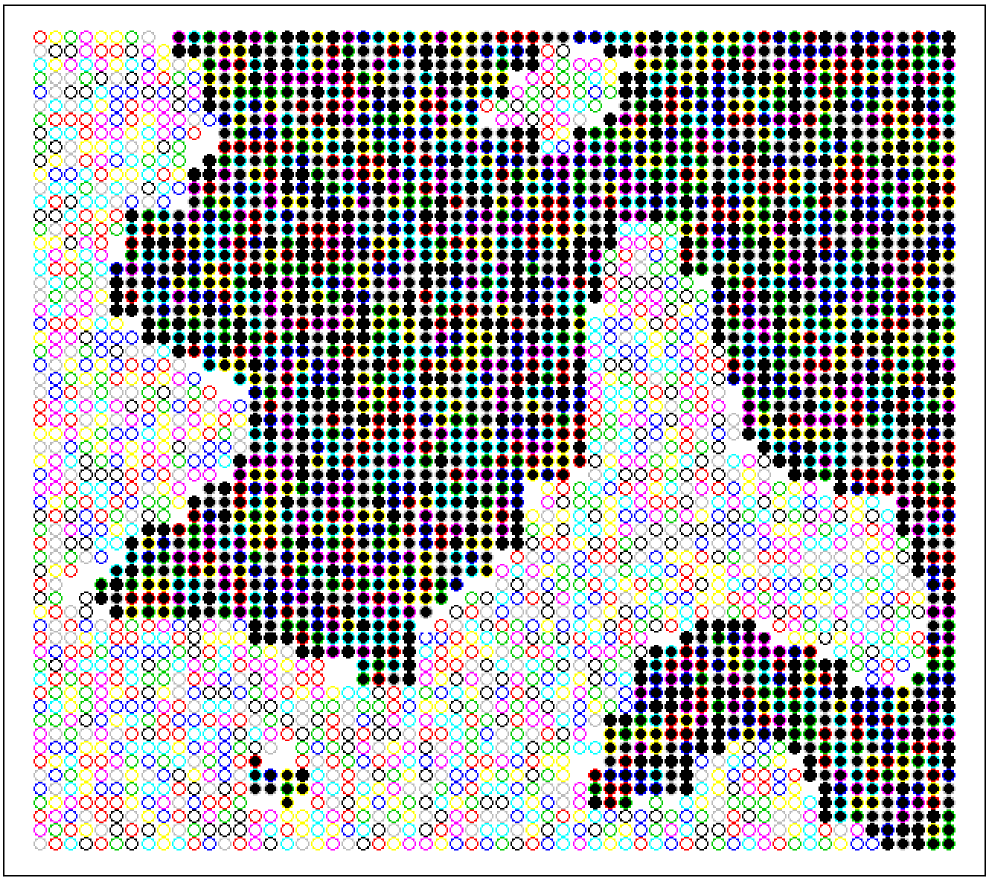
Part A
Simulation

Matthias Winkel
Department of Statistics
University of Oxford

These notes benefitted from the TT 2010 notes by Geoff Nicholls.



TT 2011

Part A
Simulation

Matthias Winkel – 8 lectures TT 2011

Prerequisites

Part A Probability and Mods Statistics

Aims

This course introduces Monte Carlo methods, collectively one of the most important analytical tools of modern Statistical Inference.

Synopsis

Motivation. Inversion. Transformation methods.
Rejection.
Variance reduction via Importance sampling.
The Metropolis algorithm (finite space), reversibility, ergodicity.
Applications: conditioned and extreme events, likelihood and missing data, sampling uniformly at random.

Problem sheets in classes will include a separate section with some examples of simulation using `R`.

Reading

S. M. Ross: *Simulation.* Elsevier, 4th edition, 2006
J. R. Norris: *Markov chains.* CUP, 1997
C. P. Robert and G. Casella: *Monte Carlo Statistical Methods.* Springer, 2004
B. D. Ripley: *Stochastic Simulation.* Wiley, 1987

Contents

1 Introduction 1

1.1 Motivation . . . 1

1.2 Overview . . . 2

1.3 Structure of the course . . . 6

2 Inversion and transformation methods 7

2.1 The inversion method . . . 7

2.2 The transformation method . . . 9

3 The rejection method 11

3.1 Simulating from probability density functions? . . . 11

3.2 Conditioning by repeated trials . . . 12

3.3 Rejection . . . 13

4 Refining simulation methods 15

4.1 More rejection . . . 15

4.2 Random permutations . . . 16

4.3 Discrete distributions and random parameters . . . 17

5 Importance sampling 19

5.1 Monte-Carlo integration . . . 19

5.2 Variance reduction via importance sampling . . . 21

5.3 Example: Importance sampling for Normal tails . . . 22

6 Markov chain Monte-Carlo 25

6.1 Simulation of Markov chains . . . 25

6.2 Monte-Carlo algorithm . . . 26

6.3 Reversible Markov chains . . . 28

6.4 The Metropolis-Hastings algorithm . . . 28

7 The Metropolis-Hastings algorithm 29

7.1 The basic Metropolis-Hastings algorithm . . . 29

7.2 The Metropolis-Hastings algorithm in a more general setting . . . 31

7.3 Example: the Ising model . . . 32

8 Unknown normalisation constants **37**
8.1 Normalisation constants 37
8.2 Rejection method 38
8.3 Importance sampling 38
8.4 Metropolis-Hastings algorithm 40
8.5 Conclusion 40

A Assignments **I**
A.1 Inversion, transformation and rejection III
A.2 Rejection and importance sampling V
A.3 Importance sampling and MCMC VII

B Solutions **IX**
B.1 Inversion, transformation and rejection IX
B.2 Rejection and importance sampling XIII
B.3 Importance sampling and MCMC XVII

C Solutions to optional `R` questions **XXI**
C.1 Inversion, transformation and rejection XXI
C.3 Importance sampling and MCMC XXIV

Lecture 1

Introduction

Reading: Ross Chapters 1 and 3
Further reading: Ripley Chapters 1, and 2 (25 years old, but the caveat is instructive)

1.1 Motivation

The aim of this section is to give brief reasons why simulation is useful and why mathematicians and statisticians should be interested in simulation. To focus ideas, we begin by an example and an attempt at defining what simulation is.

