be bounded as:

$$|I - I^{mp}| \leq Ch^2 \|\psi\|_{C^2([0,1]^d)} = CN^{-2/d} \|\psi\|_{C^2([0,1]^d)}.$$

Therefore, such a formula achieves a rate $N^{-2/d}$, with respect to the number of points used, provided $\psi \in C^2([0,1]^d)$ (hence regularity is required on the integrand to achieve such rate) and alrady for $d > 4$ the rate will be worse than Monte Carlo (this effect is usually referred to as the “curse of dimensionality”).

Chapter 6

Variance Reduction Techniques

As in the previous chapter, let Z be a random variable, output of a stochastic model, and consider the goal of computing the expected value $\mu = \mathbb{E}[Z]$. It will be useful to assume that Z can be written as $Z = \psi(X)$ with $X = (X_1, \ldots, X_d)$ a random vector with pdf $f : \mathbb{R}^d \to \mathbb{R}_+$, so that

$$\mu = \mathbb{E}[Z] = \int_{\mathbb{R}^d} \psi(x) f(x)\, dx.$$

The Monte Carlo approach (hereafter called "Crude Monte Carlo") to approximate μ consists in generating N iid replicas $Z^{(i)}, \ldots, Z^{(N)}$, with $Z^{(i)} = \psi(X^{(i)})$, $X^{(i)} \overset{\text{iid}}{\sim} f$ and computing

$$\hat{\mu}_{\text{CMC}} = \frac{1}{N} \sum_{i=1}^{N} Z^{(i)}.$$

As we have seen in Chapter 5, by the CLT we have that

$$|\mu - \hat{\mu}_{\text{CMC}}| \leq c_{1-\alpha/2} \frac{\sqrt{\mathbb{V}\text{ar}(Z)}}{\sqrt{N}}$$

with probability $1 - \alpha$, asymptotically as $N \to \infty$.

The techniques of variance reduction aim at improving the performance of a Crude Monte Carlo approximation by reducing the constant $\sqrt{\mathbb{V}\text{ar}(Z)}$, hence the name "variance reduction". The idea is simple: instead of applying the sample mean estimator $\hat{\mu} = \hat{\mu}(Z)$ to the variable Z, one applies it to a cleverly modified version $\tilde{Z}$ which satisfies

$$\mathbb{E}[\tilde{Z}] = \mathbb{E}[Z] = \mu \quad \text{and} \quad \mathbb{V}\text{ar}(\tilde{Z}) \ll \mathbb{V}\text{ar}(Z).$$

Hence, a Monte Carlo approximation with variance reduction will look like

$$\hat{\mu}_{\text{VR}} = \frac{1}{N} \sum_{i=1}^{N} \tilde{Z}^{(i)}$$

with $\tilde{Z}^{(i)} \overset{\text{iid}}{\sim} \tilde{Z}$.

6.1 Antithetic Variables

Suppose N even. Instead of generating N iid replicas of Z, the underlying idea of antithetic sampling is to generate $N/2$ iid pairs of negatively correlated random variables

$$(Z^{(1)}, Z_a^{(1)}),\ (Z^{(2)}, Z_a^{(2)}), \dots, (Z^{(N/2)}, Z_a^{(N/2)}),$$

where all $Z^{(i)}, Z_a^{(i)}$ have the same distribution as Z but $\mathrm{Cov}(Z^{(i)}, Z_a^{(i)}) < 0$, $i = 0, \dots, N/2$. If we now consider the estimator

$$\hat{\mu}_{\mathrm{AV}} = \frac{1}{N/2} \sum_{i=1}^{N/2} \frac{Z^{(i)} + Z_a^{(i)}}{2}$$

it follows immediately that

$$\mathbb{E}\left[\hat{\mu}_{\mathrm{AV}}\right] = \mathbb{E}\left[Z\right] = \mu$$

and

$$\begin{aligned}
\mathbb{V}\mathrm{ar}\left(\hat{\mu}_{\mathrm{AV}}\right) &= \frac{4}{N^2} \sum_{i=1}^{N/2} \mathbb{V}\mathrm{ar}\left(\frac{Z^{(i)} + Z_a^{(i)}}{2}\right) = \frac{1}{2N} \mathbb{V}\mathrm{ar}\left(Z^{(1)} + Z_a^{(1)}\right) \\
&= \frac{1}{2N}\left(\mathbb{V}\mathrm{ar}\left(Z^{(1)}\right) + \mathbb{V}\mathrm{ar}\left(Z_a^{(1)}\right) + 2\,\mathrm{Cov}(Z^{(1)}, Z_a^{(1)})\right) \\
&= \frac{\mathbb{V}\mathrm{ar}\left(Z\right) + \mathrm{Cov}(Z^{(1)}, Z_a^{(1)})}{N} < \mathbb{V}\mathrm{ar}\left(\hat{\mu}_{\mathrm{CMC}}\right)
\end{aligned}$$

since, by assumption, $\mathrm{Cov}(Z^{(1)}, Z_a^{(1)}) < 0$. The estimator $\hat{\mu}_{AV}$ has therefore a smaller variance than the Curde Monte Carlo estimator $\hat{\mu}_{CMC}$ at the same computational cost (provided the generation of $Z_a^{(i)}$ has the same cost as the generation of $Z^{(i)}$).

The question is now how to generate pairs of negatively correlated variables $(Z^{(i)}, Z_a^{(i)})$. The following proposition presents a situation in which variance reduction can be achieved by a rather simple construction of antithetic sampling.

Proposition 6.1. *Assume that the random variable Z has the expression $Z = \psi(X)$, with $X = (X_1, \dots, X_d)$ a random vector with independent components, such that*

- *X has a symmetric distribution around its mean, i.e. $2\mathbb{E}\left[X\right] - X \sim X$*
- *ψ is a monotone function in each of its arguments.*

Then $Z = \psi(X)$ and $Z_a = \psi(2\mathbb{E}\left[X\right] - X)$ satisfy

$$\mathbb{E}\left[Z\right] = \mathbb{E}\left[Z_a\right] \quad \textit{and} \quad \mathit{Cov}(Z, Z_a) < 0.$$

Under the assumptions of the previous proposition, a Monte Carlo approximation of $\mu = \mathbb{E}\left[Z\right]$ with antithetic variables can be constructed by the following algorithm.

Algorithm 6.1: Antithetic variables.

1 Generate $N/2$ iid replicas $X^{(1)}, \dots, X^{(N/2)}$ of X;
2 For each $X^{(i)}$ compute $Z^{(i)} = \psi(X^{(i)})$ and $Z_a^{(i)} = \psi(2\mathbb{E}[X] - X^{(i)})$;
3 Compute $\hat{\mu}_{\mathrm{AV}} = \frac{1}{N}\sum_{i=1}^{N/2}(Z^{(i)} + Z_a^{(i)})$.
4 Estimate $\hat{\sigma}^2_{\mathrm{AV}} = \frac{1}{N/2-1}\sum_{i=1}^{N/2}\left(\frac{Z^{(i)}+Z_a^{(i)}}{2} - \hat{\mu}_{\mathrm{AV}}\right)^2$
5 Output $\hat{\mu}_{\mathrm{AV}}$ and a (asymptotic) $1-\alpha$ confidence interval

$$\hat{I}_{\alpha,N} = \left[\hat{\mu}_{\mathrm{AV}} - c_{1-\alpha/2}\frac{\hat{\sigma}_{\mathrm{AV}}}{\sqrt{N/2}}, \hat{\mu}_{\mathrm{AV}} + c_{1-\alpha/2}\frac{\hat{\sigma}_{\mathrm{AV}}}{\sqrt{N/2}}\right]$$

The proof of Proposition 6.1 relies on the following Chebyshev Covariance inequality.

Lemma 6.2 (Chebyshev Covariance inequality)**.** *Let X be a real-valued random variable with pdf $f : \mathbb{R} \to \mathbb{R}_+$ and let $g, h : \mathbb{R} \to \mathbb{R}$ be functions that are both non-decreasing or both non-increasing, such that $\mathbb{E}[|g(X)|], \mathbb{E}[|h(X)|], \mathbb{E}[|g(x)h(x)|] < +\infty$. Then $Cov(g(X), h(X)) \geq 0$.*

Proof. We consider the case of g, h both non-decreasing. The other case can be proven analogously. Let $\tilde{g}(x) = g(x) - \mathbb{E}[g(x)]$ and $\tilde{h}(x) = h(x) - \mathbb{E}[h(x)]$. Observe first that

$$\begin{aligned}\frac{1}{2}\iint (g(x)-g(y))(h(x)-h(y))f(x)f(y)\,dx\,dy &= \\ &\frac{1}{2}\iint (\tilde{g}(x)-\tilde{g}(y))(\tilde{h}(x)-\tilde{h}(y))f(x)f(y)\,dx\,dy \\ = \int \tilde{g}(x)\tilde{h}(x)f(x)\,dx - \underbrace{\left(\int \tilde{g}(x)f(x)\,dx\right)}_{=0}\underbrace{\left(\int \tilde{h}(y)f(y)\,dy\right)}_{=0} &= \mathrm{Cov}(g(X), h(X))\end{aligned}$$

Hence

$$\begin{aligned}\mathrm{Cov}(g(X), h(X)) =& \frac{1}{2}\int_{x\geq y} \underbrace{(g(x)-g(y))}_{\geq 0}\underbrace{(h(x)-h(y))}_{\geq 0} f(x)f(y)dxdy \\ &+ \frac{1}{2}\int_{x<y} \underbrace{(g(x)-g(y))}_{\leq 0}\underbrace{(h(x)-h(y))}_{\leq 0} f(x)f(y)dxdy \geq 0.\end{aligned}$$

□

The previous inequality generalizes to the multivariate case, whose proof is left as exercise.

Lemma 6.3. *Let $X = (X_1, \dots, X_d) \in \mathbb{R}^d$ be a random vector with independent components and let $g, h : \mathbb{R}^d \to \mathbb{R}$ be functions whose dependence on each argument is either non-decreasing or non-increasing for both of them. Then $Cov(g(X), h(X)) \geq 0$.*

Proof of Proposition 6.1. Since $2\mathbb{E}[X] - X \sim X$ we have $Z_a \sim Z$, hence $\mathbb{E}[Z_a] = \mathbb{E}[Z]$. Moreover, observe that if $\psi(X_1, \ldots, X_d)$ is e.g. non decreasing in the i-th argument, so is the function $-\psi(2\mathbb{E}[X_1] - X_1, \ldots, 2\mathbb{E}[X_d] - X_d)$ and, from the previous Lemma, we have $\mathrm{Cov}(\psi(X), -\psi(2\mathbb{E}[X] - X)) \geq 0$, hence $\mathrm{Cov}(Z, Z_a) \leq 0$. □

Example 6.1. *Let $Z \sim \mathrm{Exp}(\lambda)$. Then $Z = -\frac{1}{\lambda} \log X = \psi(X)$ with $X \sim \mathcal{U}(0,1)$ and ψ monotone (decreasing). It follows that $\psi(X)$ and $\psi(1-X)$ are negeatively correlated and a Monte Carlo estimator with antithetic variables for the computation of $\mu = \frac{1}{\lambda} = \mathbb{E}[Z]$ is $\hat{\mu}_{AV} = \frac{1}{N}\sum_{i=1}^{N/2}(-\frac{1}{\lambda}\log(X^{(i)}) - \frac{1}{\lambda}\log(1-X^{(i)}))$, with $X^{(i)} \overset{iid}{\sim} \mathcal{U}(0,1)$.*

Example 6.2. *Consider the problem of pricing a European option $\mu = \mathbb{E}[Z]$ with $Z = \psi(S_T) = e^{-rT}(S_T - K)_+$ where S_t is the solution of the stochastic differential equation*

$$dS_t = rS_t\,dt + \sigma S_t\,dW_t$$

with W_t a standard Wiener process and S_0 given. It can be shown that $X_t = \log(S_t/S_0)$ satisfies the stochastic differential equation with constant coefficients

$$dX_t = (r - \sigma^2/2)\,dt + \sigma\,dW_t, \quad X_0 = 0$$

whose solution is $X_t = (r-\sigma^2/2)t + \sigma W_t \sim N((r-\sigma^2/2)t, \sigma^2 t)$. Hence $S_T = S_0 e^{X_T}$ has a log-normal distribution with $X_T \sim N((r-\sigma^2/2)T, \sigma^2 T)$ and $\mathbb{E}[S_T] = S_0 e^{rT}$. Observe that ψ is a non decreasing function of S_T, which, on its turn, is an increasing function of X_T whose distribution is symmetric about its mean. Hence $\tilde{\psi}(X_T) = \psi(S_0 e^{X_T})$ is non decreasing in X_T and an antithetic variable estimator

$$\hat{\mu}_{AV} = \frac{1}{N}\sum_{i=1}^{N/2}\left(\tilde{\psi}(X_T^{(i)}) + \tilde{\psi}((2r-\sigma^2)T - X_T^{(i)})\right), \qquad X_T^{(i)} \overset{iid}{\sim} N\left((r - \frac{\sigma^2}{2})T, \sigma^2 T\right)$$

will lead to variance reduction.

Example 6.3. *Consider a random walk on the integers: $Z_{n+1} = Z_n + X_{n+1}$ with X_i iid such that $\mathbb{P}(X_i = 1) = \mathbb{P}(X_i = -1) = 1/2$ and $Z_0 = 0$. We want to estimate by Monte Carlo $\mu = \mathbb{P}(Z_N \geq s)$ with $s \in \mathbb{N}$. Denote*

$$\psi(Z_N) = \mathbb{1}_{\{Z_N \geq s\}} = \mathbb{1}_{\{\sum_{n=1}^N X_n \geq s\}} = \tilde{\psi}(X_1, \ldots, X_N).$$

Then $\mu = \mathbb{E}[\psi(Z_N)] = \mathbb{E}\left[\tilde{\psi}(X_1, \ldots, X_N)\right]$. Since $\tilde{\psi}$ is a non decreasing function in each X_n, and each X_n has a simmetric distribution around its mean $\mathbb{E}[X_n] = 0$, a MC estimator with antithetic variables will lead to variance reduction. It consists in generating $N/2$ iid paths $Z_{n+1}^{(i)} = Z_n^{(i)} + X_n^{(i)}$ as well as the antithetic paths *$\tilde{Z}_{n+1}^{(i)} = \tilde{Z}_n^{(i)} - X_n^{(i)}$, and build the estimator $\hat{\mu}_{AV} = \frac{1}{N}\sum_{i=1}^{N/2}(\psi(Z_N^{(i)}) + \psi(\tilde{Z}_N^{(i)}))$.*

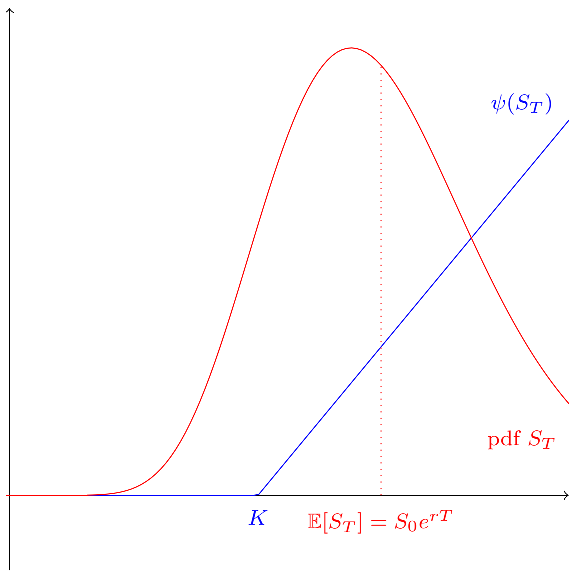


Figure 6.1: European option.

6.2 Importance Sampling

Let $X \in \mathbb{R}^d$ be a random vector with pdf $f : \mathbb{R}^d \to \mathbb{R}_+$ and $Z = \psi(X)$ with $\psi : \mathbb{R}^d \to \mathbb{R}$. Then, computing the expected value of Z corresponds to computing the multidimensional integral

$$\mu = \mathbb{E}[Z] = \int_{\mathbb{R}^d} \psi(x) f(x)\, dx$$

Let now $g : \mathbb{R}^d \to \mathbb{R}_+$ be an auxiliary pdf such that $g(x) = 0$ only if $\psi(x)f(x) = 0$. Then, the integral can be rewritten as

$$\mu = \mathbb{E}[Z] = \int_{\mathbb{R}^d} \left(\frac{\psi(x) f(x)}{g(x)} \right) g(x)\, dx = \mathbb{E}_g \left[\frac{\psi f}{g} \right].$$

where $\mathbb{E}_g$ denotes expectation under the measure $g(x)\, dx$. It follows that in a Monte Carlo approach, instead of generating iid replicas of X to estimate $\mu = \mathbb{E}_f[\psi(X)]$, we could generate iid replicas of $\tilde{X}$ having pdf g, and estimate $\mu = \mathbb{E}_g \left[\frac{\psi(\tilde{X}) f(\tilde{X})}{g(\tilde{X})} \right]$. This technique is known as *importance sampling*. The auxiliary distribution g is called the *importance sampling* or *dominating* distribution and the correcting factor $w(x) = \frac{f(x)}{g(x)}$ is often called the *likelihood ratio*.

In more general terms, if X has measure ν_X and ν^* is another probability measure that dominates ν_X, i.e. ν_X is absolutely continuous with respect to ν^*, then there exists a density $\rho = \frac{d\nu_X}{d\nu^*}$ (Radon-Nicodyn derivative), and $\mathbb{E}[Z]$ can be rewritten as

$$\mu = \mathbb{E}[Z] = \int \psi(x) d\nu_X(x) = \int \psi(x) \rho(x) d\nu^*(x) = \mathbb{E}_*[\psi\rho]$$

Algorithm 6.2: Importance sampling

1. Generate N iid replicas $\tilde{X}^{(1)}, \dots, \tilde{X}^{(N)} \sim g$
2. Compute $\hat{\mu}_{\text{IS}} = \frac{1}{N}\sum_{i=1}^N \frac{\psi(\tilde{X}^{(i)})f(\tilde{X}^{(i)})}{g(\tilde{X}^{(i)})}$
3. Estimate $\hat{\sigma}^2_{\text{IS}} = \frac{1}{N-1}\sum_{i=1}^N \left(\frac{\psi(\tilde{X}^{(i)})f(\tilde{X}^{(i)})}{g(\tilde{X}^{(i)})} - \hat{\mu}_{\text{IS}}\right)^2$
4. Output $\hat{\mu}_{\text{IS}}$ and a (asymptotic) $1-\alpha$ confidence interval

$$I_\alpha = \left[\hat{\mu}_{\text{IS}} - c_{1-\alpha/2}\frac{\hat{\sigma}_{\text{IS}}}{\sqrt{N}}, \hat{\mu}_{\text{IS}} + c_{1-\alpha/2}\frac{\hat{\sigma}_{\text{IS}}}{\sqrt{N}}\right]$$

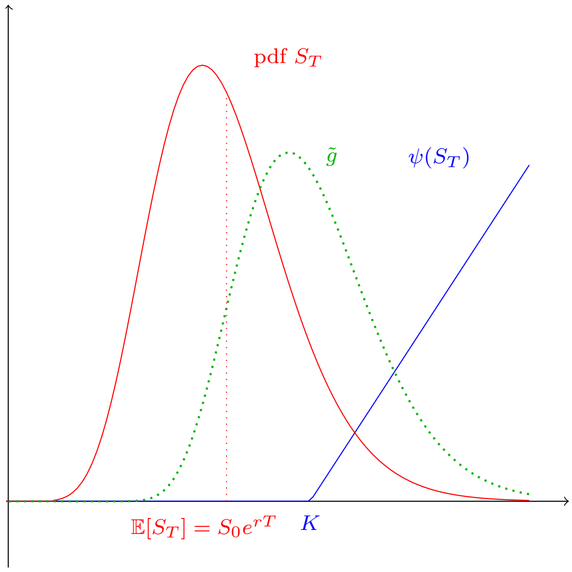


Figure 6.2: European option.

and an importance sampling strategy consists in generating iid replicas $\tilde{X}^{(i)} \overset{\text{iid}}{\sim} \nu^*$, $i = 1, \dots, N$ and estimating the empirical mean $\hat{\mu}_{\text{IS}} = \frac{1}{N}\sum_{i=1}^N \psi(\tilde{X}^{(i)})\rho(\tilde{X}^{(i)})$.