Example 1 Ross's book starts by considering an example of a pharmacist filling prescriptions. At a qualitative level, customers arrive and queue to be served. Their prescriptions require varying amounts of service time. The pharmacist cares about both customer satisfaction and not excessively exceeding his 9am-5pm working day.

A mathematical approach to the problem is to set up stochastic models for the arrival process of customers and for the service times, calibrated based on past experience. One of the most popular models is to consider independent exponentially distributed inter-arrival and service times, with certain service and inter-arrival parameters. This model is analytically tractable (see Part B Applied Probability), but it is only a first approximation of reality. We want to model varying arrival intensities and non-exponential service times that may interact with quantities such as the number of people in the queue.

Let us focus on a specific question. If the pharmacist continues to deal with all customers in the pharmacy at 5pm but no more new customers, what proportion of days will he finish after 5.30pm? Once a precise model has been formulated, a computer can simulate many days of customers and compute the proportion of simulated days he would finish after 5.30pm, a Monte-Carlo average approximating the model answer, which will be close to the real answer if our model is sufficiently realistic.

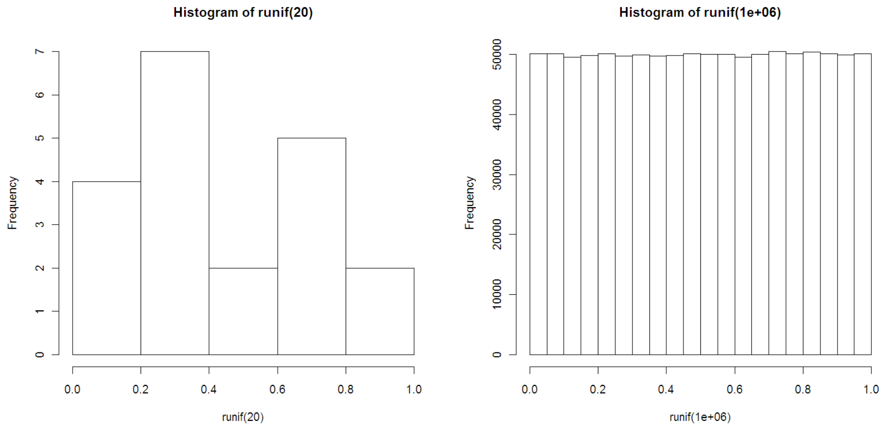
Simulation is a subject that can be studied in different ways and with different aims. As a mathematical subject, the starting point is often a sequence of independent random variables $(U_k, k \geq 1)$ that are uniformly distributed on the interval $(0, 1)$; notation: $U_k \sim \text{Unif}(0, 1)$. The aim of the game is to generate from these uniform random variables more complicated random variables and stochastic models. From an algorithmic point of view, the efficiency of such generation of random variables can be analysed. The implementation of efficient algorithms makes simulation useful to applied disciplines.

The most basic use in applied or indeed theoretical disciplines is to repeatedly and independently observe realisations of a stochastic model, which on the one hand can be compared with and related to real world systems and on the other hand can give rise to conjectures about the stochastic model itself that can then be approached mathematically. A fundamental feature of mathematical modelling is the trade-off between proximity to the real world and (analytic) tractability. Simulation methods allow to gain more proximity to the real world while keeping (computational) tractability. The complexity of the stochastic model often poses some challenges. More specific uses of simulation include the calibration of parameters in a complex system to control the failure probability or the approximation of critical values for a statistical test where the distribution of the test statistic is not explicit. Simulation then becomes a tool to help with decision making.

In this course, you will see some implementation in the statistical computing package R for illustration purposes and to help your intuition. The main focus however is on some key mathematical techniques building on Part A Probability and Mods Statistics. Indeed, simulation allows to illustrate notions and results from those courses that enhance understanding of the material of those courses.

1.2 Overview

Consider a sequence of independent Unif$(0,1)$ random variables $(U_k, k \geq 1)$. In R, any number n of these can be generated by the command runif(n). Figure 1.1 shows two plots of, respectively 20 and 1,000,000 uniform random variables in a histogram. In the first histogram, the actual realisations of uniform variables have been added between the bars and the x-axis. Note the substantial deviation from bars of equal height due to randomness in sampling. This effect disappears as the number of variables tends to infinity – this is essentially a consequence of the (Strong) Law of Large Numbers, which implies that the proportion in each of the 20 boxes converges to 1/20 almost surely. More generally, the Glivenko-Cantelli theorem shows that empirical cumulative distribution functions converge uniformly to the underlying cumulative distribution function.



R code:

```
x<-runif(20)
hist(x)
x<-runif(1000000)
hist(x)
```

Figure 1.1: Histograms of Unif$(0,1)$, resp. $n = 20$ and $n = 1,000,000$, with R code.