Example 6.4. *Let us consider again the option pricing problem of computing $\mu = \mathbb{E}[Z]$, $Z = \psi(S_T) = e^{-rT}(S_T - K)_+$ with $S_T = S_0\exp(X_T)$ and $X_T \sim N((r-\sigma^2/2)T, \sigma^2 T)$. If $K \gg \mathbb{E}[S_T] = S_0 e^{rT}$, most of the the mass of S_T falls in the region where $\psi(S_T) = 0$. Hence a crude Monte Carlo estimator will be very ineffective as only few replicas of S_T will fall in the "interesting" region $S_T > K$. The idea would then be to "artificially" push the distribution to the right. This can be achieved, for instance, by increasing the drift parameter r in the dynamics of S_t. We can therefore simulate a geometric Brownian motion*

$$d\tilde{S}_t = \tilde{r}\tilde{S}_t\,dt + \sigma\tilde{S}_t\,dW_t$$

with an increased drift rate $\tilde{r} > r$. Let $X_T = \log(S_T/S_0) \sim N((r-\sigma^2/2)T, \sigma^2 T)$ and $\tilde{X}_T = \log(\tilde{S}_T/S_0) \sim N((\tilde{r}-\sigma^2/2)T, \sigma^2 T)$, and denote by f_{X_T} and $f_{\tilde{X}_T}$ the pdfs of X_T

and $\tilde{X}_T$, respectively. It follows that

$$\mu = \int_{\mathbb{R}} \psi(S_0 e^x) f_{X_T}(x)\, dx = \int_{\mathbb{R}} \psi(S_0 e^x) w(x) f_{\tilde{X}_T}(x)\, dx$$

with likelihood ratio

$$w(x) = \frac{f_{X_T}(x)}{f_{\tilde{X}_T}(x)} = \exp\left\{ \frac{(\tilde{r}-r)((\tilde{r}+r-\sigma^2)T - 2x)}{2\sigma^2} \right\} = (e^x)^{-\frac{\tilde{r}-r}{\sigma^2}} e^{\frac{(\tilde{r}-r)(\tilde{r}+r-\sigma^2)T}{2\sigma^2}}$$

and an importance sampling estimator is

$$\hat{\mu}_{IS} = \frac{1}{N} \sum_{i=1}^N \psi(\tilde{S}_T^{(i)}) \left(\frac{S_T^{(i)}}{S_0} \right)^{-\frac{\tilde{r}-r}{\sigma^2}} e^{\frac{(\tilde{r}-r)(\tilde{r}+r-\sigma^2)T}{2\sigma^2}}$$

with

$$\log\left(\tilde{S}_T^{(i)}/S_0\right) \overset{iid}{\sim} N((\tilde{r}-\sigma^2/2)T, \sigma^2 T).$$

6.2.1 On the choice of the importance sampling distribution g

The importance sampling estimator

$$\hat{\mu}_{\mathrm{IS}} = \frac{1}{N} \sum_{i=1}^N \frac{\psi(\tilde{X}^{(i)}) f(\tilde{X}^{(i)})}{g(\tilde{X}^{(i)})}, \qquad \tilde{X}^{(i)} \overset{\mathrm{iid}}{\sim} g$$

is unbiased and has variance

$$\mathbb{V}\mathrm{ar}\left(\hat{\mu}_{\mathrm{IS}}\right) = \frac{1}{N} \mathbb{V}\mathrm{ar}_g\left(\frac{\psi f}{g}\right) = \frac{1}{N}\left(\int_{\mathbb{R}^d} \frac{\psi^2(x) f^2(x)}{g^2(x)} g(x)\, dx - \mu^2 \right) = \frac{1}{N}\left(\mathbb{E}_f\left[\psi^2 \frac{f}{g}\right] - \mu^2 \right).$$

Therefore, the optimal choice of g is the one that minimizes $\mathbb{V}\mathrm{ar}\left(\hat{\mu}_{\mathrm{IS}}\right)$, i.e. it minimizes the term $\int_{\mathbb{R}^d} \psi^2 \frac{f^2}{g}\, dx$, under the conditions $\int_{\mathbb{R}^d} g dx = 1$ and $g \geq 0$. It is clear that the optimal distribution should vanish outside $\Gamma = \mathrm{supp}(\psi^2 f^2)$. Moreover, introducing the Lagrangian function

$$\mathcal{L}(g, \lambda) = \int_\Gamma \frac{\psi^2 f^2}{g}\, dx + \lambda \left(\int_\Gamma g - 1 \right)$$

and taking variations, the (necessary) optimality condition reads

$$\frac{\partial \mathcal{L}}{\partial g}(\delta g) = -\int_{\mathbb{R}^d} \left(\psi^2 \frac{f^2}{g^2} - \lambda \right) \delta g dx = 0, \qquad \forall \delta g$$

which implies $g^2 = \psi^2 \frac{f^2}{\lambda}$. We see that the optimal g is given by

$$g^* = \frac{|\psi| f}{\mathbb{E}_f[|\psi|]}.$$

With such optimal g^*, the variance of the importance sampling estimator is $\mathbb{V}\mathrm{ar}\left(\hat{\mu}^*_{\mathrm{IS}}\right) = \mathbb{E}\left[|\psi|^2\right] - \mathbb{E}\left[\psi\right]^2$. In particular, if $\psi \geq 0$, we have $\mathbb{V}\mathrm{ar}\left(\hat{\mu}_{\mathrm{IS}}\right) = 0\,$! However, working with

g^* is clearly not practical as the normalizing constant $\mathbb{E}_f[|\psi|]$ is, in general, as difficult to compute as the original quantity $\mu = \mathbb{E}[\psi]$, and we need to know it explicitely to compute the likelihood ratio.

Although the optimal distribution g^* can not be used in practice, this optimization argument shows that the dominating density g should resemble as much as possible to $|\psi|f$ while still being easily simulatable and with explicit expression.

Often, this optimization is performed over a parametric family of pdfs $\{f(\cdot,\theta),\ \theta \in \Theta\}$. Assuming that the original pdf also belongs to the family, with parameter θ_0, i.e. $f = f(\cdot,\theta_0)$, we can take as dominating distribution

$$g(\cdot) = f(\cdot,\theta^*), \qquad \text{with } \theta^* = \operatorname*{argmin}_{\theta\in\Theta} \mathbb{E}_\theta\left[\frac{\psi^2 f^2(\cdot,\theta_0)}{f^2(\cdot,\theta)}\right] = \operatorname*{argmin}_{\theta\in\Theta} \mathbb{E}_{\theta_0}\left[\frac{\psi^2 f(\cdot,\theta_0)}{f(\cdot,\theta)}\right].$$

A typical case is when $\{f(\cdot,\theta)\}$ is an exponential family $f(x,\theta) \propto \exp(\theta^\top x - k(\theta))$, for which the likelihood ratio $\frac{f(x,\theta_0)}{f(x,\theta)}$ takes a simple form. The optimization above can be performed numerically replacing the exact expectation with a sample average over a pilot run.

Algorithm 6.3: Importance sampling with variance minimization

1 Generate $\bar{N}$ iid replicas $Y^{(1)},\dots,Y^{(\bar{N})} \sim f(\cdot,\theta_0)$
2 Solve the minimization problem

$$\hat{\theta}^*_{\boldsymbol{Y}} = \operatorname*{argmin}_{\theta\in\Theta} \frac{1}{\bar{N}} \sum_{i=1}^{\bar{N}} \psi^2(Y^{(i)}) \frac{f(Y^{(i)},\theta_0)}{f(Y^{(i)},\theta)}$$

3 Generate N iid replicas $X^{(1)},\dots,X^{(N)} \sim f(\cdot,\hat{\theta}^*_{\boldsymbol{Y}})$
4 Compute $\hat{\mu}_{\text{IS}} = \frac{1}{N}\sum_{i=1}^N \psi(X^{(i)}) \frac{f(X^{(i)},\theta_0)}{f(X^{(i)},\hat{\theta}^*)}$.
5 Compute $\hat{\sigma}^2_{IS}$ and output $\hat{\mu}_{\text{IS}}$ as well as a (asymptotic) $1-\alpha$ confidence interval $\hat{I}_{\alpha,N}$.

The estimator $\hat{\mu}_{\text{IS}}$ of algorithm 6.3 is unbiased. Indeed, if we denote by $X = (X^{(1)},\dots,X^{(N)})$ and $\boldsymbol{Y} = (Y^{(1)},\dots,Y^{(\bar{N})})$, and use the tower rule, we have

$$\mathbb{E}[\hat{\mu}_{\text{IS}}] = \mathbb{E}_{\boldsymbol{Y}}[\mathbb{E}_X[\hat{\mu}_{\text{IS}} \mid \boldsymbol{Y}]] = \mathbb{E}_{\boldsymbol{Y}}\left[\int \psi(x)\frac{f(x,\theta_0)}{f(x,\hat{\theta}^*_{\boldsymbol{Y}})} f(x,\hat{\theta}^*_{\boldsymbol{Y}})\,dx\right] = \mu.$$

6.2.2 Weighted importance sampling

In certain cases, the pdf f and/or the dominating pdf g, are known only up to a normalizing constant. (We assume, however, that we can still generate $X \sim g$ e.g. by Acceptance-Rejection). Let $f = c_g\tilde{f}$ and $g = c_g\tilde{g}$, with $c_f = (\int \tilde{f})^{-1}$ and $c_g = (\int \tilde{g})^{-1}$.

A modified (self-normalized) version of the importance sampling estimator, which does not require the explicit knowledge of the normalizing constants (c_f, c_g) is

$$\hat{\mu}_{\mathrm{IS}}^{W} = \frac{\sum_{i=1}^{N} \psi(X^{(i)}) w(X^{(i)})}{\sum_{i=1}^{N} w(X^{(i)})}$$

with $w(x) = \frac{\tilde{f}(x)}{\tilde{g}(x)}$ and $X^{(i)} \overset{\text{iid}}{\sim} g$. Calling $\tilde{w}_i = \frac{w(X^{(i)})}{\sum_{i=1}^{N} w(X^{(i)})}$, the estimator $\hat{\mu}_{\mathrm{IS}}^{W}$ can be written as a weighted average

$$\hat{\mu}_{\mathrm{IS}}^{W} = \sum_{i=1}^{N} \psi(X^{(i)}) \tilde{w}_i.$$

To see that $\hat{\mu}_{\mathrm{IS}}^{W}$ is a consistent estimator, observe that

$$\frac{1}{N} \sum_{i=1}^{N} w(X^{(i)}) \xrightarrow{\text{a.s.}} \int \frac{\tilde{f}(x)}{\tilde{g}(x)} g(x)\, dx = \frac{c_g}{c_f}$$

by the strong law of large numbers (SLLN) and

$$\frac{1}{N} \sum_{i=1}^{N} \psi(X^{(i)}) w(X^{(i)}) \xrightarrow{\text{a.s.}} \int \psi \frac{\tilde{f}}{\tilde{g}} g\, dx = \frac{c_g}{c_f} \mu$$

again by SLLN. This estimator is biased, although the bias is usually small. Observe that this weighted version of the importance sampling estimator requires the stronger condition $f(x) = 0$ if $g(x) = 0$ (as opposed to the condition $\psi(x) f(x) = 0$ if $g(x) = 0$ of the standard estimator).

6.2.3 Importance sampling for stochastic processes

Discrete time Markov Chains. Consider a discrete time Markov chain in $\mathbb{R}^d$, $\{X_n,\ n \in \mathbb{N}_0\} \sim \mathrm{Markov}\,(p_0, P)$, with Markov transition kernel defined by a probability density $p : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$:

$$P(x, A) = \mathbb{P}\left(X_{n+1} \in A \mid X_n = x\right) = \int_A p(x, y)\, dy, \quad A \in \mathcal{B}(\mathbb{R}^d),$$

and initial probability p_0, i.e. $X_0 \sim p_0$. We are interested in computing

$$\mu = \mathbb{E}\left[Z\right] = \mathbb{E}\left[\psi(X_0, \ldots, X_m)\right]$$

for some finite horizon $m \in \mathbb{N}$. Importance sampling in this case can be done by replacing the transition probability density function by another one $q : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ which dominates p, i.e. $q(x, y) = 0 \implies p(x, y) = 0$, and the initial density p_0 by a dominating one q_0. We require moreover that $\mathbb{P}_q(\tau < +\infty) = 1$. By successive conditioning, we have

$$\begin{aligned}
\mu = \mathbb{E}_p[\psi(X_0, \ldots, X_m)] &= \mathbb{E}_{X_{0:m-1}\sim p}[\,\mathbb{E}_{X_m \sim p}[\psi(X_0, \ldots, X_m)|X_{m-1}, \ldots, X_0]\,] \\
&= \mathbb{E}_{X_{0:m-1}\sim p}\left[\int \psi(X_0, \ldots, X_{m-1}, z)p(X_{m-1}, z)dz\right] \\
&= \mathbb{E}_{X_{0:m-1}\sim p}\left[\int \psi(X_0, \ldots, X_{m-1}, z)\frac{p(X_{m-1}, z)}{q(X_{m-1}, z)}q(X_{m-1}, z)dz\right] \\
&= \mathbb{E}_{X_{0:m-1}\sim p}\left[\mathbb{E}_{X_m\sim q}[\psi(X_0, \ldots, X_{m-1}, X_m)\frac{p(X_{m-1}, X_m)}{q(X_{m-1}, X_m)}|X_{m-1}, \ldots, X_0]\right] \\
&= \mathbb{E}_{X_{0:m}\sim q}\left[\psi(X_0, \ldots, X_m)\frac{p_0(X_0)}{q_0(X_0)}\prod_{j=1}^{m}\frac{p(X_{j-1}, X_j)}{q(X_{j-1}, X_j)}\right] \\
&= \mathbb{E}_{X_{0:m}\sim q}\left[\psi(X_0, \ldots, X_m)w(X_{0:m})\right]
\end{aligned}$$

with likelihood ratio

$$w(X_{0:m}) = \frac{p_0(X_0)}{q_0(X_0)}\prod_{j=1}^{m}\frac{p(X_{j-1}, X_j)}{q(X_{j-1}, X_j)}.$$

The previous argument can also be adapted to the case in which the process is stopped at some stopping time τ, e.g. $\tau = \inf\{n : X_n \in B \in \mathcal{B}(\mathbb{R}^d)\}$. Suppose now we want to compute the quantity

$$\mu = \mathbb{E}\left[Z\right] = \mathbb{E}\left[\psi_\tau(X_0, \ldots, X_\tau)\mathbb{1}_{\{\tau<+\infty\}}\right].$$

When doing importance sampling with the dominating transition probability q, we require that $\mathbb{P}_q(\tau < +\infty) = 1$. Then. it can be shown that (exercise)

$$\mu = \mathbb{E}_p[Z] = \mathbb{E}_p[\psi_\tau(X_0, \ldots, X_\tau)\mathbb{1}_{\{\tau<+\infty\}}] = \mathbb{E}_q[\psi_\tau(X_0, \ldots, X_\tau)w(X_{0:\tau})]$$

and $\mu = \mathbb{E}_p[Z]$ can be estimated by the following algorithm.

Algorithm 6.4: Importance sampling for Markov processes.

1 Generate N iid paths $X^{(i)}_{0:\tau^{(i)}} = (X^{(i)}_0, \ldots, X^{(i)}_{\tau^{(i)}})$ up to the stopping time $\tau^{(i)}$, of the Markov chain with transition probability $q : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ and initial probability $q_0 : \mathbb{R}^d \to \mathbb{R}_+$

2 Compute $\hat{\mu}_{\text{IS}} = \frac{1}{N}\sum_{i=1}^{N}\psi_{\tau^{(i)}}(X^{(i)}_{0:\tau^{(i)}})w(X^{(i)}_{0:\tau^{(i)}})$

3 Output $\hat{\mu}_{\text{IS}}$ and a confidence interval based on $\hat{\sigma}_{\text{IS}}$.

Continuous time discrete space Markov processes. Consider a continuous time Markov process $\{X_t \in \mathcal{X},\, t \geq 0\}$ taking values in the discrete space $\mathcal{X} = \{y_1, \boldsymbol{y}_2, \ldots\}$, defined by the stable and conservative generator matrix $(Q_{ij})_{ij}$ (see Section 4.7) and the initial distribution $\boldsymbol{X}_0 \sim \lambda = (\lambda_1, \lambda_2, \ldots)$, with $\lambda_i = \mathbb{P}(X_0 = y_i)$. We aim at computing

$$\mu = \mathbb{E}\left[Z\right] = \mathbb{E}\left[\psi(\{X_t\}_{0\leq t\leq T})\right]$$

where ψ is a function of the path X_t, $t \in [0, T]$ as, for example, $\psi(\{X_t\}_{0\le t\le T}) = \int_0^T \phi(X_s)ds$ or $\psi(\{X_t\}_{0\le t\le T}) = \max_{0\le t\le T} X_t$. We can do importance sampling in this case by changing the generator matrix to $\tilde{Q}$, and the initial distribution to $\tilde{\lambda}$, with the conditions that $\tilde{Q}_{ij} = 0$ and $\tilde{\lambda}_k = 0$ only if $Q_{ij} = 0$ and $\lambda_k = 0$, respectively. Then, if we denote by $N(t)$ the number of jumps of $\{X_t\}$ occurred in $[0, t]$, by J_n, $n = 1, \dots, N(T)$ the jump times, by $S_n = J_n - J_{n-1}$ the holding times, and by $Y_n = X_{J_n}$ the jump process, it can be shown that

$$\mu = \mathbb{E}_{\lambda,Q}[\psi(\{X_t\}_{0\le t\le T})] = \mathbb{E}_{\tilde{\lambda},\tilde{Q}}[\psi(\{X_t\}_{0\le t\le T})w(\{X_t\}_{0\le t\le T})]$$

with likelihood ratio given by

$$\begin{aligned}
w(\{X_t\}_{0\le t\le T}) &= \frac{\lambda_{X_0}}{\tilde{\lambda}_{X_0}} \left(\prod_{i=1}^{N(T)} \frac{Q_{Y_{i-1}Y_i}}{\tilde{Q}_{Y_{i-1}Y_i}} \frac{\exp\{-S_j Q_{Y_{j-1}}\}}{\exp\{-S_j \tilde{Q}_{Y_{j-1}}\}} \right) \frac{\exp\{-(T - J_{N(T)})Q_{Y_{N(T)}}\}}{\exp\{-(T - J_{N(T)})\tilde{Q}_{Y_{N(T)}}\}} \\
&= \frac{\lambda_{X_0}}{\tilde{\lambda}_{X_0}} \left(\prod_{i=1}^{N(T)} \frac{Q_{Y_{i-1}Y_i}}{\tilde{Q}_{Y_{i-1}Y_i}} \right) \exp\left\{ - \int_0^T (Q_{Y_s} - \tilde{Q}_{Y_s})ds \right\}
\end{aligned}$$

where we recall that $Q_i = -Q_{ii} = \sum_{j\ne i} Q_{ij}$.

6.3 Control variates

We consider again the goal of computing the expected value $\mu = \mathbb{E}[Z]$ of a random variable Z, output of a stochastic model. The idea of the control variate technique is to find an auxiliary variable Y, called *control variate*, of which we *know the mean value*, and which is strongly correlated with the variable Z. We can then construct the modified variable

$$\tilde{Z}_\alpha = Z - \alpha(Y - \mathbb{E}[Y])$$

with $\alpha \in \mathbb{R}$, that satisfies

$$\mathbb{E}[\tilde{Z}_\alpha] = \mathbb{E}[Z] = \mu$$

and

$$\mathbb{V}\mathrm{ar}(\tilde{Z}_\alpha) = \mathbb{V}\mathrm{ar}(Z) + \alpha^2 \mathbb{V}\mathrm{ar}(Y) - 2\alpha\,\mathrm{Cov}(Z, Y).$$

The latter is a quadratic expression in α and is minimized for

$$\alpha_{\mathrm{opt}} = \frac{\mathrm{Cov}(Z, Y)}{\mathbb{V}\mathrm{ar}(Y)}.$$

With such optimal choice, one has

$$\mathbb{V}\mathrm{ar}\left(\tilde{Z}_{\alpha_{\mathrm{opt}}}\right) = \mathbb{V}\mathrm{ar}(Z) - \frac{\mathrm{Cov}(Z, Y)^2}{\mathbb{V}\mathrm{ar}(Y)} = \mathbb{V}\mathrm{ar}(Z)(1 - \rho_{ZY}^2), \qquad \rho_{ZY}^2 = \frac{\mathrm{Cov}(Z, Y)^2}{\mathbb{V}\mathrm{ar}(Z)\,\mathbb{V}\mathrm{ar}(Y)}$$

which is always smaller than $\mathbb{V}\mathrm{ar}(Z)$. The amount of variance reduction increases as ρ_{ZY} approaches 1 or -1. It is clear that the ideal control variate is $Y = \gamma Z$, $\gamma \in \mathbb{R}$ for which

$\mathbb{V}\mathrm{ar}(\tilde{Z}_{\alpha_{\mathrm{opt}}}) = 0$. However, $\mathbb{E}[Y] = \gamma\mathbb{E}[Z]$ is not known in this case, and such a control variate is not a viable option. The control variate Y should be a reasonable approximation of Z, of which, however, we can compute exactly its expected value, or, more generally, a random variable highly informative on Z (hence highly correlated to Z). In practice, the optimal α is not known, but it can be estimated from a pilot run.

Algorithm 6.5: Control variate with pilot run.

1 Generate $\bar{N}$ iid replicas $(Z^{(i)}, Y^{(i)})$, $i = 1, \dots, \bar{N}$ of (Z, Y)

2 Estimate $\hat{\alpha}_{\mathrm{opt}} = \frac{\hat{\sigma}^2_{ZY}}{\sigma^2_Y}$ if σ^2_Y known, or $\hat{\alpha}_{\mathrm{opt}} = \frac{\hat{\sigma}^2_{ZY}}{\hat{\sigma}^2_Y}$ with

$$\hat{\sigma}^2_{ZY} = \frac{1}{\bar{N}-1}\sum_{i=1}^{\bar{N}}(Z^{(i)} - \hat{\mu}_Z)(Y^{(i)} - \mathbb{E}[Y]), \qquad \hat{\mu}_Z = \frac{1}{\bar{N}}\sum_{i=1}^{\bar{N}} Z^{(i)}$$

3 Generate N iid replicas $(Z^{(i)}, Y^{(i)})$ $i = 1, \dots, N$ of (Z, Y)

4 Compute $\hat{\mu}_{\mathrm{CV}} = \frac{1}{N}\sum_{i=1}^{N}(Z^{(i)} - \hat{\alpha}_{\mathrm{opt}}(Y^{(i)} - \mathbb{E}[Y]))$

5 Output $\hat{\mu}_{\mathrm{CV}}$ and a confidence interval based on $\hat{\sigma}_{CV}$.

The estimator $\hat{\mu}_{\mathrm{CV}}$ is unbiased and has variance (exercise)

$$\mathbb{V}\mathrm{ar}\left(\hat{\mu}_{\mathrm{CV}}\right) = \mathbb{E}\left[(\hat{\mu}_{\mathrm{CV}} - \mu)^2\right] = \frac{1}{N}\left(\mathbb{V}\mathrm{ar}\left(\tilde{Z}_{\alpha_{\mathrm{opt}}}\right) + \mathbb{V}\mathrm{ar}\left(\hat{\alpha}_{\mathrm{opt}}\right)\sigma^2_Y\right).$$

In the case σ^2_Y known, we have $\mathbb{V}\mathrm{ar}\left(\hat{\alpha}_{\mathrm{opt}}\right) = O(1/\bar{N})$ since $\hat{\alpha}_{\mathrm{opt}}$ is a Monte Carlo estimator. Moreover, $\mathbb{V}\mathrm{ar}\left(\tilde{Z}_{\alpha_{\mathrm{opt}}}\right)$ can be estimated by the estimator

$$\hat{\sigma}^2(\tilde{Z}_{\alpha_{\mathrm{opt}}}) = \hat{\sigma}^2_Z - \frac{\hat{\sigma}^2_{ZY}}{\sigma^2_Y}$$

which is unbiased if σ^2_Y is known, and $\sqrt{N}\frac{\hat{\mu}_{\mathrm{CV}} - \mu}{\hat{\sigma}^2(\tilde{Z}_{\alpha_{\mathrm{opt}}})} \xrightarrow{\mathrm{d}} N(0,1)$ as $N, \bar{N} \to \infty$ from which asymptotic confidence intervals can be obtained.

Alternative to the previous algorithm, which uses a pilot run to estimate α_{opt}, one may consider a "one-shot" strategy.

Algorithm 6.6: Control variate – one shot

1 Generate N iid replicas $(Z^{(i)}, Y^{(i)})$, $i = 1, \dots, N$ of (Z, Y)

2 Estimate $\hat{\alpha}_{\mathrm{opt}} = \frac{\hat{\sigma}^2_{ZY}}{\sigma^2_Y}$, with

$$\hat{\sigma}^2_{ZY} = \frac{1}{N-1}\sum_{i=1}^{N}(Z^{(i)} - \hat{\mu}_Z)(Y^{(i)} - \mathbb{E}[Y]), \qquad \hat{\mu}_Z = \frac{1}{N}\sum_{i=1}^{N} Z^{(i)}$$

3 Estimate $\hat{\mu}_{\mathrm{CV}} = \frac{1}{N}\sum_{i=1}^{N}(Z^{(i)} - \hat{\alpha}_{\mathrm{opt}}(Y^{(i)} - \mathbb{E}[Y]))$

4 Output $\hat{\mu}_{\mathrm{CV}}$ and a confidence interval based on $\hat{\sigma}_{CV}$.

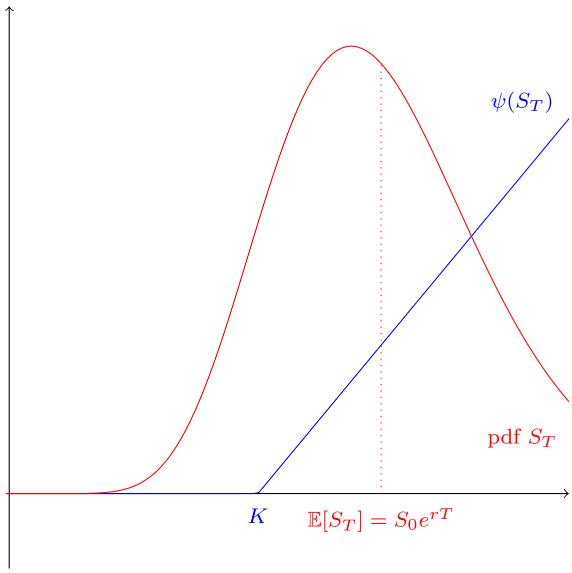


Figure 6.3: European option.

This estimator is biased, in general. However, a CLT result still holds (exercise) and

$$\sqrt{N}\frac{\hat{\mu}_{\text{CV}} - \mu}{\hat{\sigma}^2(\tilde{Z}_{\alpha_{\text{opt}}})} \xrightarrow{\text{d}} N(0,1)$$

as $N \to \infty$ from which asymptotic confidence intervals can be obtained.

Example 6.5. *Consider again the problem of pricing a European call option: $\mu = \mathbb{E}[Z]$, with $Z = \psi(S_T) = e^{-rT}(S_T - K)_+$, $S_T = S_0 e^{X_T}$ and $X_T \sim N((r - \sigma^2/2)T, \sigma^2 T)$. To compute $\mathbb{E}[Z]$ with Monte Carlo, we can use as a control variable the variable $Y = S_T$ whose exact mean is $\mathbb{E}[Y] = \mathbb{E}[S_T] = S_0 e^{rT}$. Observe that, since ψ is a non decreasing function of S_T, Z and Y are positively correlated, so that α should be taken positive. If the sample mean $\hat{\mu}_{S_T} = \frac{1}{N}\sum_{i=1}^N S_T^{(i)}$ is above the true mean $S_0 e^{rT}$, it is reasonable to assume that also the sample mean $\hat{\mu}_Z$ will be above the true (unknown) mean, since (Z, Y) are positively correlated, so we add a negative correctin to $\hat{\mu}_Z$ given by $-\alpha(\hat{\mu}_{S_T} - S_0 e^{rT})$, with $\alpha > 0$.*

6.3.1 Multiple control variates

The control variate technique can be generalized to the case in which multiple control variates $Y_1, \ldots, Y_p$ are used. We define the modified variable

$$\tilde{Z}_\alpha = Z - \sum_{j=1}^{p} \alpha_j (Y_j - \mathbb{E}[Y_j]) = Z - \boldsymbol{\alpha} \cdot (\boldsymbol{Y} - \mathbb{E}[\boldsymbol{Y}])$$

with $\boldsymbol{Y} = (Y_1, \dots, Y_p)$ and $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_p)$. Then

$$\begin{aligned}\mathbb{V}\mathrm{ar}\left(\tilde{Z}_\alpha\right) &= \mathbb{E}\left[(Z - \mu - \boldsymbol{\alpha}\cdot(\boldsymbol{Y} - \mathbb{E}[\boldsymbol{Y}]))^2\right] \\ &= \mathbb{V}\mathrm{ar}(Z) - 2\,\mathrm{Cov}(Z, \boldsymbol{Y})\cdot\boldsymbol{\alpha} + \boldsymbol{\alpha}^\top\,\mathrm{Cov}(\boldsymbol{Y}, \boldsymbol{Y})\boldsymbol{\alpha}\end{aligned}$$

where $\mathrm{Cov}(Z, \boldsymbol{Y}) = (\mathrm{Cov}(Z, Y_i))_{i=1}^p \in \mathbb{R}^p$ and $\mathrm{Cov}(\boldsymbol{Y}, \boldsymbol{Y}) = (\mathrm{Cov}(Y_i, Y_j))_{i,j=1}^p \in \mathbb{R}^{p\times p}$. Again $\mathbb{V}\mathrm{ar}\left(\tilde{Z}_\alpha\right)$ is a quadratic function in $\boldsymbol{\alpha}$ and is minimized by

$$\boldsymbol{\alpha}_{\mathrm{opt}} = \mathrm{Cov}(\boldsymbol{Y}, \boldsymbol{Y})^{-1}\,\mathrm{Cov}(Z, \boldsymbol{Y}).$$

Algorithm 6.7: Multiple control variates – one shot

1 Generate N iid replicas $(Z^{(i)}, Y_1^{(i)}, \dots, Y_p^{(i)})$ of $(Z, \boldsymbol{Y})$

2 Estimate

$$(\hat{\sigma}^2_{Z\boldsymbol{Y}})_j = \frac{1}{N-1}\sum_{i=1}^N (Z^{(i)} - \hat{\mu}_Z)(Y_j^{(i)} - \mathbb{E}[Y_j]), \quad j = 1, \dots, p$$

and

$$(\hat{\sigma}^2_{\boldsymbol{YY}})_{jk} = \frac{1}{N}\sum_{i=1}^N (Y_j^{(i)} - \mathbb{E}[Y_j])(Y_k^{(i)} - \mathbb{E}[Y_k]).$$

3 Estimate the optimal $\boldsymbol{\alpha}_{\mathrm{opt}}$ by $\hat{\alpha}_{\mathrm{opt}} = (\hat{\sigma}^2_{\boldsymbol{YY}})^{-1}\hat{\sigma}^2_{Z\boldsymbol{Y}}$

4 Compute $\hat{\mu}_{\mathrm{CV}} = \frac{1}{N}\sum_{i=1}^N (Z^{(i)} - \hat{\alpha}_{\mathrm{opt}}\cdot(\boldsymbol{Y}^{(i)} - \mathbb{E}[\boldsymbol{Y}]))$.

5 Output $\hat{\mu}_{\mathrm{CV}}$ and a confidence interval based on $\hat{\sigma}_{CV}$.

6.4 Stratification

As in the previous sections, we consider the problem of computing $\mu = \mathbb{E}[Z]$ where Z is the output of a stochastic model. We assume here that $Z = \psi(X_1, \dots, X_d) = \psi(X)$ where $X \in \mathbb{R}^d$ is a random vector with pdf $f : \Omega \subset \mathbb{R}^d \to \mathbb{R}_+$ so that $\mu = \int_\Omega \psi(x) f(x)\,dx$.

The idea of stratification is to divide the sample space Ω into S non overlapping regions $\Omega_1, \dots, \Omega_S$ called *strata* such that $\mathbb{P}(X \in \Omega_j) = \int_\Omega \mathbb{1}_{\Omega_j}(x) f(x)\,dx = p_j$ is known and $\sum_{j=1}^s p_j = 1$. Assume now that we can generate X conditional upon $X \in \Omega_j$. The conditional density of X given $X \in \Omega_j$ is $f_j(x) = \frac{1}{p_j} f(x)\mathbb{1}_{\{X\in\Omega_j\}}$. Let now $X_j \sim f_j$ and $Z_j = \psi(X_j), \quad j = 1, \dots, s$. Clearly, $\mu = \mathbb{E}[Z] = \sum_{j=1}^s \mathbb{E}[Z \mid X \in \Omega_j]\,\mathbb{P}(X \in \Omega_j) = \sum_{i=1}^s p_j \mathbb{E}[Z_j]$. The idea is then to sample independently each $Z_j = \psi(X_j)$ leading to the following stratified estimator

$$\hat{\mu}_{\mathrm{Str}} = \sum_{j=1}^s p_j \hat{\mu}_i, \quad \hat{\mu}_i = \frac{1}{N_j}\sum_{i=1}^{N_j} Z_j^{(i)}, \qquad \text{with } Z_j^{(i)} \overset{\text{iid}}{\sim} Z_j. \tag{6.1}$$

The stratified estimator (6.1) has the following properties:

1. The estimator $\hat{\mu}_{\text{Str}}$ is *unbiased.* Indeed,

$$\mathbb{E}\left[\hat{\mu}_{\text{Str}}\right] = \sum_{j=1}^{s} p_j \mathbb{E}\left[\hat{\mu}_j\right] = \sum_{j=1}^{s} p_j \mathbb{E}\left[Z_j\right] = \mathbb{E}\left[Z\right].$$

2. The variance of the estimator satisfies

$$\mathbb{V}\text{ar}\left(\hat{\mu}_{\text{Str}}\right) = \sum_{j=1}^{s} p_j^2 \mathbb{V}\text{ar}\left(\hat{\mu}_j\right) = \sum_{j=1}^{s} p_j^2 \frac{\mathbb{V}\text{ar}\left(Z_j\right)}{N_j}$$

   and can be estimated by

$$\hat{\sigma}_{Str}^2 = \sum_{j=1}^{s} p_j^2 \frac{\hat{\sigma}_j^2}{N_j}, \qquad \hat{\sigma}_j^2 = \frac{1}{N_j - 1} \sum_{i=1}^{N_j} (Z_j^{(i)} - \hat{\mu}_j)^2.$$

3. Let $N = \sum_{j=1}^{s} N_j$ and choose $N_j = \phi(N)$ such that $\lim_{N\to\infty} \frac{N}{N_j} < +\infty$ for any $j = 1, \dots, s$. Then $\lim_{N\to\infty} N\mathbb{V}\text{ar}\left(\hat{\mu}_{\text{Str}}\right) < +\infty$ and it can be shown (exercise) that

$$\frac{\hat{\mu}_{\text{Str}} - \mu}{\sqrt{\mathbb{V}\text{ar}\left(\hat{\mu}_{\text{Str}}\right)}} \xrightarrow{\text{d}} N(0,1), \quad \text{as } N \to \infty.$$

   Therefore, a computable $1-\alpha$ asymptotic confidence interval is given by

$$I_\alpha = [\hat{\mu}_{\text{Str}} - c_{1-\alpha/2}\hat{\sigma}_{Str},\ \hat{\mu}_{\text{Str}} + c_{1-\alpha/2}\hat{\sigma}_{Str}]$$

We summarize the procedure in the following Algorithm.

Algorithm 6.8: Stratification

for $j = 1, \dots, s$ **do**
  Generate N_j iid replicas $Z_j^{(i)}$, $i = 1, \dots, N_j$ of Z_j
  Compute $\hat{\mu}_j = \frac{1}{N_j} \sum_{i=1}^{N_j} Z_j^{(i)}$ and $\hat{\sigma}_j^2 = \frac{1}{N_j - 1} \sum_{i=1}^{N_j} (Z_j^{(i)} - \hat{\mu}_j)^2$
end
Compute $\hat{\mu}_{\text{Str}} = \sum_{j=1}^{s} p_j \hat{\mu}_j$ and $\hat{\sigma}_{Str}^2 = \sum_{j=1}^{s} p_j^2 \frac{\hat{\sigma}_j^2}{N_j}$
Output $\hat{\mu}_{\text{Str}}$ and a confidence interval $I_\alpha = [\hat{\mu}_{\text{Str}} - c_{1-\alpha/2}\hat{\sigma}_{Str},\ \hat{\mu}_{\text{Str}} + c_{1-\alpha/2}\hat{\sigma}_{Str}]$

Stratification guarantees that each stratum contains a fixed number of evaluations. It remains the question of how to choose N_j in each stratun and quantify the amount of variance reduction that we can achieve.

6.4.1 Proportional allocation

If N is the total sample size, proportional allocation simply chooses $N_j = Np_j$. With this choice, we have

$$\mathbb{V}\text{ar}\left(\hat{\mu}_{\text{Str}}\right) = \sum_{j=1}^{s} p_j^2 \frac{\mathbb{V}\text{ar}\left(Z_j\right)}{N_j} = \frac{1}{N} \sum_{j=1}^{s} p_j \mathbb{V}\text{ar}\left(Z_j\right).$$

Defining the discrete random variable $J \in \{1, \ldots, s\}$, $J = j \iff \{X \in \Omega_j\}$, we can rewrite $\mathbb{V}\mathrm{ar}(\hat{\mu}_{\mathrm{Str}})$ as

$$\mathbb{V}\mathrm{ar}(\hat{\mu}_{\mathrm{Str}}) = \frac{1}{N}\sum_{j=1}^{s} p_j \mathbb{V}\mathrm{ar}(Z \mid J = j) = \frac{1}{N}\mathbb{E}\left[\mathbb{V}\mathrm{ar}(Z \mid X \in \Omega_j)\right]$$

and, recalling the law of total variance $\mathbb{V}\mathrm{ar}(Z) = \mathbb{V}\mathrm{ar}(\mathbb{E}[Z \mid J]) + \mathbb{E}[\mathbb{V}\mathrm{ar}(Z \mid J)]$ we have

$$\mathbb{V}\mathrm{ar}(\hat{\mu}_{\mathrm{Str}}) = \frac{1}{N}\left(\mathbb{V}\mathrm{ar}(Z) - \mathbb{V}\mathrm{ar}(\mathbb{E}[Z \mid J])\right) \leq \frac{\mathbb{V}\mathrm{ar}(Z)}{N} = \mathbb{V}\mathrm{ar}(\hat{\mu}_{\mathrm{CMC}}).$$

Hence proportional allocation always leads to variance reduction. The amount of variance reduction is given by $\gamma = \mathbb{E}[\mathbb{V}\mathrm{ar}(Z \mid J)] / \mathbb{V}\mathrm{ar}(Z)$.

Example 6.6. *Let $X \sim \mathcal{U}(0,1)$ and $Z = \psi(X)$ for some function $\psi : [0,1] \to \mathbb{R}$. To compute $\mu = \mathbb{E}[Z] = \int_0^1 \psi(x)\,dx$, we could use stratification by dividing the interval $\Omega = (0,1)$ in s subintervals of equal size, $\Omega_j = \left(\frac{j-1}{s}, \frac{j}{s}\right)$, $j = 1, \ldots, s$. Then*

$$\mu = \sum_{j=1}^{s} \int_{\frac{j-1}{s}}^{\frac{j}{s}} \psi(x)\,dx = \sum_{j=1}^{s} \frac{1}{s}\int_{\frac{j-1}{s}}^{\frac{j}{s}} \psi(x) s\,dx = \sum_{j=1}^{s} \frac{1}{s}\mathbb{E}[\psi(X_j)], \quad \textit{with } X_j \sim \mathcal{U}\left(\frac{j-1}{s}, \frac{j}{s}\right)$$

and a stratified estimator reads

$$\hat{\mu}_{Str} = \sum_{j=1}^{s} \frac{1}{s}\frac{1}{N_j}\sum_{i=1}^{N_j} \psi(X_j^{(i)}), \qquad \textit{with } X_j^{(i)} \overset{iid}{\sim} \mathcal{U}\left(\frac{j-1}{s}, \frac{j}{s}\right).$$

Figure 6.4 gives an illustration of the stratification procedure with 7 strata and 2 replicas per stratum.

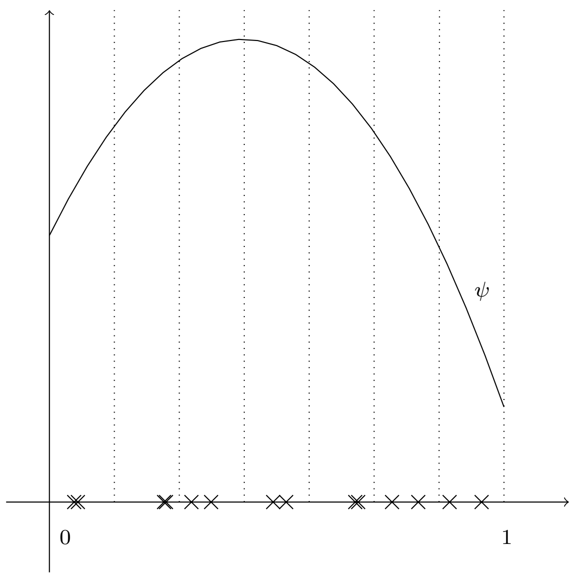
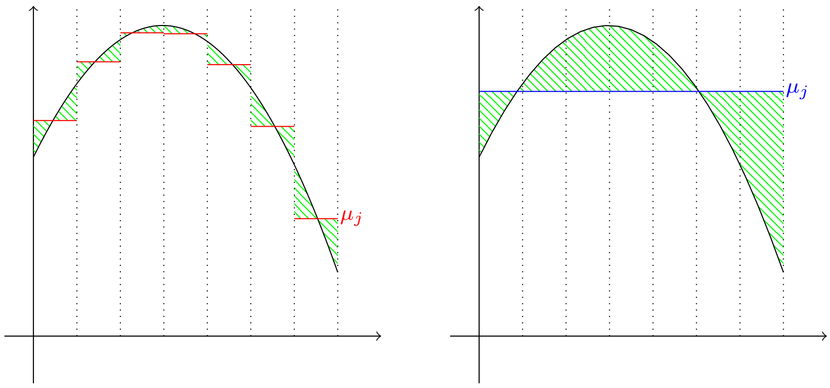


Figure 6.4: Stratification.