Strictly speaking, any computer will generate at best *pseudo*-random numbers (deterministic, but apparently random), the question of what actual random numbers are has a deep philosophical dimension. While the mathematical theory (based on Kolmogorov's axioms) is sound, the interface with the real world, also computers, is problematic.

However, the behaviour of random numbers generated by modern random number generators (whose construction is based on number theory $N_{k+1} = (aN_k + c) \mod m$ for suitable a, c, m, then $U_{k+1} = N_{k+1}/(m+1)$, e.g. $m = 2^{19937} - 1$ as the default in R), resembles mathematical random numbers in many respects. In particular, standard tests for uniformity, independence etc. do not show significant deviations. We will not go into any more detail of the topics touched upon here and refer the reader to the literature.

The following examples introduce some important ideas to dwell on and they collectively give an overview of the course.

Example 2 For independent $U_1, U_2 \sim \text{Unif}(0,1)$, the pair (U_1, U_2) has bivariate uniform distribution on $(0,1)^2$; the variable $V_1 = a + (b-a)U_1$ is uniformly distributed on (a,b) with density $1/(b-a)$ on (a,b); the pair $(V_1, V_2) = (a + (b-a)U_1, c + (d-c)U_2)$ is uniformly distributed on $(a,b) \times (c,d)$. This is an example of the *transformation method*, here, linear transformation. Non-linear transformations give non-uniform distributions, in general. For example, $W_1 = -\ln(1-U_1) \sim \text{Exp}(1)$, since

$$F(w) = \mathbb{P}(W_1 \le w) = \mathbb{P}(-\ln(1-U_1) \le w) = \mathbb{P}(U_1 \le 1 - e^{-w}) = 1 - e^{-w}, \qquad w \ge 0.$$

Note that $F^{-1}(u) = -\ln(1-u)$, $u \in [0,1)$, is the inverse cumulative distribution function. Setting $W_1 = F^{-1}(U_1)$ is an instance of the *inversion method.*

Discrete uniform distributions can be obtained as $Z_1 = [nU_1]$ where $[\cdot]$ denotes the integer part function:

$$\mathbb{P}([nU_1] = j) = \mathbb{P}(j \le nU_1 < j+1) = \mathbb{P}(U_1 \in [j/n, (j+1)/n)) = 1/n, \quad j = 0, \ldots, n-1.$$

Example 3 Uniform distributions on subsets $A \subset [a,b] \times [c,d]$ can be obtained via conditioning: conditionally given that $(V_1, V_2) \in A$, the pair (V_1, V_2) is uniformly distributed on A. Intuitively, if we repeat simulating $(V_1^{(k)}, V_2^{(k)})$ until we get $(V_1^{(k)}, V_2^{(k)}) \in A$, the resulting $(V_1, V_2) = (V_1^{(k)}, V_2^{(k)})$ is uniformly distributed on A.

Let us formulate a specific example in pseudo-code, i.e. phrases close to computer code that are easily implemented, but using fuller sentence structure:

```
1. Generate two independent random numbers $U_1 \sim \text{Unif}(0,1)$ and $U_2 \sim \text{Unif}(0,1)$.
2. If $U_1^2 + U_2^2 < 1$, go to 3., else go to 1.
3. Return the vector $(U_1, U_2)$.
```

This algorithm simulates a uniform variable in the quarter disk. This is an example of the *rejection method.* Analogously for a uniform distribution on $A = [0,1]^2 \cup [1,2]^2$:

```
1. Generate two independent random numbers $U_1 \sim \text{Unif}(0,1)$ and $U_2 \sim \text{Unif}(0,1)$.
2. Set $(V_1, V_2) = (2U_1, 2U_2)$. If $(V_1, V_2) \in A$, go to 3., else go to 1.
3. Return the vector $(V_1, V_2)$.
```

Note that (V_1, V_2) has marginal distributions $V_1, V_2 \sim \text{Unif}(0,2)$, but V_1 and V_2 are not independent. Generating dependent random variables from independent ones is not just useful in simulation, but also in measure-theoretic probability/integration, where the first step is the existence of Lebesgue measure on $[0,1]$, the next steps Lebesgue measure on finite products $[0,1]^n$ and infinite products $[0,1]^{\mathbb{N}}$. Product measures reflect independence and can be transformed/restricted to get measures with dependent marginals.

Example 4 Sometimes we do not need the exact distribution of a random variable X, but only a tail probability $\mathbb{P}(X > c)$ or an expectation $\mathbb{E}(X)$ or $\mathbb{E}(f(X))$. We could use the (Strong) Law of Large Numbers on independent copies X_j of X to approximate

$$\frac{X_1 + \cdots + X_n}{n} \to \mathbb{E}(X) \qquad \text{“almost surely” (strengthens “in probability”) as } n \to \infty.$$

This result holds as soon as $\mathbb{E}(|X|) < \infty$. We can always apply it to random variables

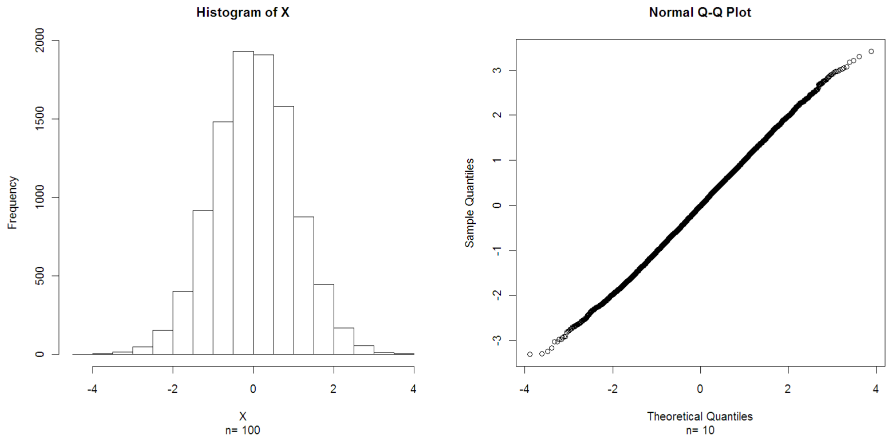
$$Y_j = 1_{\{X_j > c\}} = \begin{cases} 1 & \text{if } X_j > c, \\ 0 & \text{otherwise,} \end{cases}$$

and to $Z_j = f(X_j)$ if $\mathbb{E}(|Z|) = \mathbb{E}(|f(X)|) < \infty$. This method is often not very efficient, particularly if c is large or f is small for most, but not all possible values of X. *Importance sampling* boosts the number and compensates the weight of “relevant” realisations of X.

Example 5 The normal distribution is the limit in the Central Limit Theorem (CLT):

$$Z_n = \frac{U_1 + \cdots + U_n - n/2}{\sqrt{n/12}} \to \text{Normal}(0,1), \qquad \text{in distribution, as } n \to \infty,$$

so for large n, the approximating Z_n will be almost normally distributed, see Figure 1.2. The case $n = 12$ used to be popular, but is not accurate, and larger n are less efficient. But, we can similarly use the Markov chain convergence and ergodic theorems to approximate stationary distributions of Markov chains. Example: for a graph (V, E), consider the random walk on V, where each step is to a neighbouring vertex chosen uniformly at random. A much more general example is the *Metropolis-Hastings algorithm.*



R code:

```
n<-100
k<-10000
U<-runif(n*k)
M<-matrix(U,n,k)
X<-apply(M,2,'sum')
Z<-(X-n/2)/sqrt(n/12)
hist(Z)
qqnorm(Z)
```

Figure 1.2: Histogram and Q-Q plot of $k = 10,000$ Normal(0,1) via CLT for $n = 100$.

We can actually investigate the quality of approximation, when we “apply” the CLT for small values of n. See Figure 1.3, which uses the same code as Figure 1.2, just for $n = 1$, $n = 2$, $n = 3$ and $n = 5$. We see that the approximation, which starts with uniform and triangular distributions for $n = 1$ and $n = 2$ becomes quite good (at this graphical level) for $n = 5$, and further improves in the tails for $n = 10$ and $n = 100$.