(a) $\mathbb{E}\left[\mathbb{V}\mathrm{ar}\left(Z \mid J\right)\right]$: sum of green 'squared' areas

(b) $\mathbb{V}\mathrm{ar}\left(Z\right)$: green 'squared' area

Figure 6.5: Proportional allocation

Figure 6.5 illustrates the variance reduction when considering proportional allocation. The variance of a crude Monte Carlo estimator is proportional to the green area in the right plot, whereas the variance of the stratificed estimator is proportional to the green area in the left plot. From this graphical illustration we see that large varianve reduction has to be expected when the function psi is highly non-constant. If ψ is piecewise constant over the partion of the domain, then we even have $\mathbb{V}\mathrm{ar}\left(\hat{\mu}_{Str}\right) = 0$.

6.4.2 Optimal allocation

Instead of doing a proportional allocation, one may try to find the best choice of N_j that minimises $\mathbb{V}\mathrm{ar}\left(\hat{\mu}_{\mathrm{Str}}\right)$:

$$\{N_j^*\} = \underset{(N_1,\ldots,N_s)}{\mathrm{argmin}} \sum_{j=1}^{s} p_j^2 \frac{\mathbb{V}\mathrm{ar}\left(Z_j\right)}{N_j} \quad \text{such that} \quad \sum_{j=1}^{s} N_j = N.$$

Introducing a Lagrangian function $\mathcal{L}(\boldsymbol{N},\lambda) = \sum_{j=1}^{s} p_j^2 \frac{\mathbb{V}\mathrm{ar}(Z_j)}{N_j} + \lambda(\sum_{j=1}^{s} N_j - 1)$, we have

$$\frac{\partial \mathcal{L}}{\partial N_j} = -p_j^2 \frac{\mathbb{V}\mathrm{ar}\left(Z_j\right)}{N_j^2} + \lambda = 0 \qquad \Longrightarrow \qquad N_j = p_j \sqrt{\frac{\mathbb{V}\mathrm{ar}\left(Z_j\right)}{\lambda}}$$

and, enforcing the constraint $\sum_j N_j = N$, we obtain $\sqrt{\lambda} = \frac{\sum_{j=1}^{s} p_j \sqrt{\mathbb{V}\mathrm{ar}(Z_j)}}{N}$ which leads to the optimal choice

$$N_j^* = \frac{N p_j \sigma_j}{\sum_{k=1}^{s} p_k \sigma_k}, \qquad \sigma_j = \sqrt{\mathbb{V}\mathrm{ar}\left(Z_j\right)}$$

and optimal variance $\mathbb{V}\mathrm{ar}\left(\hat{\mu}^*_{\mathrm{Str}}\right) = \frac{1}{N}\left(\sum_{j=1}^{s} p_j \sigma_j\right)^2$.

Since this variance is smaller than that with proportional allocation, stratification with optimal allocation will always lead to variance reduction. In practice, the σ_j are not known and can be obtained from a pilot run.

Algorithm 6.9: Stratification with optimal allocation

for $j = 1, \dots, s$ **do**
  Generate $\bar{N}_j$ iid replicas $Z_j^{(i)}, \quad i = 1, \dots, N_j$ of Z_j
  Estimate $\hat{\sigma}_j^2 = \frac{1}{\bar{N}_j - 1} \sum_{i=1}^{\bar{N}_j} (Z_j^{(i)} - \hat{\mu}_j)^2$
end
Choose $N = (c_{1-\alpha/2} \sum_{j=1}^{s} p_j \hat{\sigma}_j / \mathrm{tol})^2$ (to guarantee that $|\hat{I}_{\alpha,N}| < 2\mathrm{tol}$)
For $j = 1, \dots, s$, generate $N_j^* = \frac{N p_j \hat{\sigma}_j}{\sum_k p_k \hat{\sigma}_k}$ iid replicas $Z_j^{(i)}$ of Z_j
Compute $\hat{\mu}_i = \frac{1}{N_j^*} \sum_{i=1}^{N_j^*} Z_j^{(i)}$ and $\hat{\mu}_{\mathrm{Str}}^* = \sum_{j=1}^{s} p_j \hat{\mu}_j$

6.5 Latin Hypercube Sampling

Consider the problem of computing the expected value μ of $Z = \psi(X_1, \dots, X_d)$ where $X_j \in \mathbb{R}$ are independent and with pdf $f_j : \mathbb{R} \to \mathbb{R}_+$. One might want to stratify each variable X_j in s strata. However, this would lead to s^d strata which becomes unaffordable for large d. A way to overcome this problem is offered by the Latin Hypercube Sampling (LHS). For simplicity of exposition, let us assume that $X = (X_1, \dots, X_d) \sim \mathcal{U}([0,1]^d)$. The idea of LHS is to stratify each component X_j but not the whole sampling domain $\Omega = [0,1]^d$. In particular, N (correlated) points $X^{(i)}$, $i = 1, \dots, N$ are drawn in $[0,1]^d$ in such a way that each component is stratified with N strata and one point per stratum. Figure 6.6 illustrates the idea.

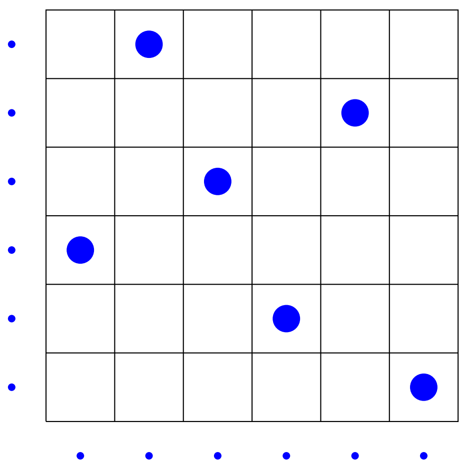


Figure 6.6: Latin hypercube.

A Latin hypercube sampling design can be generated by the following Algorithm.

Algorithm 6.10: LHS

1 Generate N iid points $\boldsymbol{U}^{(i)} \overset{\text{iid}}{\sim} \mathcal{U}((0,1)^d)$, $i = 1, \ldots, N$
2 Generate d independent permutations π_j, $j = 1, \ldots, d$ of $\{1, \ldots, N\}$. Let $\boldsymbol{\pi}^{(i)} = (\pi_1(i), \pi_2(i), \ldots, \pi_d(i))$
3 Return $X^{(i)} = \frac{\boldsymbol{\pi}^{(i)} - 1 + \boldsymbol{U}^{(i)}}{N}$, $\quad i = 1, \ldots, N$.

Once the LHS desing generated, the LHS estimator of $\mu = \mathbb{E}\left[\psi(X)\right]$ is simply

$$\hat{\mu}_{\text{LHS}} = \frac{1}{N} \sum_{i=1}^{N} \psi(X^{(i)}).$$

The following proposition illustrates the two main properties of the LHS sample and estimator.

Proposition 6.4. *Let $\{X^{(i)}, i = 1, \ldots, N\}$ be a LHS design. Then*

- $X^{(i)} \sim \mathcal{U}((0,1)^d)$ *(not independent, though)*
- *The LHS estimator is unbiased,* $\mathbb{E}\left[\hat{\mu}_{LHS}\right] = \mathbb{E}\left[\psi(X)\right]$.

Proof. By construction, each vector $X^{(i)} = \frac{\boldsymbol{\pi}^{(i)} - 1 + \boldsymbol{U}^{(i)}}{N}$ has independent components. Therefore it is enough to show that each component $X_j^{(i)}$, $j = 1, \ldots, d$, is uniformly distributed in $(0,1)$. Now, $\pi_j^{(i)} = \pi_j(i)$ is the i-th component of a random permutation of $\{1, \ldots, N\}$, hence $\mathbb{P}\left(\pi_j^{(i)} = k\right) = \frac{1}{N}$ for all $k = 1, \ldots, N$. Moreover, the conditional cumulative distribution function of $X_j^{(i)}$ given $\pi_j^{(i)} = k$ is

$$F_{X_j^{(i)} \mid \pi_j^{(i)} = k}(x) = \mathbb{P}\left(X_j^{(i)} \leq x \mid \pi_j^{(i)} = k\right) = \begin{cases} 0, & x < \frac{k-1}{N} \\ Nx - k + 1, & x \in \left[\frac{k-1}{N}, \frac{k}{N}\right] \\ 1, & x > \frac{k}{N} \end{cases}$$

i.e. $X_j^{(i)} \mid \pi_j^{(i)} = k$ has distribution $\mathcal{U}\left(\frac{k-1}{N}, \frac{k}{N}\right)$ and

$$\mathbb{P}\left(X_j^{(i)} \leq x\right) = \sum_{k=1}^{N} \frac{1}{N} \mathbb{P}\left(X_j^{(i)} \leq x \mid \pi_j^{(i)} = k\right) = x.$$

From the uniform distribution of each $X^{(i)}$, it follows immediately that $\mathbb{E}\left[\hat{\mu}_{\text{LHS}}\right] = \mathbb{E}\left[\frac{1}{N} \sum_i \psi(X^{(i)})\right] = \mathbb{E}\left[\psi(X)\right]$. □

Concerning the variance of the estimator $\hat{\mu}_{\text{LHS}}$, we mention the following two results.

Proposition 6.5 (A. Owen 1997)**.** *Let $Z = \psi(X)$, $X \sim \mathcal{U}((0,1)^d)$, with $\mu = \mathbb{E}\left[Z\right] < +\infty$ and $\sigma^2 = \mathbb{V}\text{ar}\left(Z\right) < +\infty$. The LHS estimator $\hat{\mu}_{LHS}$ based on N points satisfies*

$$\mathbb{V}\text{ar}\left(\hat{\mu}_{LHS}\right) \leq \frac{\sigma^2}{N-1}.$$

This result shows that, asymptotically, $\mathbb{V}\mathrm{ar}\left(\hat{\mu}_{\mathrm{LHS}}\right)$ is not worse than $\mathbb{V}\mathrm{ar}\left(\hat{\mu}_{\mathrm{CMC}}\right) = \frac{\sigma^2}{n}$ since $\lim_{N\to\infty} \mathbb{V}\mathrm{ar}\left(\hat{\mu}_{\mathrm{LHS}}\right)/\mathbb{V}\mathrm{ar}\left(\hat{\mu}_{\mathrm{CMC}}\right) \leq 1$. Moreover, LHS is very effective if the function $\psi(X)$ has an additive structure $\psi(X) = \mu + \sum_{i=1}^{d} \psi_j(X_j)$ as the estimator $\hat{\mu}_{\mathrm{LHS}}$ corresponds to a stratified estimator with N strata on each function ψ_j. For a general $\psi : \mathbb{R}^d \to \mathbb{R}$, let

$$\hat{\psi}_j(x_j) = \int_{[0,1]^{d-1}} \left(\psi(x_1, \ldots, x_d) - \mu\right) dx_1 \ldots dx_{j-1} dx_{j+1} \ldots dx_d$$

and $\psi^{\mathrm{add}}(X) = \mathbb{E}\left[\psi\right] + \sum_{i=1}^{d} \hat{\psi}_j(x_j)$. Then, it can be shown that:

Proposition 6.6 (Stein 1987). *For $Z = \psi(X)$, $X \sim \mathcal{U}((0,1)^d)$ and $\mu = \mathbb{E}\left[Z\right] < +\infty$, $\sigma^2 = \mathbb{V}\mathrm{ar}\left(Z\right) < +\infty$ and $\hat{\mu}_{LHS}$, the LHS estimator for μ based on N points satisfies*

$$\mathbb{V}\mathrm{ar}\left(\hat{\mu}_{LHS}\right) = \frac{\mathbb{V}\mathrm{ar}\left(\psi - \psi^{add}\right)}{N} + o\left(\frac{1}{N}\right).$$

Moreover, if $\mathbb{E}\left[Z^4\right] < +\infty$, then $\sqrt{N}(\hat{\mu}_{LHS} - \mu) \xrightarrow{d} N(0, \mathbb{V}\mathrm{ar}\left(\psi - \psi^{add}\right))$ as $N \to \infty$.

This result highlights the amount of variance reduction that can be achieved by the LHS estimator, compared to the CMC one. Unfortunately, the estimate of $\mathbb{V}\mathrm{ar}\left(\hat{\mu}_{LHS}\right)$ in Proposition 6.6 is not computable and can not be used to build confidence intervals for the estimator $\hat{\mu}_{LHS}$.

To control the error in the LHD estimator, we proceed in a different way by generating few independent replicas of $\hat{\mu}_{\mathrm{LHS}}$ and estimating its variance by a sample varaince estimator.

Algorithm 6.11: LHS estimator

---

1. Generate K independent LHS designs $\{X^{(i,j)}\}_{i=1}^{N}$ of size N, for $j = 1, \ldots, K$.
2. Compute $\hat{\mu}_{\mathrm{LHS}} = \frac{1}{K}\sum_{j=1}^{K} \frac{1}{N}\sum_{i=1}^{N} \psi(X^{(i,j)}) = \frac{1}{kN}\sum_{i,j} \psi(X^{(i,j)})$
3. Compute $\hat{\sigma}^2_{\mathrm{LHS}} = \frac{1}{K-1}\sum_{j=1}^{K}\left(\frac{1}{N}\sum_{i=1}^{N} \psi(X^{(i,j)}) - \hat{\mu}_{\mathrm{LHS}}\right)^2$
4. Output $\hat{\mu}_{\mathrm{LHS}}$ and the confidence interval
   $\hat{I}_\alpha = [\hat{\mu}_{\mathrm{LHS}} - c_{1-\alpha/2}\frac{\hat{\sigma}_{\mathrm{LHS}}}{\sqrt{K}},\ \hat{\mu}_{\mathrm{LHS}} + c_{1-\alpha/2}\frac{\hat{\sigma}_{\mathrm{LHS}}}{\sqrt{K}}]$.

Chapter 7

Quasi Monte Carlo methods

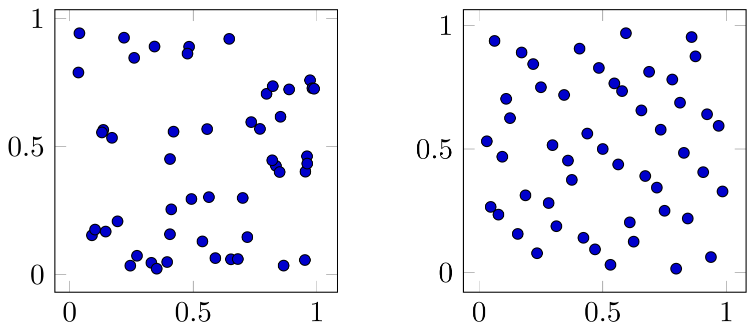
As in the previous chapter, we consider the problem of computing the expected value $\mu = \mathbb{E}[Z]$, of some random variable Z output of a stochastic model. We assume in this chapter that $Z = \psi(\boldsymbol{X})$, with $\boldsymbol{X} = (X_1, \ldots, X_d) \sim \mathcal{U}([0,1]^d)$, hence computing μ turns into computing a possibly high dimensional integral over the unit hypercube

$$\mu = \int_{[0,1]^d} \psi(x_1, \ldots, x_d)\, dx_1 \ldots dx_d.$$

A Crude Monte Carlo estimator $\hat{\mu}_{CMC}$ that uses N iid replicas of $\boldsymbol{X}$, achieves an error

$$|\mu - \hat{\mu}_{\mathrm{CMC}}| \leq c_{1-\alpha/2} \frac{\sqrt{\mathbb{V}\mathrm{ar}\,(\psi(\boldsymbol{X}))}}{\sqrt{N}}$$

with asymptotic confidence $1 - \alpha$. The idea of Quasi Monte Carlo (QMC) sampling, is to consider, instead, a *purely deterministic* sample $\{\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(N)}\}$ to improve the rate $1/\sqrt{N}$, while keeping the simple structure of the sample average estimator $\hat{\mu}_{\mathrm{QMC}} = \frac{1}{N}\sum_{i=1}^{N} \psi(X^{(i)})$ with equal weights $1/N$. It relies on the observation that a random uniform sample does not seem to cover "uniformly" the hypercube and hopefully there exist better designs that achieve this goal.



(a) random sampling

(b) QMC Sampling (Sobol sequence)

Figure 7.1: Comparing a uniform random sample (left) and a QMC sample with the same numebr of points on the unit hypercube.

Figure 7.1 shows a random sample and a QMC sample, with 50 points each, on the unit square.

The main notion behind QMC sampling is that of *discrepancy*. We introduce the following notation: for a point $\boldsymbol{y} \in [0,1]^d$, $\boldsymbol{y} = (y_1, \dots, y_d)$, we denote by $[0, \boldsymbol{y}]$ the hyper-rectangle $[0, \boldsymbol{y}] = \prod_{i=1}^d [0, y_i]$, with volume $\mathrm{Vol}([0, \boldsymbol{y}]) = \prod_{i=1}^d y_i$. For an arbitrary sample $\mathcal{P} = \{\boldsymbol{X}^{(i)}, \dots, \boldsymbol{X}^{(N)}\}$ of N points in $[0,1]^d$, hereafter called a *point set*, we introduce the empirical volume estimator for $\mathrm{Vol}([0, \boldsymbol{y}])$, based on the point set $\mathcal{P}$.

$$\widehat{\mathrm{Vol}}_{\mathcal{P}}([0, \boldsymbol{y}]) = \frac{1}{N} \sum_{i=1}^N \mathbb{1}_{[0,\boldsymbol{y}]}(\boldsymbol{X}^{(i)}) = \frac{\#\{\boldsymbol{X}^{(i)} \in [0, \boldsymbol{y}]\}}{N}.$$

Definition 7.1. *We call* ***discrepancy function*** $\Delta_{\mathcal{P}} : [0,1]^d \to [-1, 1]$ *the function*

$$\Delta_{\mathcal{P}}(\boldsymbol{y}) = \widehat{\mathrm{Vol}}_{\mathcal{P}}([0, \boldsymbol{y}]) - \mathrm{Vol}([0, \boldsymbol{y}]) = \frac{1}{N} \sum_{i=1}^N \mathbb{1}_{[0,\boldsymbol{y}]}(\boldsymbol{X}^{(i)}) - \prod_{j=1}^d y_j.$$

From $\Delta_{\mathcal{P}}$, we define the following measures of discrepancy of a point set $\mathcal{P}$:

$$\begin{aligned} &L_q\text{-discrepancy:} && D_{N,q}(\mathcal{P}) = \|\Delta_{\mathcal{P}}\|_{L^q} = \left(\int_{[0,1]^d} |\Delta_{\mathcal{P}}(\boldsymbol{y})|^q \, d\boldsymbol{y} \right)^{1/q}, \quad 1 \le q < \infty, \\ &\text{Star-discrepancy:} && D_N^*(\mathcal{P}) = \|\Delta_{\mathcal{P}}\|_{L^\infty} = \sup_{\boldsymbol{y} \in [0,1]^d} |\Delta_{\mathcal{P}}(\boldsymbol{y})|. \end{aligned}$$

Remark 7.1. *There is actually nothing special in choosing only the rectangles $[0, \boldsymbol{y}]$, so one can define also the so called* extreme discrepancy

$$D_N(\mathcal{P}) = \sup_{\substack{\boldsymbol{y}, \boldsymbol{z} \in [0,1]^d \\ \boldsymbol{z} < \boldsymbol{y}}} |\widehat{\mathrm{Vol}}_{\mathcal{P}}([\boldsymbol{z}, \boldsymbol{y}]) - \mathrm{Vol}([\boldsymbol{z}, \boldsymbol{y}])|.$$

It can be easily shown that $D_N^*(\mathcal{P}) \le D_N(\mathcal{P}) \le 2^d D_N^*(\mathcal{P})$. The left inequality is obvious and the right one follows from the observation that a rectangle $[\boldsymbol{z}, \boldsymbol{y}]$ can be written as a composition (union/intersection) of 2^d rectangles of the type $[0, \boldsymbol{z}]$. Hence, it is enough to study only the star-discrepancy.

The reason why the discrepancy plays an important role in the study of QMC quadrature formulas follows from the famous Koksma-Hlawka inequality, which we illustrate first in dimension $d = 1$. We start by deriving the following identity.

Lemma 7.1 (**Zaremba's identity**)**.** *Let $\psi : [0,1] \to \mathbb{R}$ be an absolutely continuous function with integrable derivative and let $\mathcal{P} = \{X^{(1)}, \dots, X^{(N)}\}$ be any point set in $[0,1]$. Then*

$$\begin{aligned} \int_0^1 \psi(x)\, dx - \frac{1}{N} \sum_{i=1}^N \psi(X^{(i)}) &= \int_0^1 \psi'(y) \Delta_{\mathcal{P}}(y)\, dy \qquad (7.1) \\ &= \int_0^1 \psi'(y) \Delta_{\mathcal{P}}(y)\, dy - \Delta_{\mathcal{P}}(1)\psi(1). \end{aligned}$$

Proof. Using the identity $\psi(x) = \psi(1) - \int_x^1 \psi'(y)\,dy$ in the left hand side of (7.1), we have

$$\int_0^1 \psi(x)\,dx - \frac{1}{N}\sum_{i=1}^N \psi(X^{(i)}) = \psi(1) - \underbrace{\int_0^1\int_x^1 \psi'(y)\,dy\,dx}_{=\int_0^1\int_0^y \psi'(y)\,dx\,dy} - \frac{1}{N}\sum_{i=1}^N \psi(1) + \frac{1}{N}\sum_{i=1}^N \underbrace{\int_{X^{(i)}}^1 \psi'(y)\,dy}_{=\int_0^1 \psi'(y)\mathbb{1}_{[0,y]}(X^{(i)})dy}$$
$$= \int_0^1 \psi'(y)\left[\frac{1}{N}\sum_{i=1}^N \mathbb{1}_{[0,y]}(X^{(i)}) - y\right] dy$$
$$= \int_0^1 \psi'(y)\Delta_{\mathcal{P}}(y)\,dy.$$

The second inequality follows immediatlely by observing that $\Delta_{\mathcal{P}}(1) = 0$ for any point set $\mathcal{P}$. $\square$

From the Zaremba's identity, we derive easily the **Koksma-Hlawka** inequality:

$$\left|\int_0^1 \psi(x)\,dx - \frac{1}{N}\sum_{i=1}^N \psi(X^{(i)})\right| \le \|\psi'\|_{L_p}\|\Delta_{\mathcal{P}}\|_{L_q}, \qquad \forall p,q \in [1,\infty],\ \ \frac{1}{p}+\frac{1}{q} = 1. \quad (7.2)$$

Ineqaulity (7.2) shows that the quadrature error is proportional to the discrepancy measure $\|\Delta_{\mathcal{P}}\|_{L_q}$, provided that $\psi' \in L_p(0,1)$, i.e. $\psi \in W^{1,p}(0,1)$. In particular, if ψ' is integrable (or ψ has bounded total variation) then

$$\left|\int_0^1 \psi(x)\,dx - \frac{1}{N}\sum_{i=1}^N \psi(X^{(i)})\right| \le \|\psi\|_{\mathrm{TV}} D_N^*(\mathcal{P}).$$

The previous analysis extends with same care to the multi-dimensional setting. We introduce the following notation: let $\mathbf{u} = \{u_1,\dots,u_k\} \subset \{1,\dots,d\}$ be a subset of dimensions (without repetition) and set $|\mathbf{u}| = k$. For $\boldsymbol{x} = (x_1,\dots,x_d) \in [0,1]^d$, we denote by $\boldsymbol{x}_{\mathbf{u}} = (x_{u_1},\dots,x_{u_k}) \in [0,1]^k$ and $\boldsymbol{z} = (\boldsymbol{x}_{\mathbf{u}}, 1)$ the vector with components $z_j = x_j$ if $j \in \mathbf{u}$ and $z_j = 1$ if $j \notin \mathbf{u}$. With this notation at hand, the Zaremba's identity generalizes to the multi-dimensional case as follows.

Lemma 7.2 (**Hlawka's identity**). *Let $\psi : [0,1]^d \to \mathbb{R}$ be an integrable function with integrable mixed first order derivatives of all order, and let $\mathcal{P} = \{\boldsymbol{X}^{(1)},\dots,\boldsymbol{X}^{(N)}\}$ be an arbitrary point set in $[0,1]^d$. Then*

$$\frac{1}{N}\sum_{i=1}^N \psi(\boldsymbol{X}^{(i)}) - \int_{[0,1]^d} \psi(\boldsymbol{x})\,d\boldsymbol{x} = \sum_{\mathbf{u}\subset\{1,\dots,d\}} (-1)^{|\mathbf{u}|}\int_{[0,1]^{|\mathbf{u}|}} \frac{\partial^{|\mathbf{u}|}\psi}{\partial \boldsymbol{x}_{\mathbf{u}}}(\boldsymbol{x}_{\mathbf{u}},1)\Delta_{\mathcal{P}}(\boldsymbol{x}_{\mathbf{u}},1)\,d\boldsymbol{x}_{\mathbf{u}}$$

where $\frac{\partial^{|\mathbf{u}|}\psi}{\partial \boldsymbol{x}_{\mathbf{u}}} = \frac{\partial^k \psi}{\partial x_{u_1}\dots\partial x_{u_k}}$ *is a* mixed *first order derivative.*

Proof. By induction on d, one can prove the following identity

$$\psi(\boldsymbol{x}) = \sum_{\mathbf{u}\subset\{1,\dots,d\}} (-1)^{|\mathbf{u}|}\int_{[\boldsymbol{x}_{\mathbf{u}},\mathbf{1}]} \frac{\partial^{|\mathbf{u}|}\psi}{\partial \boldsymbol{x}_{\mathbf{u}}}(\boldsymbol{y}_{\mathbf{u}},1)\,d\boldsymbol{y}_{\mathbf{u}}, \qquad \forall \boldsymbol{x} \in [0,1]^d, \quad (7.3)$$

which generalizes the $d=1$ identity $\psi(x_1)=\psi(1)-\int_{x_1}^1 \frac{\partial\psi}{\partial x_1}(y)\,dy$ already used in the proof of Lemma 7.1. In (7.3) we have used the convention that for $\mathbf{u}=\emptyset$, $(-1)^{|\mathbf{u}|}\int_{[\boldsymbol{x}_\mathbf{u},\mathbf{1}]}\frac{\partial^{|\mathbf{u}|}}{\partial\boldsymbol{x}_\mathbf{u}}\psi(\boldsymbol{y}_\mathbf{u},1)\,d\boldsymbol{y}_\mathbf{u}=\psi(1,\ldots,1)$. Then

$$
\begin{aligned}
&\frac{1}{N}\sum_{i=1}^N\psi(\boldsymbol{X}^{(i)})-\int_{[0,1]^d}\psi(\boldsymbol{x})\,d\boldsymbol{x}\\
&\quad=\sum_{\mathbf{u}\subset\{1,\ldots,d\}}(-1)^{|u|}\left(\frac{1}{N}\sum_{i=1}^N\underbrace{\int_{[\boldsymbol{X}_\mathbf{u}^{(i)},1]}\frac{\partial^{|\mathbf{u}|}}{\partial\boldsymbol{x}_\mathbf{u}}\psi(\boldsymbol{y}_\mathbf{u},1)\,d\boldsymbol{y}_\mathbf{u}}_{=\int_{[0,1]^{|\mathbf{u}|}}\frac{\partial^{|\mathbf{u}|}}{\partial\boldsymbol{x}_\mathbf{u}}\psi(\boldsymbol{y}_\mathbf{u},1)\mathbb{1}_{[0,\boldsymbol{y}_\mathbf{u}]}(\boldsymbol{X}_\mathbf{u}^{(i)})\,d\boldsymbol{y}_\mathbf{u}}-\underbrace{\int_{[0,1]^d}\int_{[\boldsymbol{x}_\mathbf{u},1]}\frac{\partial^{|\mathbf{u}|}}{\partial\boldsymbol{x}_\mathbf{u}}\psi(\boldsymbol{y}_\mathbf{u},1)\,d\boldsymbol{y}_\mathbf{u}\,d\boldsymbol{x}}_{=\int_{[0,1]^{|\mathbf{u}|}}\int_{[0,\boldsymbol{y}_\mathbf{u}]}\frac{\partial^{|\mathbf{u}|}}{\partial\boldsymbol{x}_\mathbf{u}}\psi(\boldsymbol{y}_\mathbf{u},1)\,d\boldsymbol{x}_\mathbf{u}\,d\boldsymbol{y}_\mathbf{u}}\right)\\
&\quad=\sum_{\mathbf{u}\subset\{1,\ldots,d\}}(-1)^{|\mathbf{u}|}\int_{[0,1]^{|u|}}\frac{\partial^{|\mathbf{u}|}}{\partial\boldsymbol{x}_\mathbf{u}}\psi(\boldsymbol{y}_\mathbf{u},1)\underbrace{\left(\frac{1}{N}\sum_{i=1}^N\mathbb{1}_{[0,\boldsymbol{y}_\mathbf{u}]}(\boldsymbol{X}_\mathbf{u}^{(i)})-\mathrm{Vol}([0,\boldsymbol{y}_\mathbf{u}])\right)}_{\Delta_\mathcal{P}(\boldsymbol{y}_\mathbf{u},1)}
\end{aligned}
$$

□

From the Hlawka's identity, the multidimensional version of the Koksma-Hlawka inequality follows. Let us define the following norm

$$
\|\psi\|_{p,p'}=\left(\sum_{\mathbf{u}\subset\{1,\ldots,d\}}\left(\int_{[0,1]^{|\mathbf{u}|}}\left|\frac{\partial^{|\mathbf{u}|}}{\partial\boldsymbol{x}_\mathbf{u}}\psi(\boldsymbol{y}_\mathbf{u},1)\right|^p d\boldsymbol{y}_\mathbf{u}\right)^{p'/p}\right)^{1/p'}.
$$

Then, the multidimensional **Koksma-Hlawka inequality** reads

$$
\left|\int_{[0,1]^d}\psi(\boldsymbol{x})\,d\boldsymbol{x}-\frac{1}{N}\sum_{i=1}^N\psi(\boldsymbol{X}^{(i)})\right|\le\|\psi\|_{p,p'}\|\Delta_\mathcal{P}\|_{q,q'},\qquad\text{with }\frac{1}{p}+\frac{1}{q}=\frac{1}{p'}+\frac{1}{q'}=1,\tag{7.4}
$$

provided $\|\psi\|_{p,p'}<+\infty$. In particular, if $\|\psi\|_{1,1}<+\infty$, then

$$
\left|\int_{[0,1]^d}\psi(\boldsymbol{x})\,d\boldsymbol{x}-\frac{1}{N}\sum_{i=1}^N\psi(\boldsymbol{X}^{(i)})\right|\le\|\psi\|_{1,1}D_N^*(\mathcal{P}).
$$

Again, this inequality shows that the quadrature error is proportional to the star-discrepancy $D_N^*(\mathcal{P})$ provided ψ has *integrable mixed first order derivatives.*

7.1 Low discrepancy sequences and point sets

There exist constructions of point sets $\mathcal{P}=\{\boldsymbol{X}^{(1)},\ldots,\boldsymbol{X}^{(N)}\}\subset[0,1]^d$ that have star-discrepancy as low as $D_N^*(\mathcal{P})=O\left(\frac{(\log N)^{d-1}}{N}\right)$. It is widely believed that this result is sharp, i.e. there do not exist points sets that achieve a better bound. In general, these constructions do not lead to a nested sequence of points, that is, the point set $\mathcal{P}^1=\{\boldsymbol{X}^{(1)},\ldots,\boldsymbol{X}^{(M)}\}$ with $M>N$ does not contain $\mathcal{P}$, in general.

For the nested point sets, i.e. point sets $\mathcal{P} = \{\boldsymbol{X}^{(1)}, \dots, \boldsymbol{X}^{(N)}\}$ that are generated as the first N points of an infinite sequence $\mathcal{S} = \{\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, \dots\}$ the lowest achievable star-discrepancy is slightly worse, namely $D_N^*(P) = O\left(\frac{(\log N)^d}{N}\right)$. In view of these results, we give the following definition.

Definition 7.2. (low discrepancy sets).

- A family $\mathcal{P} = \{\mathcal{P}_N\}_{N\in\mathbb{N}}$ of non-nested point sets $\mathcal{P}_N = \{\boldsymbol{X}^{(1)}, \dots, \boldsymbol{X}^{(N)}\} \subset [0,1]^d$ is called a *low discrepancy family of point sets* if $D_N^*(\mathcal{P}) = O\left(\frac{(\log N)^{d-1}}{N}\right)$;
- A point sequence $\mathcal{S} = \{\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, \dots\} \subset [0,1]^d$ is called a *low discrepancy sequence* if $D_N^*(\mathcal{S}) = O\left(\frac{(\log N)^d}{N}\right)$.

From the above definitions and considerations, we see that a QMC quadrature formula can achieve convergence rate $1/N$ up to logarithmic terms (which however grow exponentially in the dimension!), provided the integrand function has integrable mixed first derivatives. Before presenting some common low discrepancy sequences/points sets, we give two important clarifying examples:

Example 7.1. *Consider the family $\mathcal{P} = \{\mathcal{P}_N\}_{N\in\mathbb{N}}$ of point sets $P_N = \{\boldsymbol{X}^{(1)}, \dots, \boldsymbol{X}^{(N)}\}$ with $\boldsymbol{X}^{(i)} \overset{iid}{\sim} \mathcal{U}([0,1]^d)$, i.e.* a random *iid sample. Then, $|\Delta_p(\boldsymbol{y})| = |\widehat{\mathrm{Vol}}([0,\boldsymbol{y}]) - \mathrm{Vol}([0,\boldsymbol{y}])|$ is the error of the sample average estimator of $\mathrm{Vol}([0,\boldsymbol{y}])$, which decays as $O\left(\frac{1}{\sqrt{N}}\right)$, in a probabilistic sense. We conclude that the family of random iid point sets has* not *low discrepancy.*

Example 7.2. *Consider the family $\mathcal{P} = \{\mathcal{P}_N\}_{N\in\mathbb{N}}$ of point sets given by regular lattices (see figure 7.2)*

$$\mathcal{P}_N = \left\{\left(\frac{k_1 + 1/2}{m}, \dots, \frac{k_d + 1/2}{m}\right), 0 \le k_j \le m-1, j = 1, \dots, d\right\}, \quad N = m^d$$

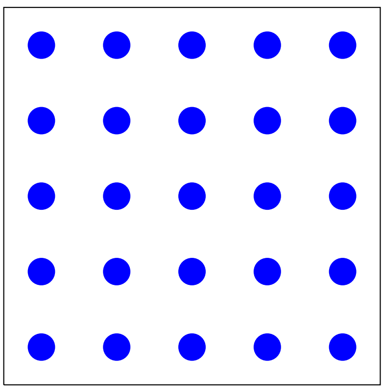


Figure 7.2: Regular lattice.

For $d = 1$, it is easy to see that $D_N^*(\mathcal{P}) = \frac{1}{2m} = \frac{1}{2N}$, hence $\mathcal{P}$ has low discrepancy. On the other hand, in dimension $d > 1$ we have

$$D_N^*(\mathcal{P}) = \sup_{\boldsymbol{y} \in [0,1]^d} |\Delta_{\mathcal{P}}(\boldsymbol{y})| \geq \sup_{t \in [0,1]} |\Delta_{\mathcal{P}}(t, 1, \ldots, 1)| = \frac{1}{2m} = \frac{1}{2N^{1/d}}.$$

We conclude then that the family of regular lattices has not low discrepancy in higher dimension.

Van der Corput-Halton sequence

Let $b \geq 2$ be an integer. Any natural number $n \in \mathbb{N}_0$ can be expanded in a b-adic expansion $n = n_0 + n_1 b + n_2 b^2 + \ldots$. The radical inverse of n is defined as

$$\varphi_b(n) = \frac{n_0}{b} + \frac{n_1}{b} + \ldots.$$

Obviously $\varphi_b : \mathbb{N}_0 \to [0, 1)$. In 1D, the b-adic Van der Corput sequence is

$$\varphi_b(0), \varphi_b(1), \varphi_b(2), \ldots$$

For example, for $b = 2$, the Van der Corput sequence is $0, \frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{5}{8}, \frac{3}{8}, \frac{7}{8}, \ldots$.

The *Halton* sequence generalizes this construction for $d \geq 2$: Let $b_1, \ldots, b_d \geq 2$ be integers pairwise relatively prime. Typically $b_1, \ldots, b_d$ are taken as the first d prime numbers. Then, the Halton sequence is

$$\mathcal{S} = \{\boldsymbol{X}^{(n)}, n \in \mathbb{N}_0\}, \qquad \boldsymbol{X}^{(i)} = (\varphi_{b_1}(n), \varphi_{b_2}(n), \ldots, \varphi_{b_d}(n))$$

and achieves the optimal bound on the star-discrepancy $D_N^*(\mathcal{S}) \leq c(d) \frac{(\log N)^d}{N}$.

Hammersley point set

It is derived from the Halton sequence by taking $\mathcal{P}_N = \{\boldsymbol{X}^{(0)}, \ldots, \boldsymbol{X}^{(N-1)}\}$ with $\boldsymbol{X}^{(n)} = \left(\frac{n}{N}, \varphi_{b_1}(n), \ldots, \varphi_{b_{d-1}}(n)\right)$. The family $\mathcal{P} = \{\mathcal{P}_N\}$ of Hammersley point sets is non-nested and achieves the better bound $D_N^*(\mathcal{P}) = c(d) \frac{(\log N)^{d-1}}{N}$.

Lattice point sets

Let $N \in \mathbb{N}$ and $\boldsymbol{g} \in \mathbb{N}^d$, $\boldsymbol{g} = (g_1, \ldots, g_d)$ such that g_j has no factor in common with N. (Typically N is taken as a prime number.) Then the N-lattice point set with generating vector $\boldsymbol{g}$ is defined as

$$\mathcal{P}_N = \left\{ \frac{n\boldsymbol{g}}{N} \right\}_{n=0}^{N-1}$$

where $\{\cdot\}$ denotes the fractional part. Figure 7.3 shows an example of lattice point set. Good choices of $\boldsymbol{g}$ lead to low discrepancy non-nested point sets.

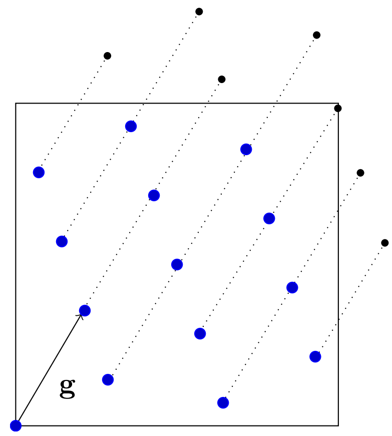


Figure 7.3: Lattice point set with $N = 14$ and $\mathbf{g} = (3, 5)$

(t-m-d)-nets and (t-d) sequences

Let $0 \le t \le m \in \mathbb{N}$ and $b \ge 2$. A $(t\text{-}m\text{-}d)$-net in base b is a point set $\mathcal{P}_N$ consisting of $N = b^m$ points such that each elementary rectangle of volume b^{t-m},

$$R_a = \prod_{j=1}^{d} \left[\frac{a_j - 1}{b^{p_j}}, \frac{a_j}{b^{p_j}} \right), \qquad a_j = 1, \ldots, b^{p_j}$$

with $p_1 + p_2 + \ldots + p_d = m - t$ contains exactly b^t points. E.g. if $t = 0$, each elementary rectangle of volume b^{-m} contains exactly 1 point.

Example 7.3. *A (0-3-2)-net in base $b = 2$ is a point set with $N = 2^3 = 9$ points, such that each elementary rectangle with volume $2^{-(m-t)} = 2^{-3} = 1/8$ contains exactly $2^t = 1$ point. Figure 7.4 shows graphically this property.*

A $(t\text{-}d)$ sequence in base b is a sequence $\mathcal{S} = \{\boldsymbol{X}_0, \boldsymbol{X}_1, \ldots\}$ such that for any $m > t$, every block of b^m points $\{\boldsymbol{X}^{(\ell b^n)}, \ldots, \boldsymbol{X}^{((\ell+1)b^m - 1)}\}$, $\ell \in \mathbb{N}$ is a $(t\text{-}m\text{-}d)$-net in base b. The star-discrepancy of a $(t\text{-}m\text{-}d)$-net satisfies $D_N^*(\mathcal{P}) = O\left(b^t \frac{(\log N)^{d-1}}{N}\right)$ and similarly for a $(t\text{-}d)$-sequence $D_N^*(\mathcal{S}) = O\left(b^t \frac{(\log N)^d}{N}\right)$. Famous $(t\text{-}d)$-sequences are those of Sobol, Niederreiter and Faure. For a description of their construcion we refer to [2].