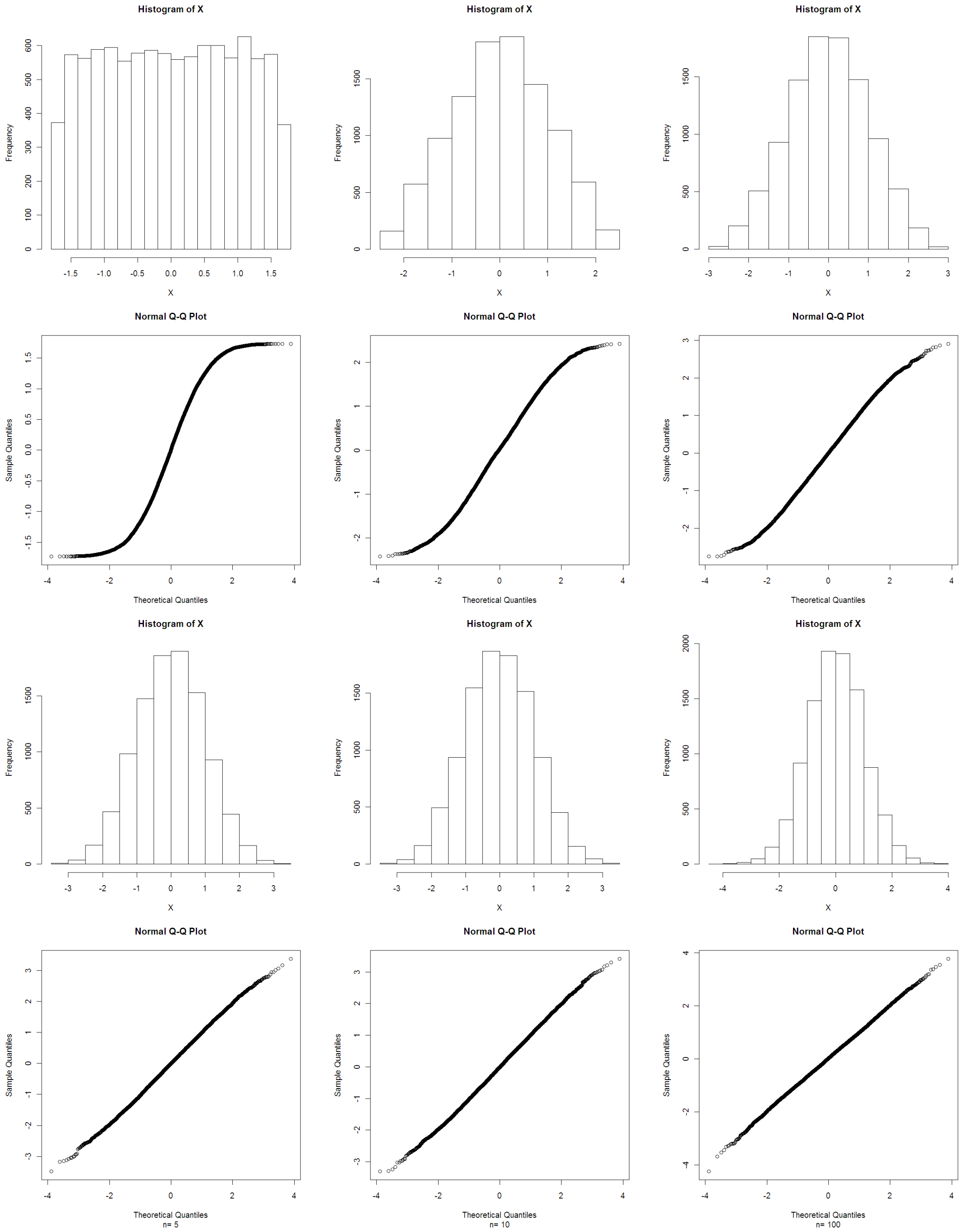


Figure 1.3: Histogram, Q-Q plot of $k = 10,000$ Normal$(0, 1)$ for $n = 1, 2, 3, 5, 10, 100$.

1.3 Structure of the course

We will now go back to the simulation of real-valued and multivariate (mostly bivariate) random variables. Lecture 2 will treat the *inversion method* to simulate general one-dimensional distributions and the *transformation method* as a generalisation to deal with the multivariate case. Lecture 3 will discuss the *rejection method* and some more examples combining rejection and transformation. Most exercises for this material will be on Assignment 1.

Lectures 4 and 5 will refine the rejection method to deal with importance sampling, Assignment 2. This leaves Lectures 6-8 for the Metropolis-Hastings algorithm and more about Markov chain Monte-Carlo (MCMC) and applications, Assignment 3.

Lecture 2

Inversion and transformation methods

Reading: Robert and Casella Sections 2.1 and 2.2, Ross Section 5.1
Further reading: Ripley Chapter 3

2.1 The inversion method

Recall that a function $F\colon \mathbb{R} \to [0,1]$ is a cumulative distribution function if

- F is increasing, i.e. $x \leq y \Rightarrow F(x) \leq F(y)$;
- F is right-continuous, i.e. $F(x+\varepsilon) \to F(x)$ as $\varepsilon \downarrow 0$;
- $F(x) \to 0$ as $x \to -\infty$;
- and $F(x) \to 1$ as $x \to \infty$.

A random variable $X\colon \Omega \to \mathbb{R}$ has cumulative distribution function F if $\mathbb{P}(X \leq x) = F(x)$ for all $x \in \mathbb{R}$. This implies

- $\mathbb{P}(X \in (a,b]) = F(b) - F(a)$, $\mathbb{P}(X \in [a,b]) = F(b) - F(a-)$, $\mathbb{P}(X \in (a,b)) = F(b-) - F(a)$ etc., where $F(a-) = \lim_{\varepsilon \downarrow 0} F(a-\varepsilon)$. In particular, $\mathbb{P}(X = a) = F(a) - F(a-) > 0$ iff F jumps at a.
- If F is differentiable on $\mathbb{R}$, with derivative f, then X is continuously distributed with probability density function f.

If $F\colon \mathbb{R} \to (0,1)$ is a continuous and strictly increasing cumulative distribution function, we can define its inverse $F^{-1}\colon (0,1) \to \mathbb{R}$.

Proposition 6 (Inversion) *If $F\colon \mathbb{R} \to (0,1)$ is a continuous and strictly increasing cumulative distribution function with inverse F^{-1}, then $F^{-1}(U)$ has cumulative distribution function F, if $U \sim \mathrm{Unif}(0,1)$.*

In fact, this result holds for every cumulative distribution function, provided we appropriately define $F^{-1}(u)$. We will explore this on the first assignment sheet, see also Example 8 below.

Proof: Let $x \in \mathbb{R}$. Since F and F^{-1} are strictly monotonic increasing, application of F or F^{-1} preserves inequalities, and so

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

Since this holds for all $x \in \mathbb{R}$, $F^{-1}(U)$ has cumulative distribution function F. □

This is useful only if we have the inverse cumulative distribution function at our disposal. This is not usually the case, but it includes the case for the exponential distribution and more generally for the Weibull distribution. If we slightly modify the proposition to apply to strictly increasing continuous functions $F\colon [0,\infty) \to [0,1)$ with $F(0) = 0$ and $F(x) \to 1$ as $x \to \infty$, the proof still applies (but as we have said it actually holds in full generality):

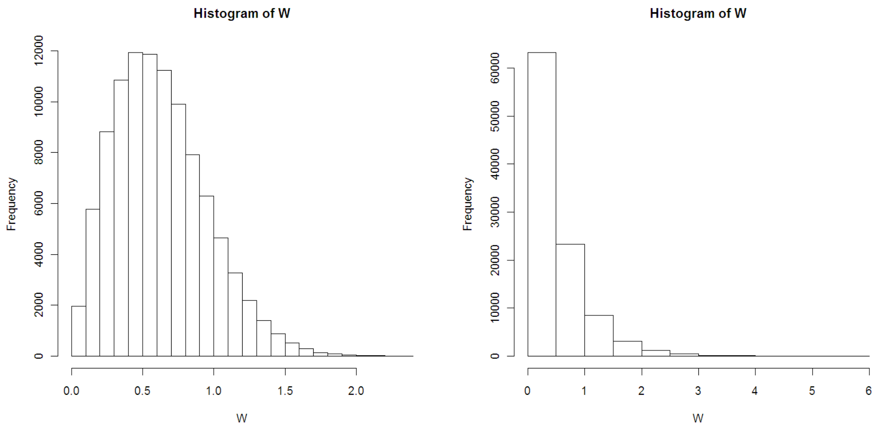
Example 7 (Weibull) Let $\alpha, \lambda \in (0,\infty)$. The distribution with survival function

$$\overline{F}(x) = 1 - F(x) = \exp\left(-\lambda x^{\alpha}\right), \qquad x \geq 0,$$

is called the Weibull distribution. We calculate

$$u = F(x) \iff \ln(1-u) = -\lambda x^{\alpha} \iff x = (-\ln(1-u)/\lambda)^{1/\alpha},$$

so $F^{-1}(u) = (-\ln(1-u)/\lambda)^{1/\alpha}$. Since, $U \sim 1 - U$, we can generate a Weibull random variable as $(-\ln(U)/\lambda)^{1/\alpha}$, see Figure 2.1



R code:

```
n<-100000
U<-runif(n)
a<-2
r<-2
W<-(-log(U)/r)^(1/a)
hist(W)
a<-1
W<-(-log(U)/r)^(1/a)
hist(W)
```

Figure 2.1: Histogram of Weibull(2,2) and Weibull(1,2)=Exp(2)

Clearly, cumulative distribution functions of $\mathbb{N}$-valued random variables are neither continuous nor strictly increasing. Let us analyse this case.

Example 8 (Bernoulli) Consider the Bernoulli distribution with cumulative distribution function

$$F(x) = \begin{cases} 0 & x < 0 \\ 1-p & 0 \leq x < 1 \\ 1 & 1 \leq x \end{cases}$$

Is there an increasing function $F^{-1}:(0,1)\to\mathbb{R}$ such that $F^{-1}(U)\sim$ Bernoulli(p)? Since we want F^{-1} to only take values 0 and 1 (as a Bernoulli variable does), and length $1-p$ of $(0,1)$ should be mapped onto 0, length p onto 1, we need

$$F^{-1}(u)=\begin{cases}0 & 0<u<1-p\\ 1 & 1-p<u<1.\end{cases}$$

The value of F^{-1} at $1-p$ is irrelevant, but we can choose right-continuity.

In general, the same reasoning, gives, by induction, that for a general $\mathbb{N}$-valued random variable with $\mathbb{P}(X=n)=p(n)$, we get

$$F^{-1}(u)=n \iff F(n-1)=\sum_{j=0}^{n-1}p(j)<u<\sum_{j=0}^{n}p(j)=F(n).$$

This is the same as $F^{-1}(u)=\inf\{n\ge 0:F(n)>u\}$ for all $u\in(0,1)\setminus\{F(n),n\ge 0\}$. Note that right-continuity of F^{-1} will hold if we ask for $F^{-1}(u)=\inf\{n\ge 0:F(n)>u\}$ everywhere, whereas $u\mapsto F^{-1}(u-)=\inf\{n\ge 0:F(n)\ge u\}$ is left-continuous.

For many distributions, the inverse cumulative distribution function is not available, at least not in a usefully explicit form. Sometimes the use of several independent Unif$(0,1)$-variables allows some more efficient transformations.

2.2 The transformation method

The inversion method transforms a single uniform random variable into the random variable required. It is sometimes more efficient to use several independent uniform (or other previously generated) random variables. Since the key reason for doing this is efficiency and not generality, we discuss a selection of useful examples.

Example 9 (Gamma) The two-parameter Gamma distribution has probability density function $\Gamma(\alpha)^{-1}\lambda^\alpha x^{\alpha-1}e^{-\lambda x}$ on $(0,\infty)$. For independent $X_j\sim$ Gamma(α_j,λ), we have $X_1+\cdots+X_n\sim$ Gamma$(\alpha_1+\cdots+\alpha_n,\lambda)$. Since Gamma$(1,\lambda)=$ Exp(λ), we can represent Gamma(n,λ) variables as $Z=X_1+\cdots+X_n$ for $X_j\sim$ Exp(λ). With $X_j=-\ln(U_j)/\lambda$, this can be written as

$$Z=T(U_1,\ldots,U_n)=-\sum_{j=1}^{n}\ln(U_j)/\lambda=-\ln\left(\prod_{j=1}^{n}U_j\right)/\lambda.$$

For continuously distributed random variables, a tool is the transformation formula for probability density functions (a corollary of the change of variables formula for multiple integrals): for $D\subset\mathbb{R}^m$ and $R\subset\mathbb{R}^m$, let $T=(T_1,\ldots,T_m):D\to R$ be a diffeomorphism whose inverse has Jacobian matrix

$$J(z)=\left(\frac{d}{dz_j}T_i^{-1}(z)\right)_{1\le i,j\le m}.$$

Then a D-valued random variable X with probability density function f_X gives rise to an R-valued transformed random variable $Z=T(X)$ with probability density function

$$f_Z(z)=f_X(T^{-1}(z))|\det(J(z))|,\qquad z\in R.$$

Proposition 10 (Box-Muller) *For independent* $U_1, U_2 \sim \mathrm{Unif}(0,1)$, *the pair*

$$Z = (Z_1, Z_2) = T(U_1, U_2) = (\sqrt{-2\ln(U_1)}\cos(2\pi U_2), \sqrt{-2\ln(U_1)}\sin(2\pi U_2))$$

is a pair of independent $\mathrm{Normal}(0,1)$ *random variables.*

Proof: $T\colon (0,1)^2 \to \mathbb{R}^2$ is bijective. Note $Z_1^2 + Z_2^2 = -2\ln(U_1)$ and $Z_2/Z_1 = \tan(2\pi U_2)$, so

$$(U_1, U_2) = T^{-1}(Z_1, Z_2) = (e^{-(Z_1^2+Z_2^2)/2}, (2\pi)^{-1}\arctan(Z_2/Z_1))$$