7.2 Randomized QMC formulas

Let us consider a point set $\mathcal{P}_N = \{\boldsymbol{X}^{(1)}, \ldots, \boldsymbol{X}^{(N)}\}$ and the QMC quadrature formula

$$\hat{\mu}_{\text{QMC}} = \frac{1}{N} \sum_{i=1}^{N} \psi(\boldsymbol{X}^{(i)}).$$

The question is how to estimate the error $|\mu - \hat{\mu}_{\text{QMC}}|$. Since the points $\boldsymbol{X}^{(i)}$ are not random iid, we can not use a variance estimator or a CLT as in the Monte Carlo estimator. On

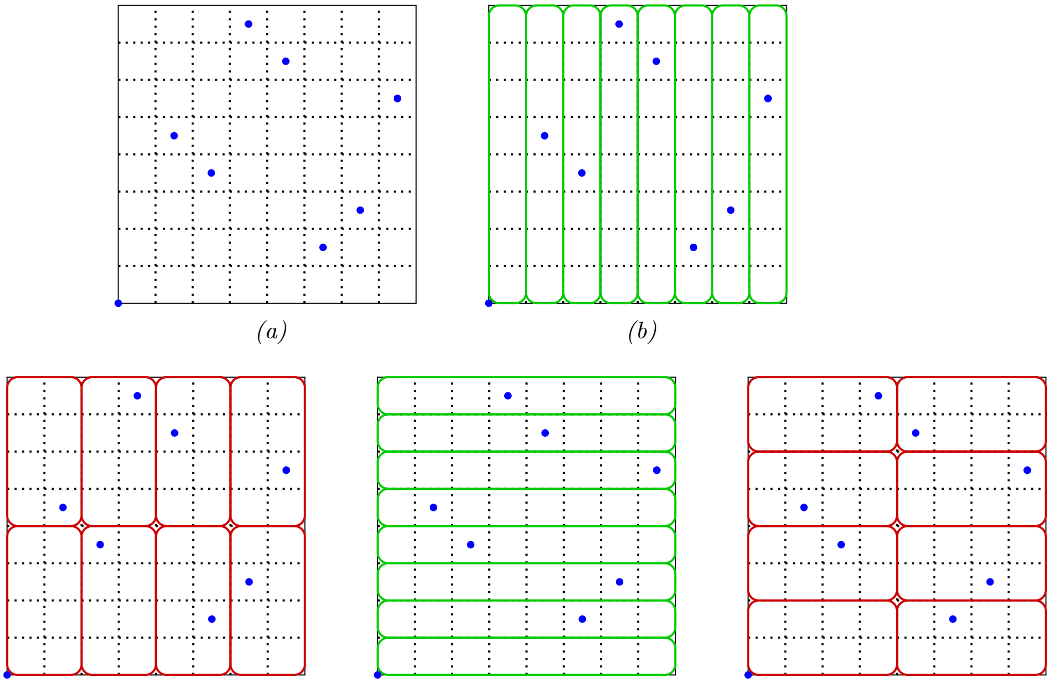


Figure 7.4: Example in base 2.

the other hand, the error estimates in (7.4) can not be really used in practice to provide a bound on the quadrature error as they involve quantities such as the discrepancy or TV-norm of the integrad that are not known and can not be easily estimated.

An easy idea to obtain error bounds is to randomize the QMC formula. Let $\boldsymbol{U} \sim \mathcal{U}([0,1]^d)$. If $\mathcal{P} = \{\boldsymbol{X}^{(1)}, \dots, \boldsymbol{X}^{(N)}\}$ is a low discrepancy point set, so is

$$P_U = \{\{\boldsymbol{X}^{(1)} + \boldsymbol{U}\}, \{\boldsymbol{X}^{(2)} + \boldsymbol{U}\}, \dots, \{\boldsymbol{X}^{(N)} + \boldsymbol{U}\}\}$$

where the same shift is applied to all points and again $\{\cdot\}$ denotes the fractional part. P_U is called a randomly shifted point set. We could then compute $\hat{\mu}^{(j)}_{\mathrm{QMC}}$, $j = 1, \dots, k$, for few randomly shifted point sets and average the obtained results. The resulting randomly shifted QMC estimator is then $\hat{\mu}_{QMC} = \frac{1}{k}\sum_{j=1}^{k} \hat{\mu}^{(j)}_{\mathrm{QMC}}$. Since $\boldsymbol{U}^{(j)} \sim \mathcal{U}([0,1]^d)$, so is $\{\boldsymbol{X}^{(i)} + \boldsymbol{U}^{(j)}\}$ for any $i = 1, \dots, N$. It follows that $\hat{\mu}_{\mathrm{QMC}}$ is an *unbiased* estimator of $\mu = \mathbb{E}[\psi]$. Moreover, since $\hat{\mu}^{(j)}_{\mathrm{QMC}}$ are independent, the variance of the estimator is $\mathbb{V}\mathrm{ar}\left(\hat{\mu}_{\mathrm{QMC}}\right) = \frac{\sigma^2_{\mathrm{QMC}}}{k}$ with $\sigma^2_{\mathrm{QMC}} = \mathbb{E}\left[(\hat{\mu}^{(j)}_{\mathrm{QMC}} - \mu)^2\right] = O\left(\frac{(\log N)^{2(d-1)}}{N^2}\right)$ hence, very small, in general, and can be estimated by the standard sample variance estimator $\hat{\sigma}^2_{\mathrm{QMC}} = \frac{1}{k-1}\sum_{j=1}^{k}(\hat{\mu}^{(j)}_{\mathrm{QMC}} - \hat{\mu}_{\mathrm{QMC}})^2$.

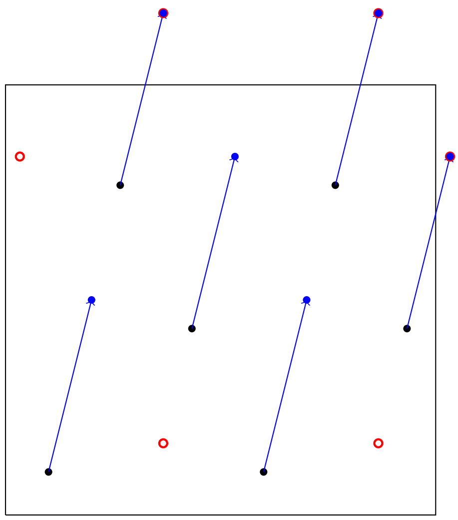


Figure 7.5: Randomized QMC.

Algorithm 7.1: Randomly shifted QMC.

1 Generate $\boldsymbol{U}^{(1)}, \ldots, \boldsymbol{U}^{(k)} \overset{\text{iid}}{\sim} \mathcal{U}([0,1]^d)$;

2 For $j = 1, \ldots, k$, compute $\hat{\mu}^{(j)}_{\text{QMC}} = \frac{1}{N} \sum_{i=1}^{N} \psi(\{\boldsymbol{X}^{(i)} + \boldsymbol{U}^{(j)}\})$;

3 Compute $\hat{\mu}_{\text{QMC}} = \frac{1}{k} \sum_{j=1}^{k} \hat{\mu}^{(j)}_{\text{QMC}}$ as well as $\hat{\sigma}^2_{QMC} = \frac{1}{k-1} \sum_{j=1}^{k} (\hat{\mu}^{(j)}_{QMC} - \hat{\mu}_{QMC})^2$;

4 Output $\hat{\mu}_{QMC}$ as well as a $1 - \alpha$ confidence

$$I_\alpha = [\hat{\mu}_{\text{QMC}} - c_{1-\alpha/2} \frac{\hat{\sigma}_{\text{QMC}}}{\sqrt{k}},\ \hat{\mu}_{\text{QMC}} + c_{1-\alpha/2} \frac{\hat{\sigma}_{\text{QMC}}}{\sqrt{k}}$$

.

Chapter 8

Markov Chain Monte Carlo

Let π be a given probability density function a state space $\mathcal{X} \subset \mathbb{R}^n$ and $\varphi : \mathcal{X} \to \mathbb{R}$ an integrable function with respect to π. We consider the goal of computuing $\mu = \mathbb{E}_\pi[\varphi] = \int_{\mathcal{X}} \varphi(x)\, \pi(x) dx$.

If we can generate independent replicas of $Z \sim \pi$, then μ can be computed by Monte Carlo or any improved version using variance reduction techniques.

Assume, however, that sampling directly from π is not viable either because the the expression of π is too complicated and possibly high dimensional, or because π is known only up to a multiplicative constnant and computing the normalization constant might be too expensive, if not impossible.

Example 8.1 (Bayesian statistics)**.** *Let $\boldsymbol{X} = (X_1, \ldots, X_n)$ be an iid sample from a parametric density $g(\boldsymbol{x} \mid \theta)$. Then the joint density of $\boldsymbol{X}$ given θ is $g(\boldsymbol{X} \mid \theta) = \prod_{i=1}^n g(X_i \mid \theta)$ and we want to estimate θ from the sample $\boldsymbol{X}$. In the Bayesian paradigm, θ is thought as a random variable itself, with prior density $\pi_0(\theta)$, which summarizes any prior information on θ in the absence of data. Then, the* posterior *density of θ given the data is*

$$\pi(\theta) = \frac{1}{Z(\boldsymbol{X})} g(\boldsymbol{X} \mid \theta)\pi_0(\theta)$$

with $Z(\boldsymbol{X}) = \int g(\boldsymbol{X} \mid \theta)\pi_0(\theta)\, d\theta$ which is often unknown and difficult to compute.

Example 8.2 (Statistical physics)**.** *Let $x \in \mathcal{X}$ be a configuration of a physical system and $\mathcal{X}$ the configuration space. Let $H : \mathcal{X} \to \mathbb{R}$ be an energy function and T the temperature. Then the probabilily density function of finding the system in a given state x is*

$$\pi(x) = \frac{1}{Z} \exp\left\{ -\frac{H(x)}{kT} \right\}$$

where k is the Boltzmann constant and $Z = \int e^{-H(x)/kT}\, dx$ is the partition function, often difficult to compute.

The idea of Markov chain Monte Carlo (MCMC) is to construct an ergodic Markov Chain $\{X_n\}_n \sim \text{Markov}\,(\lambda, P)$ on $\mathcal{X}$ that has π as its invariant distribution. Then we can approximate $\mu = \mathbb{E}_\pi[\varphi]$ by the ergodic estimator

$$\hat{\mu}_N^{\mathrm{MCMC}} = \frac{1}{N}\sum_{i=1}^{N} \varphi(X_i)$$

or

$$\hat{\mu}_{N,N_0}^{\mathrm{MCMC}} = \frac{1}{N}\sum_{i=1}^{N} \varphi(X_{i+N_0})$$

if we want to "cut" out the first part of the chain, which might be to sensitive to the initial state $X_0 \sim \lambda$ of the chain (this operation is usually called "burn-in"). We will see that constructing a Marcov Chain with a given invariant distribution is not so difficult and can be achieved by the well known and celebrated Metropolis-Hastings algorithm. Before discussing such algorithm, however, it is worth recalling some basic concepts in the theory of Markov Chains. We will do so in the finite state space case in the next section and briefly mention generalizations to general state spaces in Section 8.2.

8.1 Markov Chains on finite state space (review)

Let $\mathcal{X} = \{x_1, x_2, \ldots, x_d\}$ be a finite state space, $\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_d\}$ a probability mass function on $\mathcal{X}$, with $\lambda_i \geq 0$ for all $i = 1, \ldots, d$, $\sum_i \lambda_i = 1$, and $P = \{P_{ij}\}_{i,j=1}^d$ a stochastic matrix, such that $P_{ij} \geq 0$ for all $i, j = 1, \ldots, d$ and $\sum_j P_{ij} = 1$ for all $i = 1, \ldots, d$.

We consider hereafter a homogeneous Markov chain $\{X_n,\ n \in \mathbb{N}_0\} \sim \mathrm{Markov}\,(\lambda, P)$ having initial state $\boldsymbol{X}_0 \sim \lambda$ and transition matrix P (See Chapter 4 for the definition of a Markov Chain). To highlight the dependence of the chain on the initial distribution λ, we denote by $\mathbb{P}_\lambda(A)$ the probability of an event A under $X_0 \sim \lambda$. If $\lambda = \delta_{x_i}$ (i.e. $\mathbb{P}\,(X_0 = x_i) = 1$), we use the notation $\mathbb{P}_{x_i}$ or simply $\mathbb{P}_i$. We introduce also the following notion

Definition 8.1 (Stopping time)**.** *A random variable τ is called a stopping time if the event $\{\tau \leq n\}$ depends only on $X_0, \ldots, X_n$, i.e. the event $\{\tau \leq n\}$ is measurable with respect to the σ-algebra $\sigma(X_0, \ldots, X_n)$ generated by $X_0, \ldots, X_n$.*

A typical example of a stopping time is the hitting time of a subset $A \subset \mathcal{X}$: $\tau_A = \inf\{n \geq 0 :\ X_n \in A\}$ with the convention that $\tau_A = +\infty$ if $X_n \notin A$ for any n. From the definition of a homogeneous Markov chain $\mathrm{Markov}\,(\lambda, P)$, the following *Markov proprety* follow.

Lemma 8.1. *Let $\{X_n,\ n \in \mathbb{N}_0\}$ be* Markov (λ, P)*.*

- (Weak Markov Property)*. Conditional on $X_m = x_i$, $\{X_{m+n},\ n \in \mathbb{N}_0\}$ is* Markov (δ_{x_i}, P) *and independent of $\{X_0, \ldots, X_m\}$.*

- (Strong Markov property)*. Let τ be a stopping time of $\{X_n\}$. Conditional on $\tau < +\infty$ and $X_\tau = x_i$, $\{X_{\tau+n},\ n \in \mathbb{N}_0\}$ is* Markov (δ_{x_i}, P) *independent of $X_0, \ldots, X_\tau$.*

Given $\{X_n, n \in \mathbb{N}_0\} \sim \text{Markov}\,(\lambda, P)$, let $P^{(n)}$ be the *n-step transition matrix*, i.e. $P^{(n)}_{ij} = \mathbb{P}\,(X_{m+n} = x_j \mid X_m = x_i)$. Thanks to the Markov property, $P^{(n)}_{ij}$ does not depend on m. Clearly $P^{(1)} = P$ and for $n > 1$,

$$\begin{aligned} P^{(n)}_{ij} &= \sum_\ell \mathbb{P}\,(X_{m+n} = x_j \mid X_{m+n-1} = x_\ell, x_m = x_i)\,\mathbb{P}\,(X_{m+n-1} = x_\ell \mid X_m = x_i) \\ &= \sum_\ell P_{\ell j} P^{(n-1)}_{i\ell}. \end{aligned}$$

Introducing the matrix multiplication $(P^2)_{ij} = \sum_\ell P_{i\ell} P_{\ell j}$, we see that $P^{(n)} = P^n$. More generally, $P^{(n+m)} = P^n P^m$ which is often referred to as the *Chapman Kolmogorov equation.*

We may also ask what is the probability distribution of X_n at any given $n > 0$, i.e. the probability mass function $\pi^{n,\lambda} = (\pi^{n,\lambda}_1, \ldots, \pi^{n,\lambda}_1)$, with $\pi^{n,\lambda}_i = \mathbb{P}_\lambda(X_n = x_i)$. It is easy to see that

$$\pi^{n,\lambda}_i = \sum_\ell \mathbb{P}\,(X_n = x_i \mid X_{n-1} = x_\ell)\,\mathbb{P}\,(X_{n-1} = x_\ell) = \sum_\ell P_{\ell i} \pi^{n-1,\lambda}_\ell.$$

In matrix notation,

$$\pi^{n,\lambda} = \pi^{n-1,\lambda} P = \lambda P^n.$$

If we denote by $M_1(\mathcal{X}) = \{(\mu_1, \ldots, \mu_d) \in \mathbb{R}^d : \ \mu_i \geq 0, \ \sum_i \mu_i = 1\}$ the set of probability mass functions on $\mathcal{X}$, then the transition matrix P can be interpreted as an operator $P : M_1(\mathcal{X}) \to M_1(\mathcal{X})$ acting (to the left) on probability measures. We may ask if such an operator has a fixed point.

Definition 8.2. *A probability mass function $\pi \in M_1(\mathcal{X})$ is called* invariant distribution *for P if $\pi P = \pi$.*

Hence, for a Markov chain $\{X_n\} \sim \text{Markov}\,(\lambda, P)$ whose initial state $X_0 \sim \pi$ is distributed as the invariant distribution π, it follows that $X_n \sim \pi$ for any n and the chain is said to be "at equilibrium" or "at stationarity". Observe that if an invariant distribution π exists, then it is a *left* eigenvector of the transition matrix P, associated to the eigenvalue $\lambda_1 = 1$.

Consider now the set $\mathcal{F}(\mathcal{X}) = \{\varphi : \mathcal{X} \to \mathbb{R}\}$ of measurable functions on $\mathcal{X}$, which can be identified with $\mathbb{R}^d$. We represent any function $\varphi \in \mathcal{F}(\mathcal{X})$ as a column vector $\varphi = (\varphi_1, \ldots, \varphi_d)^\top \in \mathbb{R}^d$. Given $\varphi \in \mathcal{F}(\mathcal{X})$, we can define the following function $g \in \mathcal{F}(\mathcal{X})$:

$$g_i = \mathbb{E}\,[\varphi(X_{n+1}) \mid X_n = x_i] = \mathbb{E}_{x_i}[\varphi(X_1)], \quad i = 1, \ldots, d,$$

the last equality being justified thanks to the Markov property. Clearly we have $g_i = \sum_{j=1}^d \varphi(x_j)\mathbb{P}\,(X_{n+1} = x_j \mid X_n = x_i) = \sum_j \varphi_j P_{ij}$ which, in matrix notation gives

$$g = P\varphi.$$

Hence, the transition matrix P can also be interpreted as an operator $P : \mathcal{F}(\mathcal{X}) \to \mathcal{F}(\mathcal{X})$ acting (to the right) on functions. Observe, in particular, that the constant unit function

$\varphi = (1, \dots, 1) \in \mathcal{F}(\mathcal{X})$ satisfies

$$(P\varphi)_i = \sum_j 1 \cdot P_{ij} = 1 = \varphi_i,$$

since P is a stochastic matrix, and is therefore a *right eigenvector* of P corresponding to the eigenvalue $\lambda_1 = 1$. This argument shows that $\lambda = 1$ is *always* an eigenvalue of P and, at least in the finite dimensional case considered here, an invariant distribution (corresponding left eigenvector) always exists. The existence of an invariant measure is a more delicate matter in infinite dimensional state spaces.

The eigenvalue $\lambda_1 = 1$ turns out to be the largest one in absolute value.

Lemma 8.2. *Given a stochastic matrix $P \in \mathbb{R}^{d\times d}$, all eigenvalues $\lambda_i(P)$ satisfy $|\lambda_i(P)| \leq 1, \quad i = 1, \dots, d$.*

Proof. Let λ be an eigenvalue of P and v a left eigenvector associated to it, so that $vP = \lambda v$. Then

$$|\lambda v_i| = |\sum_j v_j P_{ji}| \leq \sum_j |v_j| P_{ji}.$$

Hence

$$|\lambda| \sum_i |v_i| \leq \sum_i \sum_j |v_j| P_{ji} = \sum_j |v_j| \underbrace{\sum_i P_{ji}}_{=1} = \sum_j |v_j|$$

which implies $|\lambda| \leq 1$. □

It follows that the an invariant distribution π is a left eigenvector of P corresponding to the largest (in absolute value) eigenvalue. The iterates $\pi^{n,\lambda} = \lambda P^n$ correspond to power iterations so we should expect $\pi^{n,\lambda}$ to converge to π as long as $\lambda_1 = 1$ is a simple eigenvalue and there are no other eigenvalues with absolute value 1.

In practice, in MCMC algorithms, we construct a Markov Chain so that the target distribution we want to sample from corresponds to an invariant distribution of the Markov Chain. (This also guarantees existence of an invariant distribution in the infinite dimensional case). However, it remains the question whether such invariant distribution is unique ($\lambda_1 = 1$ is simple) and whether the second largest eigenvalue $\beta = \max_{i=2,\dots,d} |\lambda_i(P)|$ in absolute value is strictly smaller than one as the *spectral gap* $1 - \beta$ will dictate the speed of convergence of $\pi^{n,\lambda}$ to π. We postpone this discussion to Section 8.1.2.