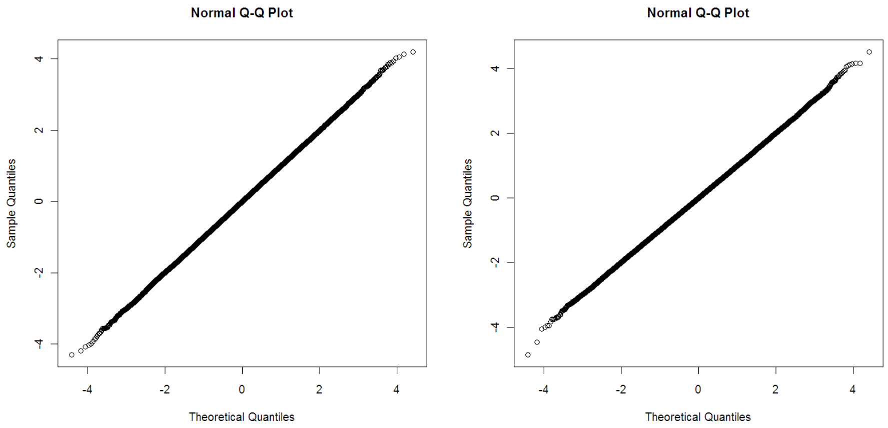
(on appropriate branches of arctan, not important here). The Jacobian of T^{-1} is

$$J(z) = \begin{pmatrix} -z_1 e^{-(z_1^2+z_2^2)/2} & -z_2 e^{-(z_1^2+z_2^2)/2} \\ -\frac{1}{2\pi}\frac{z_2}{z_1^2}\frac{1}{1+z_2^2/z_1^2} & \frac{1}{2\pi}\frac{1}{z_1}\frac{1}{1+z_2^2/z_1^2} \end{pmatrix} \qquad \Rightarrow |\det(J(z))| = \frac{1}{2\pi}e^{-(z_1^2+z_2^2)/2}$$

and so, as required,

$$f_Z(z) = f_{U_1,U_2}(T^{-1}(z))|\det(J(z))| = \frac{1}{2\pi}e^{-(z_1^2+z_2^2)/2} = \frac{1}{\sqrt{2\pi}}e^{-z_1^2/2}\frac{1}{\sqrt{2\pi}}e^{-z_2^2/2}.$$

□



R code:

```
n<-100000
U1<-runif(n)
U2<-runif(n)
C<--sqrt(-2*log(U1))
Z1<-C*cos(2*pi*U2)
Z2<-C*sin(2*pi*U2)
qqnorm(Z1)
qqnorm(Z2)
```

Figure 2.2: Q-Q plots of $n = 100,000$ Box-Muller samples.

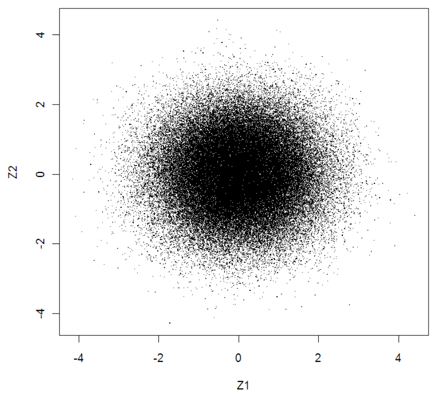


Figure 2.3: plot of pairs of $n = 100,000$ Box-Muller samples: `plot(Z1,Z2,pch=".")`

Lecture 3

The rejection method

Reading: Ross Sections 5.2 and 5.3, Section 2.3
Further reading: Ripley Chapter 3

3.1 Simulating from probability density functions?

The inversion method is based on (inverses of) cumulative distribution functions. For many of the most common distributions, the cumulative distribution function is not available explicitly. Often, distributions are defined via their probability mass functions ("discrete case") or probability density functions ("continuous case"). Let us here consider the continuous case.

Lemma 11 *Let $f : \mathbb{R} \to [0, \infty)$ be a probability density function. Consider*

- *X with density f, and conditionally given $X = x$, let $Y \sim \text{Unif}(0, f(x))$;*
- *$V = (V_1, V_2)$ uniformly distributed on $A_f = \{(x, y) \in \mathbb{R} \times [0, \infty) : 0 \leq y \leq f(x)\}$.*

Then (X, Y) and (V_1, V_2) have the same joint distribution.