We now address the important case in which the transition matrix P features some symmetry properties. This will be indeed the case for the most popular MCMC algorith, namely the Metropolis-Hastings one. Let P be a transition matrix with invariant distribution π, and $\{X_n\}_{n=0}^N \sim \text{Markov}(\pi, P)$ a Markov chain at equilibrium. Let us look at the chain $\{Y_n = X_{N-n},\ n = 0, \dots, N\}$, called the *time-reversal* of $\{X_n\}$. It is not difficult to see that $\{Y_n\}_n$ is also a Markov chain. Indeed, assuming that

$\mathbb{P}\left(X_{N-n+1} = x_{i_{n-1}}, \ldots, X_N = x_0\right) > 0$, we have for any $n = 1, \ldots, N$

$$\begin{aligned}\mathbb{P}\left(Y_n = x_{i_n} \mid Y_0 = x_{i_0}, \ldots, Y_{n-1} = x_{i_{n-1}}\right) & \\ &= \mathbb{P}\left(X_{N-n} = x_{i_n} \mid X_N = x_{i_0}, \ldots, X_{N-n+1} = x_{i_{n-1}}\right) \\ &= \frac{\mathbb{P}\left(X_{N-n} = x_{i_n}, \ldots, X_N = x_{i_0}\right)}{\mathbb{P}\left(X_{N-n+1} = x_{i_{n-1}}, \ldots, X_N = x_{i_0}\right)} \\ &= \frac{P_{i_1 i_0} P_{i_2 i_1} \ldots P_{i_n i_{n-1}} \mathbb{P}\left(X_{N-n} = x_{i_n}\right)}{P_{i_1 i_0} P_{i_2 i_1} \ldots P_{i_{n-1} i_{n-2}} \mathbb{P}\left(X_{N-n+1} = x_{i_{n-1}}\right)} \\ &= P_{i_n i_{n-1}} \frac{\pi_{i_n}}{\pi_{i_{n-1}}} =: \hat{P}_{i_{n-1}, i_n}.\end{aligned}$$

Hence, the probability $\mathbb{P}\left(Y_n = x_{i_n} \mid Y_0 = x_{i_0}, \ldots, Y_{n-1} = x_{i_{n-1}}\right)$ of Y_n given the past depends only on i_{n-1} and $\{Y_n\}_{n=0}^N$ is a Markov chain $\{Y_n\}_{n=0}^N \sim \text{Markov}(\pi, \hat{P})$ with transition matrix

$$\hat{P}_{ij} = P_{ji} \frac{\pi_j}{\pi_i}.$$

Definition 8.3. *A stochastic matrix P and a probability distribution λ are said to be in* detailed balance *if $\lambda_i P_{ij} = \lambda_j P_{ji}$ for all i, j.*

Lemma 8.3. *If (P, λ) are in detailed balance, then λ is invariant for P and the transition matrix of the time-reversal chain satisfies $\hat{P} = P$.*

Proof. From direct calculation

$$(\lambda P)_i = \sum_j \lambda_j P_{ji} = \sum_j \lambda_i P_{ij} = \lambda_i \sum_j P_{ij} = \lambda_i.$$

Hence λ is an invariant distribution. Moreover, since $\lambda = \pi$ is invariant, the detailed balance condition directly implies $\hat{P} = P$. □

Definition 8.4. *Let P be a stochastic matrix, π an invariant distribution of P and $\{X_n\} \sim$ Markov (λ, P) a Markov chain at equilibrium. We say that $\{X_n\}_{n\geq 0}$ is* reversible *if for all $N \geq 1$, $\{X_{N-n}\}_{n=0}^N \sim$ Markov (λ, P).*

Clearly, $\{X_n\}$ is reversible if and only if (π, P) are in detailed balance. The detailed balance is a useful condition to verify that a certain distribution λ is invariant (often easier than verifying $\lambda P = \lambda$). Intuitively, it says that under λ, the probability of going from i to j is the same as the probability of going from j to i.

Another way to interpret the detailed balance equation is the following. Let us define the space $\ell^2_\pi = \{\varphi : \mathcal{X} \to \mathbb{R} : \sum_i \varphi_i^2 \pi_i < +\infty\}$ with inner product $(\varphi, \psi)_\pi = \sum_i \varphi_i \psi_i \pi_i$. Then, the matrix P is symmetric with respect to such an inner product (the corresponding operator $P : \ell^2_\pi \to \ell^2_\pi$ is self adjoint). Indeed,

$$(P\varphi, \psi)_\pi = \sum_i \pi_i (P\varphi)_i \psi_i = \sum_{i,j} \pi_i P_{ij} \varphi_j \psi_i = \sum_{i,j} \pi_j P_{ji} \psi_i \varphi_j = (\varphi, P\psi)_\pi.$$

Hence, if (P, π) are in detailed balance, all eigenvalues of P are real and the matrix is diagonalizable by an ℓ^2_π-orthonormal set of eigenvectors.

8.1.1 Metropolis-Hastings algorithm in finite state space

We come back to the original goal of constructing a Markov chain $\{X_n\} \sim \text{Markov}(\lambda, P)$ on $\mathcal{X}$ which has a given invariant distribution π. We assume $\pi_i > 0$ for all i. The Metropolis-Hastings is probably the most popular algorithm used for this purpose. It constructs a transition matrix P which is in detailed balance with the target invariant distribution π. The idea is the following:

- Take a stochastic matrix Q with the condition that $Q_{ij} = 0 \iff Q_{ji} = 0$. Q is called the *proposal*. In general, Q will not have π as invariant distribution so we have to "correct" it.
- For any $i, j \in \{1, \ldots, d\}$, define the acceptance probability

$$\alpha(i,j) = \min\left\{1, \frac{\pi_j Q_{ji}}{\pi_i Q_{ij}}\right\} \text{ if } Q_{ij} \neq 0, \qquad \alpha(i,j) = 0, \text{ if } Q_{ij} = 0.$$

The Metropolis-Hastings algorithm then reads:

Algorithm 8.1: Metropolis-Hastings

Given: λ (initial distribution), Q (proposal), π (target distribution)

1 Generate $X_0 \sim \lambda$ **for** $n = 0, 1, \ldots,$ **do**
2   Generate candidate new state $\tilde{X}_{n+1} \sim Q_{X_n,:}$
3   Generate $U \sim \mathcal{U}([0,1])$
4   **if** $U \leq \alpha(X_n, \tilde{X}_{n+1})$ **then**
5   set $X_{n+1} = \tilde{X}_{n+1}$ // $\tilde{X}_n$ `accepted with prob.` $\alpha(X_n, \tilde{X}_{n+1})$
6   **else**
7   set $X_{n+1} = X_n$ // $\tilde{X}_n$ `rejected with prob.` $1 - \alpha(X_n, \tilde{X}_{n+1})$
8   **end**
9 **end**

If Q is symmetric, then the acceptance probability simplifies to $\alpha(i,j) = \min\left\{1, \frac{\pi_j}{\pi_i}\right\}$. In this case, step 4 of the algorithm will always accept $\tilde{X}_{n+1}$ if the probability mass of the new state $\pi_{\tilde{X}_{n+1}}$ is higher than the probability mass of the old state π_{X_n}. In case where $\pi_{\tilde{X}_{n+1}} < \pi_{X_n}$, the new state is accepted only with probability $\pi_{\tilde{X}_{n+1}}/\pi_{X_n}$. Hence, if $\pi_{\tilde{X}_{n+1}} \ll \pi_{X_n}$, the new state has a high chance to be rejected.

Lemma 8.4. *Let $\alpha_j^* = \sum_j \alpha(i,j) Q_{ij}$. Then, the transition matrix of the chain produced by the Metropolis-Hastings algorithm is given by*

$$P_{ij} = \alpha(i,j) Q_{ij} + (1 - \alpha_j^*) \delta_{ij}. \tag{8.1}$$

Proof. For $j \neq i$, we have

$$\begin{aligned} P_{ij} &= \mathbb{P}(X_{n+1} = j \mid X_n = i) = \mathbb{P}\left(\tilde{X}_{n+1} = j, X_{n+1} = \tilde{X}_{n+1} \mid X_n = i\right) \\ &= \mathbb{P}\left(X_{n+1} = \tilde{X}_{n+1} \mid \tilde{X}_{n+1} = j, X_n = i\right) \mathbb{P}\left(\tilde{X}_{n+1} = j \mid X_n = i\right) \\ &= \alpha(i,j) Q_{ij}. \end{aligned}$$

On the other hand, if $j = i$,

$$\begin{aligned}
P_{ii} &= \mathbb{P}\left(X_{n+1} = i \mid X_n = i\right) \\
&= \mathbb{P}\left(\tilde{X}_{n+1} = i, X_{n+1} = \tilde{X}_{n+1} \mid X_n = i\right) + \mathbb{P}\left(X_{n+1} \neq \tilde{X}_{n+1} \mid X_n = i\right) \\
&= \alpha(i,i)Q_{ii} + \sum_j \mathbb{P}\left(\tilde{X}_{n+1} = j, X_{n=1} \neq \tilde{X}_{n+1} \mid X_n = i\right) \\
&= \alpha(i,i)Q_{ii} + \sum_j (1 - \alpha(i,j))Q_{ij} \\
&= \alpha(i,i)Q_{ii} + (1 - \alpha_i^*).
\end{aligned}$$

□

The quantity $\alpha_i^* = \sum_j \alpha(i,j)Q_{ij}$ represents the overall probability of rejecting a new state while being in state i. If such rejection probability is very close to 1, with high probability the chain will not move, hence the random variables $\{X_n\}$ will be highly correlated. A very high acceptance probability might not be desirable either. Consider the two possible strategies: a) jump only to neighboring states with high acceptance rate; b) jump to far away states but with lower acceptance rate. It is not obvious which strategy is more effective in decorrelating (mixing) the chain. Rule of thumb says that the average acceptance rate should be around 0.2.

That Algorithm 8.1 produces the right chain, i.e. a chain that has invariant distribution π, is shown in the following Lemma and is a consequence of the fact that the transition matrix P in 8.1 is in detailed balance with π.

Lemma 8.5. *The transition matrix P in 8.1 is in detailed balance with π. Hence, the chain produced by Algorithm 8.1 is reversible and has π as invariant distribution.*

Proof. We have to show that $\pi_i P_{ij} = \pi_j P_{ji}$ for all i, j. This is obviously true for $i = j$. Consider then $i \neq j$. If $\pi_i P_{ij} = 0$, then $P_{ij} = 0$ which implies $Q_{ij} = Q_{ji} = 0$ so $P_{ji} = 0$ and $\pi_i P_{ij} = \pi_j P_{ji}$. If $\pi_i P_{ij} \neq 0$, then $P_{ij} \neq 0$ so $Q_{ij} \neq 0$, $Q_{ji} \neq 0$ and

$$\begin{aligned}
\pi_i P_{ij} = \pi_i \alpha(i,j) Q_{ij} &= \pi_i Q_{ij} \min\left\{1, \frac{\pi_j Q_{ji}}{\pi_i Q_{ij}}\right\} \\
&= \min\left\{\pi_i Q_{ij}, \pi_j Q_{ji}\right\} \\
&= \min\left\{\frac{\pi_i Q_{ij}}{\pi_j Q_{ji}}, 1\right\} \pi_j Q_{ji} = \pi_j \alpha(j,i) Q_{ji} = \pi_j P_{ji}.
\end{aligned}$$

□

8.1.2 Convergence to the invariant distribution

Let $\{X_n\} \sim \text{Markov}(\lambda, P)$ be a Markov chain with invariant distribution π. We want to understand under which conditions π is the *unique* invariant distribution and the sequence $\pi^{n,\lambda}$ converges to π as $n \to \infty$. Two concepts are key to answer this question: *irreducibility* and *aperiodicity*.

Definition 8.5 (Irreducible chain)**.** *Let P is a transition matrix on $\mathcal{X}$. We say that a state $x_i \in \mathcal{X}$ communicates with another state $x_j \in \mathcal{X}$ if $\mathbb{P}(X_n = x_j$ for some $n \mid X_0 = x_i) > 0$. Equivalently, there exists $n > 0$: $P_{ij}^{(n)} > 0$.*

The transition matrix P is said to be irreducible if every state communicates with every other state, i.e. for all i, j, there exists $n > 0$ such that $P_{ij}^{(n)} > 0$. A Markov chain Markov (λ, P) *is irreducible if P is so.*

Figure 8.1 shows an example of an irreducible chain (left) and a reducible one (right).

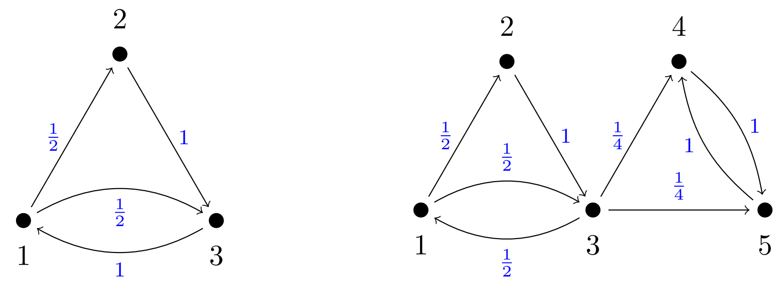


Figure 8.1: Left: irreducible chain – every state communicates with every other state. Right: reducible chain – $\{4, 5\}$ is an absorbing class and does not communicate with $\{1, 2, 3\}$.

Lemma 8.6. *Let P be an irreducible transition matrix and π an invariant distribution for P. Then $\pi_i > 0$ for all i and π is the* unique *invariant distribution of P (i.e. the eigenspace associated to $\lambda_1 = 1$ has dimension 1).*

Proof. We have $\pi_i \geq 0$ for all i and $\pi \neq 0$ so there is at least one element $\pi_j > 0$. Since P is irreducible, for any $i = 1, \dots, d$, there exists $n > 0$ such that $P_{ji}^{(n)} > 0$. Then

$$\pi_i = (\pi P^n)_i = \sum_k \pi_k P_{ki}^{(n)} = \underbrace{\pi_j P_{ji}^{(n)}}_{>0} + \underbrace{\sum_{k \neq j} \pi_k P_{ki}^{(n)}}_{\geq 0} > 0.$$

Suppose now that there exists another eigenvector w associated to $\lambda_1 = 1$, such that $\lambda w = wP$, which we can assume real (since P_1 and λ_1 are real). Let us assume that there is at least one i such that $w_i > 0$ (otherwise take $-w$) and take $j = \operatorname{argmax}_\ell \frac{w_\ell}{\pi_\ell}$. Take $u = \pi - \frac{\pi_j}{w_j} w$ which is also a left eigenvector and satisfies

$$u_\ell = \pi_\ell - \frac{\pi_j}{w_j} w_\ell = \pi_\ell - \frac{\pi_j}{w_j} \underbrace{\frac{w_\ell}{\pi_\ell}}_{\leq w_j / \pi_j} \pi_\ell \geq 0, \quad \forall \ell = 1, \dots, d,$$

and $u_j = 0$. If $u \neq 0$, then it can be normalized to obtain a probability distribution, but this is a contradiction with the above argument that says that any invariant distribution is strictly positive. Hence the only possibility is $u = 0$, i.e. w proportional to π. $\square$

Definition 8.6 (Aperiodic chain)**.** *Given a transition matrix P, we say that a state x_i is* aperiodic *if $P_{ii}^{(n)} > 0$ for all sufficiently large n, or equivalently if the set $\{n > 0 : P_{ii}^{(n)} > 0\}$ has no common divisor other than 1.*

Using the Chapman-Kolmogorov equation, it is easy to see that if P is irreducible and has an aperiodic state x_i, then all state $x_j \in \mathcal{X}$ are aperiodic. We will then say that P is aperiodic. That periodicity may prevent the sequence $\{\pi^{n,\lambda}\}_n$ to converge, is shown in the following example.

Example 8.3. *Consider the transition matrix* $P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ *which is clearly irreducible and has an invariant distribution* $\pi = (\frac{1}{2}, \frac{1}{2})$*. However, if we take the initial distribution* $\lambda = (1,0)$*, we have* $\pi^{1,\lambda} = \lambda P = (0,1)$*,* $\pi^{2,\lambda} = \pi^{1,\lambda} P = (1,0)$ *and clearly* $\pi^{n,\lambda}$ *does not converge to* π*. The problem in this example is that the transition matrix* P *is periodic with period 2.*

The following result holds.

Lemma 8.7. *Let* P *be an irreducible, aperiodic stochastic matrix. Then* $|\lambda_i(P)| < 1$ *for all* $i = 2, \dots, d$*.*

It follows that if P is irreducible and aperiodic, it has a positive spectral gap $\gamma = 1 - \beta = 1 - \max_{i>1} |\lambda_i(P)| > 0$ which guarantees convergence of $\pi^{n,\lambda} = \lambda P^n$ to the invariant measure π for any choice of initial distribution λ.

We can measure convergence of $\pi^{n,\lambda}$ to π for instance in the ℓ_1 norm which corresponds to total variation (TV) norm between (discrete) probability measures:

$$\|\pi^{n,\lambda} - \pi\|_{\mathrm{TV}} = \|\pi^{n,\lambda} - \pi\|_{\ell^1} = \left\| \frac{\pi^{n,\lambda}}{\pi} - 1 \right\|_{\ell^1_\pi} = \sum_{i=1}^{d} |\pi_i^{n,\lambda} - \pi_i|.$$

Definition 8.7. *An irreducible, aperiodic Markov chain* $\{X_n\}$ *with transition matrix* P *and invariant distribution* π *is*

- geometrically ergodic *if there exists a positive function* $h : \mathcal{X} \to \mathbb{R}$*, with* $\mathbb{E}_\pi[h] < \infty$ *and* $r \in (0,1)$ *such that*

$$\|\pi^{n,\delta_{x_i}} - \pi\|_{TV} \leq h(x_i) r^n, \qquad \forall x_i \in \mathcal{X};$$

- uniformly ergodic *if there exists* $C > 0$ *and* $r \in (0,1)$ *such that*

$$\|\pi^{n,\delta_{x_i}} - \pi\|_{TV} \leq C r^n, \qquad \forall x_i \in \mathcal{X}.$$

From the previous arguments we have the following result:

Theorem 8.8. *Let* P *be an irreducible, aperiodic stochastic matrix on a finite dimensional state space* $\mathcal{X}$*, with invariant distribution* π*. Then, for any initial measure* λ *on* $\mathcal{X}$*, the Markov Chain* $\{X_n\} \sim \mathrm{Markov}\,(\lambda, P)$ *is uniformly ergodic with* $r = \beta = \max_{i>1} |\lambda_i(P)|$ *if* P *is diagonalizable and* $r = (\beta + \varepsilon)$ *for any* $\varepsilon > 0$ *if* P *is not diagonalizable.*

Corollary 8.9. *Under the same assumptions as in Theorem 8.8, if* (P, π) *are in detailed balance, then for any distribution* λ *on* $\mathcal{X}$*, there exists* $C_\lambda > 0$ *such that* $\|\pi^{n,\lambda} - \pi\|_{TV} \leq C_\lambda \beta^n$*.*

In the reversible case (P symmetric with respect to $(\cdot,\cdot)_\pi$), the second eigenvalue of largest modulus β can be equivalently characterized as

$$\beta = \|P - 1\pi\|_{\ell^2_\pi \to \ell^2_\pi} = \sup_{\substack{f \in \ell^2_\pi \\ \mathbb{E}_\pi[f]=0}} \frac{\|Pf\|_{\ell^2_\pi}}{\|f\|_{\ell^2_\pi}}$$

Under the same assumptions of Theorem 8.8, an ergodic theorem holds (even on countable state spaces)

Theorem 8.10. *Let P be irreducible, aperiodic, with invariant distribution π, and $\{X_n\} \sim$ Markov (λ, P). For any $\varphi : \mathcal{X} \to \mathbb{R}$ such that $\mathbb{E}_\pi[|\varphi|] < \infty$, it holds*

$$\mathbb{P}_\lambda \left(\lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} \varphi(X_{j+n_0}) = \mathbb{E}_\pi[\varphi] \right) = 1.$$

Moreover, if $\mathbb{E}_\pi[\varphi^2] < \infty$ and $\{X_n\}$ is uniformly ergodic, then there exists $\tilde{\sigma}^2 > 0$ such that

$$\frac{1}{\sqrt{n}} \left(\sum_{j=1}^{n} \varphi(X_{j+n_0}) - \mathbb{E}_\pi[\varphi] \right) \xrightarrow{d} N(0, \tilde{\sigma}^2).$$

8.2 Markov chains on general state space

We give here a brief overview of how the theory of Markov chains generalizes to a continuous state space $\mathcal{X}$, typically a subset of $\mathbb{R}^d$ with non zero Lebesgue measure.

Definition 8.8. *A Markov transition kernel on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$, where $\mathcal{B}(\mathcal{X})$ is the Borel σ-algebra on $\mathcal{X}$, is a function $P : \mathcal{X} \times \mathcal{B}(\mathcal{X}) \to [0,1]$ s.t.*

1. *for all $x \in \mathcal{X}$, $P(x,\cdot)$ is a probability measure on $\mathcal{X}$,*
2. *for all $A \in \mathcal{B}(\mathcal{X})$, $P(\cdot, A)$ is measurable.*

Whenever $P(x,\cdot)$ admits a density with respect to the Lebesgue measure, we denote it by $p : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ i.e. for all $x \in \mathcal{X}$, $A \in \mathcal{B}(\mathcal{X})$,

$$P(x, A) = \int_A p(x,y)\, dy.$$

Definition 8.9. *Given a Markov transition kernel P and a measure λ on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$, a sequence of random variables $\{X_n \in \mathcal{X}, n \geq 0\}$ is a Markov chain with transition Kernel P and initial distribution λ, in short $\{X_n\} \sim$ Markov (λ, P) if*

- $X_0 \sim \lambda$
- $\mathbb{P}(X_{n+1} \in A \mid X_n = x_n, \dots, X_0 = x_0) = \mathbb{P}(X_{n+1} \in A \mid X_n = x_n) = P(x_n, A)$

A Markov chain $\{X_n\} \sim \text{Markov}(\lambda, P)$ satisfies the strong Markov property. Let τ be a stopping time; conditional on $\tau < +\infty$, it holds

$$\mathbb{E}_\lambda[h(X_{\tau+1}, X_{\tau+2}, \dots)] = \mathbb{E}_{X_\tau}[h(X_1, X_2, \dots)]$$

for any bounded function $h : \mathcal{X}^{\mathbb{N}} \to \mathbb{R}$.

The n-step transition kernel $P^{(n)}(x, A) = \mathbb{P}(X_n \in A \mid X_0 = x)$ is given by the recursion

$$P^{(n)}(x, A) = \int_{\mathcal{X}} P^{(n-1)}(y, A)P(x, dy), \quad P^{(1)}(x, A) = P(x, A).$$

Similarly, if $p^{(n)} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ denotes the density of $P^{(n)}$ (provided it exists), then

$$p^{(n)}(x, y) = \int_{\mathcal{X}} p^{(n-1)}(z, y)p(x, z)\, dz, \quad p^{(1)}(x, y) = p(x, y).$$

To each Markov transition kernel P we can associate the transition operator $\mathcal{P}$ acting to the left on measures, $\mathcal{P} : \mathcal{M}_1(\mathcal{X}) \to \mathcal{M}_1(\mathcal{X})$, with $\mathcal{M}_1(\mathcal{X})$ the set of probability measures on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ as

$$\mu = \lambda\mathcal{P} \quad \Longrightarrow \quad \mu(A) = \int_{\mathcal{X}} P(y, A)\lambda(dy), \quad \forall A \in \mathcal{B}(\mathcal{X}).$$

Notice that

$$\lambda\mathcal{P}^2 = (\lambda\mathcal{P})\mathcal{P} = \int_{\mathcal{X}}\int_{\mathcal{X}} P(x, \cdot)\mathcal{P}(y, dx)\lambda(dy) = \int_{\mathcal{X}} P^{(2)}(y, \cdot)\lambda(dy)$$

so $\mathcal{P}^2$ is the operator associated to $P^{(2)}$ and more generally $\mathcal{P}^n$ is the operator associated to $P^{(n)}$. If $\pi^{n,\lambda}$ denotes the measure of X_n, i.e. $\pi^{n,\lambda}(A) = \mathbb{P}_\lambda(X_n \in A)$, it follows that $\pi^{n,\lambda} = \lambda\mathcal{P}^n = \int_{\mathcal{X}} P^{(n)}(y, \cdot)\lambda(dy)$.

Definition 8.10. *A measure π on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ is called invariant (or stationary) if $\pi = \pi\mathcal{P} = \int_{\mathcal{X}} P(y, \cdot)\pi(dy)$. If the measure π has a density $f : \mathcal{X} \to \mathbb{R}_+$ (i.e. $\pi(A) = \int_A f(y)\, dy$, $\forall A \in \mathcal{B}(\mathcal{X})$), and the kernel P has a density p, then $f(x) = \int_{\mathcal{X}} p(y, x)f(y)\, dy$.*

Similarly, a Markov transition kernel P defines an operator acting on functions to the right, $\mathcal{P} : \mathcal{F}(\mathcal{X}) \to \mathcal{F}(\mathcal{X})$, where $\mathcal{F}(\mathcal{X})$ is the set of measurable functions on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$, as

$$g = \mathcal{P}\varphi \quad \Longrightarrow \quad g(x) = \int_{\mathcal{X}} P(x, dy)\varphi(y) = \mathbb{E}_{\delta_x}[\varphi(X_1)].$$

Definition 8.11. *A chain $\{X_n\}_{n=0}^N \sim \text{Markov}(\lambda, P)$ is reversible if the chain $\{Y_n = X_{N-n}\}_{n=0}^N \sim \text{Markov}(\lambda, P)$.*

As for discrete state spaces, $\{X_n\}$ is reversible if and only if (λ, P) are in detailed balance, which in this case reads

$$\int_A P(x, B)\lambda(dx) = \int_B P(y, A)\lambda(dy), \quad \forall A, B \in \mathcal{B}(\mathcal{X}), \ \ \lambda(A), \lambda(B) > 0.$$

If (P, π) are in detailed balance, then π is an invariant distribution for $\mathcal{P}$. Indeed,

$$\int_{\mathcal{X}} P(x, B)\pi(dx) = \int_B \underbrace{P(y, \mathcal{X})}_{=1}\pi(dy) = \pi(B).$$

8.3 Metropolis-Hastings algorithm in general state space

We generalize here the Metropolis-Hastings algorithm, already introduced in Section 8.1.1, to the case of a general state space, as a tool to construct a Markov Chain $\{X_n\} \sim$ Markov (λ, P) on $\mathcal{X} \subset \mathbb{R}^d$ which has a given invariant measure π with density $f : \mathcal{X} \to \mathbb{R}_+$ with respect to the Lebesgue measure. In the following discussion, we accept that the density f may be know only up to a multiplicative constant, i.e. it does not necessarily integrates to one (in which case, the invariant density is $\tilde{f}(x) = f(x)/\int_{\mathcal{X}} f(y)dy$).

Let $Q : \mathcal{X} \times \mathcal{B}(\mathcal{X}) \to [0,1]$ be a Markov transition kernel with density $q : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$, i.e. $Q(x, A) = \int_A q(x, y)\, dy$ for all $x \in \mathcal{X}$ and $A \in \mathcal{B}(\mathcal{X})$, also called the *proposal* or instrumental *density*, and define the following acceptance rate $\alpha : \mathcal{X} \times \mathcal{X} \to [0,1]$,

$$\alpha(x, y) = \min\left\{\frac{f(y)}{f(x)}\frac{q(y, x)}{q(x, y)}, 1\right\}, \quad \text{if } q(x, y) \neq 0, \qquad \alpha(x, y) = 0, \;\; \text{if } q(x, y) = 0.$$

The Metropolis-Hasting algorithm then reads

Algorithm 8.2: Metropolis-Hastings.

Given: λ (initial measure), q (proposal density), f (target density)

Generate $X_0 \sim \lambda$
for $n = 0, 1, \ldots,$ **do**
  Generate $Y_{n+1} \sim q(X_n, \cdot)$ `// proposal state`
  Generate $U \sim \mathcal{U}(0,1)$ **if** $U \leq \alpha(X_n, Y_{n+1})$ **then**
    set $X_{n+1} = Y_{n+1}$ `// accept proposal`
  else
    set $X_{n+1} = X_n$ `// reject proposal`
  end
end

For the algorithm to work, the chain has to be able to explore the wohle density f. Let us denote $D_f = \text{supp}(f) = \overline{\{x \in \mathcal{X} : f(x) > 0\}}$ the support of f. Minimum requirements are:

- $X_0 \in D_f$, otherwise $\alpha(X_0, \cdot)$ is not defined. This guarantees, in particular, that $X_n \in D_f$, $\forall n$;
- $\bigcup_{x \in D_f} \text{supp}(q(x, \cdot)) \supset D_f$, otherwise the chain fails to visit some parts of D_f.

We derive now the transition kernel P (resp. density p) of the Markov chain generated by the Metropolis-Hastings algorithm. There is a non-zero probability that $X_{n+1} = X_n$, so $P(x_n, \cdot)$ has a point mass in X_n:

$$\mathbb{P}\left(X_{n+1} = x \mid X_n = x\right) = \int_{\mathcal{X}} q(x, y)(1 - \alpha(x, y))\, dy = 1 - \int_{\mathcal{X}} \alpha(x, y) q(x, y)\, dy$$

so the transition density p is

$$p(x, y) = \alpha(x, y) q(x, y) + (1 - \alpha^*(x))\delta_x(y), \quad \alpha^*(x) = \int_{\mathcal{X}} \alpha(x, y) q(x, y)\, dy$$

where $\delta_x(y)$ is a Dirac mass in x. Equivalently, the transition kernel P is given by

$$P(x, A) = \int_A \alpha(x, y)q(x, y)dy + (1 - \alpha^*(x))\mathbb{1}_A(x).$$

As in the finite state space case, we can verify that P and f are in detailed balance.

Lemma 8.11. *The transition kernel P of the Metropolis-Hastings algorithm 8.2, with density $p(x, y) = \alpha(x, y)q(x, y)+(1-\alpha^*(x))\delta_x(y)$ is in detailed balance with the probability density f. Hence f is an invariant probability density for P.*

Proof. Observe first that

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

Hence

$$\begin{aligned} \int_A P(x, B)f(x)\, dx &= \int_A \left(\int_B (\alpha(x, y)q(x, y) + (1 - \alpha^*(x))\delta_x(y))\, dy \right) f(x)\, dx \\ &= \int_A \int_B f(y)\alpha(y, x)q(y, x)\, dy\, dx + \int_{A\cap B} (1 - \alpha^*(x))f(x)\, dx \\ &= \int_B \left(\int_A (\alpha(y, x)q(y, x) + (1 - \alpha^*(y))\delta_y(x))\, dx \right) f(y)\, dy \\ &= \int_B P(y, A)f(y)\, dy. \end{aligned}$$

□

To assess the convergence to equilibrium of the chain, we should further check irreducibility and aperiodicity. In particular, irreducibility should be checked with respect to the invariant density f.

- f-irreducibility is something that should be checked every time depending on the choice of the proposal density. If it holds, then for all $\varphi : \mathcal{X} \to \mathbb{R}$, $\mathbb{E}_f[|\varphi|] < +\infty$,

$$\lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} \varphi(X_j) = \mathbb{E}_f[\varphi] = \int_{\mathcal{X}} \varphi(x)f(x)\, dx.$$

- Concerning aperiodicity, observe that in general $\mathbb{P}(X_{n+1} = x \mid X_n = x) > 0$ as long as $\alpha^*(x) < 1$, since the transition kernel $P(x, \cdot)$ has an atom at x. Consider the set $C = \{x : \alpha(x) < 1\}$. This is a f-zero measure set, i.e. $\int_C f(x)\, dx = 0$, if and only if (exercise) $f(x)q(x, y) = f(y)q(y, x)$ for f-almost every $x, y \in D_f$ which corresponds to the case in which the porposal q is in detailed balance with f, In this case, the acceptance-rejection step is useless and one should check the aperiodicity of q. If, on the other hand, (q, f) are *not* in detailed balance, then the chain is aperiodic. If it is, moreover, f-irreducible, then for any initial distribution $\lambda \ll f$ we have (denoting π the measure associated to f)

$$\lim_{n\to\infty} \|\pi^{n,\lambda} - \pi\|_{\mathrm{TV}} = 0.$$

We describe in the next subsections few methods to choose proposal densities q.

8.3.1 Independence sampler

Let $g : \mathcal{X} \to \mathbb{R}_+$ be a probability density function such that $g(x) > 0$ whenever $f(x) > 0$ (i.e. $f \ll g$). We choose simply $q(x, y) = g(y)$ independently of the current state x (hence the name of *independence sampler*).

Algorithm 8.3: Independence sampler Metropolis-Hastings

Given: $X_0 \sim \lambda$, $\mathrm{supp}(\lambda) \subset D_f$

for $n = 0, 1, \dots,$ **do**

  Generate $Y_{n+1} \sim g$

  Compute $\alpha(X_n, Y_{n+1}) = \min\left\{\frac{f(Y_{n+1})}{f(X_n)}\frac{g(X_n)}{g(Y_{n+1})}, 1\right\}$

  Generate $U \sim \mathcal{U}(0, 1)$ and set

$$X_{n+1} = \begin{cases} Y_{n+1}, & \text{if } U \leq \alpha(X_n, Y_{n+1}) \\ X_n, & \text{otherwise} \end{cases}$$

end

Concerning the convergence to equilibrium, we recall first a useful result for general state space Markov chains.

Lemma 8.12. *Let $P : \mathcal{X} \times \mathcal{B}(\mathcal{X}) \to [0, 1]$ be a Markov transition kernel with invariant measure π. If there exists $\epsilon \in (0, 1)$ and a probability measure ν on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ such that*

$$P(x, A) \geq \epsilon\nu(A), \quad \forall x \in \mathcal{X}, \;\; A \in \mathcal{B}(\mathcal{X}) \tag{8.2}$$

then

$$\|\pi^{n,\lambda} - \pi\|_{TV} \leq 2(1 - \epsilon)^n.$$

More generally, if there exists $k_0 \in \mathbb{N}$ such that $P^{(k_0)}(x, A) \geq \epsilon\nu(A)$ for all $x \in \mathcal{X}$, $A \in \mathcal{B}(\mathcal{X})$, then $\|\pi^{n,\lambda} - \pi\|_{TV} \leq 2(1 - \epsilon)^{\lfloor n/k_0 \rfloor}$. The condition (8.2) *is called* uniform minorizing condition.

In the case of the independence sampler, the following result holds.

Theorem 8.13. *If there exists $M < +\infty$ such that $f(x) \leq Mg(x)$ for all $x \in \mathcal{X}$, then the chain generated by the independence sampler algorithm 8.3 is uniformly ergodic and*

$$\|\pi^{n,\lambda} - \pi\|_{TV} \leq \left(1 - \frac{\int f(x)\,dx}{M}\right)^n, \quad \textit{for any } \lambda.$$

Proof. If f is not normalized, let $\tilde{f} = f/C$, $C = \int_{\mathcal{X}} f$. Notice that

$$\alpha(x, y)q(x, y) = g(y)\min\left\{\frac{f(y)}{f(x)}\frac{g(x)}{g(y)}, 1\right\} = \min\left\{f(y)\underbrace{\frac{g(x)}{f(x)}}_{\geq 1/M}, \underbrace{g(y)}_{\geq f(y)/M}\right\} \geq \frac{1}{M}f(y).$$

It follows that for any $A \in \mathcal{B}(\mathcal{X})$,

$$P(x,A) = \int_A (\alpha(x,y)q(x,y) + (1-\alpha^*(x))\delta_x(y))\,dy \geq \frac{1}{M}\int_A f(y)\,dy \geq \frac{C}{M}\pi(A)$$

and the result follows from Lemma 8.12. □

Under the same condition as in Theorem 8.13, it can be shown that the expected acceptance probability satisfies $\mathbb{E}\left[\alpha(X_n, Y_{n+1})\right] \geq \frac{C}{M}$ (exercise). This result has to be compared with a pure acceptance-rejection sampling strategy, for which the expected acceptance probability is $\frac{C}{M}$. Hence, independence MH sampler accepts more often than a pure acceptance-rejection sampler.

8.3.2 Random walk Metropolis Hastings

Let $g_\sigma : \mathcal{X} \to \mathbb{R}_+$ be a probability density function with zero mean, σ being a scaling parameter; a typical choice is $g_\sigma = N(0,\sigma^2)$. In the random walk Metropolis we choose $q(x,y) = g_\sigma(y-x)$, i.e. the proposal density is g_σ centred in the current state x. If we further assume $g_\sigma(\cdot)$ symmetric around the origin, the acceptance probability takes the simplified form

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

The choice of σ is rather delicate. Small σ imply small steps from the current state, hence high correlation in the chain. Large steps might lead to high rejection rate, hence the chain will stay for a long time in the given state, which also leads to high correlation in the chain. One should then expect that some "optimal" choice of σ exists.

Concerning convergence of this algorithm, one could try to verify a uniform minorizing condition

$$g_\sigma(y-x) \geq \epsilon f(y)$$

for all $x, y \in D_f$. By the same arguments as for independence sampler, this would imply $P(x,A) \geq \epsilon\pi(A)$ hence uniform ergodicity $\|\pi^{n,\lambda} - \pi\|_{\mathrm{TV}} \leq 2(1-\epsilon)^n$ for all initial distributions λ. However, such minoirizing condition does not hold, in general for unbounded or non-compact $D_f \subset \mathcal{X}$. We mention a result by Mengersen and Tweedie ('96) showing geometric ergodicity for tail-log-concave f and $\mathcal{X} = \mathbb{R}$.

Definition 8.12. *A probability density function f on $\mathbb{R}$ is log-concave in the tails if there exists $\alpha, M > 0$ such that $\log f(x) - \log f(y) \geq \alpha(|y| - |x|)$ for all $|y| \geq |x| \geq M$.*

Theorem 8.14. *If the invariant density f on $\mathbb{R}$ is log concave in tails for some $\alpha, M > 0$ and $\inf_{|x|\leq R} f(x) > 0$ for all $R > 0$, then the Markov chain generated by the random walk Metropolis-Hastings algorithm with symmetric proposal $g_\sigma(\cdot)$ is geometrically ergodic.*

8.3.3 One Variable at a time Metropolis-Hastings

Suppose that a state $x \in \mathcal{X}$ has several components, $x = (x^{(1)}, \ldots, x^{(d)})$, with $x^{(i)} \in \mathcal{X}^{(i)}$. Once can thus construct a Metropolis-Hastings algorithm by updating one component at a time, either chosen randomly or by performing a systematic sweep over the components. Say that the i-th component has been chosen. We use the notation $x = (x^{(i)}, x^{(-i)})$ with $x^{(-i)} = (x^{(1)}, \ldots, x^{(i-1)}, x^{(i+1)}, \ldots, x^{(d)})$. Let $q_i : \mathcal{X} \times \mathcal{X}^{(i)} \to \mathbb{R}$ be a family of proposal density functions on $\mathcal{X}^{(i)}$, i.e. $q_i(x, \cdot)$ is a density function on $\mathcal{X}^{(i)}$ for any $x \in \mathcal{X}$. Then the one variable at a time MH algorithm with random coordinate selection reads:

Algorithm 8.4: One variable at a time MH with random selection.

Generate $X_0 \sim \lambda$
for $n = 0, 1, \ldots$ **do**
  Draw index $i_n \sim \beta$ (p.m.f on $\{1, \ldots, d\}$)
  Draw $y \sim q_{i_n}(X_n, \cdot)$ and set $Y_{n+1} = (y, X_n^{(-i_n)})$
  Compute $\alpha_{i_n}(X_n, Y_{n+1}) = \min\left\{\frac{f(Y_{n+1})}{f(X_n)} \frac{q_{i_n}(Y_{n+1}, X_n^{(i_n)})}{q_{i_n}(X_n, Y_{n+1}^{(i_n)})}, 1\right\}$
  Set $X_{n+1} = \begin{cases} Y_{n+1} & \text{with prob. } \alpha_{i_n}(X_n, Y_{n+1}) \\ X_n & \text{otherwise} \end{cases}$
end

whereas the one variable at a time MH algorithm with systematic sweep over the coordinates reads:

Algorithm 8.5: One variable at a time MH with systematic sweep.

Generate $X_0 \sim \lambda$
for $n = 0, 1, \ldots$ **do**
  Set $Y_{n+1,0} = X_n$
  for $i = 1, \ldots, d$ **do**
    Draw $y \sim q_i(X_n, \cdot)$ and set $\tilde{Y} = (y, Y_{n+1,i-1}^{(-i)})$
    Set $Y_{n+1,i} = \begin{cases} \tilde{Y}, & \text{with prob. } \alpha_i(Y_{n+1,i-1}, \tilde{Y}) \\ Y_{n+1,i-1}, & \textit{otherwise} \end{cases}$
  end
  $X_{n+1} = Y_{n+1,d}$
end

Bibliography

[1] Y. S. Chow and Herbert Robbins. On the asymptotic theory of fixed-width sequential confidence intervals for the mean. *Ann. Math. Statist.*, 36:457–462, 1965.

[2] D. Kroese, T. Taimre, and Z. Botev. *Handbook of Monte Carlo Methods.* Wiley, 2011.

[3] P. L'Ecuyer and R. Simard. TestU01: A C library for empirical testing of random number generators. *ACM Transactions on Mathematical Software*, 33(Article 22), 2007.

[4] S. Meyn and R.L. Tweedie. *Markov chains and stochastic stability.* Cambridge University Press, Cambridge, second edition, 2009. With a prologue by Peter W. Glynn.