Proof: Compare joint densities. Clearly $\mathbb{P}((X, Y) \in A_f) = \mathbb{P}(X \in \mathbb{R}, 0 \leq Y \leq f(X)) = 1$ means we only need to consider $(x, y) \in A_f$:

$$f_{X,Y}(x, y) = f_X(x) f_{Y|X=x}(y) = f(x) \frac{1}{f(x)} = 1 = f_{V,W}(x, y).$$

□

So, it seems that we can simulate X by simulating (V_1, V_2) from the uniform distribution on A_f and setting $X = V_1$. This is true, but not useful in practice. It is, however, a key to the rejection method, because it reduces the study of continuous distributions on $\mathbb{R}$ to the more elementary study of uniform distributions on subsets of $\mathbb{R}^2$ (with area 1, so far).

Let $C \subset \mathbb{R}^2$ have area $\text{area}(C) \in (0, \infty)$. Recall that $V \sim \text{Unif}(C)$ means

$$\mathbb{P}(V \in A) = \frac{\text{area}(A \cap C)}{\text{area}(C)}, \qquad A \subset \mathbb{R}^2.$$

An elementary property of uniform distributions is that associated conditional distributions are also uniform: for $B \subset C$ with $\text{area}(B) > 0$, we have

$$\mathbb{P}(V \in A | V \in B) = \frac{\mathbb{P}(V \in A \cap B)}{\mathbb{P}(V \in B)} = \frac{\text{area}(A \cap B)}{\text{area}(B)}, \qquad A \subset \mathbb{R}^2.$$

Another elementary property of uniform distributions is that linear transformations of uniformly distributed random variables are also uniform.

Example 12 Let $h \colon \mathbb{R} \to [0, \infty)$ be a density, $V = (V_1, V_2) \sim \text{Unif}(A_h)$ and $M > 0$, then $(V_1, MV_2) \sim \text{Unif}(A_{Mh})$, where $A_{Mh} = \{(x, y) \in \mathbb{R} \times [0, \infty) : 0 \leq y \leq Mh(x)\}$. This follows straight from the transformation formula for their joint densities.

3.2 Conditioning by repeated trials

To simulate conditional distributions, we can use the following observation:

Lemma 13 *Let $X, X_1, X_2, \ldots$ be independent and identically distributed $\mathbb{R}^d$-valued random variables, $B \subset \mathbb{R}^d$ with $p = \mathbb{P}(X \in B) > 0$. Let $N = \inf\{n \geq 1 : X_i \in B\}$. Then*

- *$N \sim \text{geom}(p)$, i.e. $\mathbb{P}(N = n) = (1 - p)^{n-1} p$, $n \geq 1$,*
- *and $\mathbb{P}(X_N \in A) = \mathbb{P}(X \in A | X \in B)$ for all $A \subset \mathbb{R}^d$,*
- *N and X_N are independent.*

Proof: We calculate the joint distribution, for all $n \geq 1$ and $A \subset \mathbb{R}^d$

$$\begin{aligned} \mathbb{P}(N = n, X_N \in A) &= \mathbb{P}(X_1 \notin B, \ldots, X_{n-1} \notin B, X_n \in A \cap B) \\ &= (1 - p)^{n-1} \mathbb{P}(X_n \in A \cap B) = (1 - p)^n p \mathbb{P}(X \in A | X \in B). \end{aligned}$$

This factorises into a function of n and a function of A, and we read off the factors. [If this is not familiar, sum over n to get $\mathbb{P}(X_N \in A) = \mathbb{P}(X \in A | X \in B)$, or set $A = \mathbb{R}$ to get $\mathbb{P}(N = n) = (1 - p)^n p$, and identify the right-hand side as $\mathbb{P}(N = n)\mathbb{P}(X_N \in A)$.] □

Example 14 Let $X, X_1, X_2, \ldots \sim \text{Normal}(0, 1)$ and $B = [-1, 2]$. We obtain from tables or from `R` that

$$p = \mathbb{P}(X \in B) = \texttt{pnorm(2)} - \texttt{pnorm(-1)} \approx 0.8186.$$

We can simulate from the standard normal distribution conditioned to fall into B by:

1. Generate $X \sim \text{Normal}(0, 1)$.
2. If $X \in [-1, 2]$, go to 3., otherwise go to 1.
3. Return X.

The number of trials (iterations of steps 1. and 2.) is a random variable $N \sim \text{geom}(p)$. Recall the expectation of a geometric random variable:

$$\mathbb{E}(N) = \frac{1}{p} \approx 1.2216.$$

This is the expected number of trials, which indicates the efficiency of the algorithm – the fewer trials the quicker the algorithm (in the absence of other factors affecting efficiency).

3.3 Rejection

Now we are ready to pull threads together. The rejection algorithm applies "conditioning by repeated trials" to conditioning a random variable $V \sim \mathrm{Unif}(A_{Mh})$ to fall into A_f; here $A_{Mh} = \{(x, y) \in \mathbb{R} \times [0, \infty) : 0 \leq y \leq Mh(x)\}$. This is useful, if we have

- explicit probability density functions f and h,
- can simulate from the *proposal density* h, but not from the *target density* f,
- there is a (not too large) $M > 1$ with $A_f \subset A_{Mh}$, i.e. $f \leq Mh$.

Proposition 15 (Rejection) *Let f and h be densities such that $f \leq Mh$ for some $M \geq 1$. Then the algorithm*

1. generate a random variable X with density h and $U \sim \mathrm{Unif}(0, 1)$;
2. if $MUh(X) \leq f(X)$, go to 3., otherwise go to 1;
3. return X;

simulates a random variable X with density f.

Proof: Note that $Y = Uh(X)$ is uniformly distributed on $(0, h(X))$ as in Lemma 11. Hence, $(X, Y) = (X, Uh(X)) \sim \mathrm{Unif}(A_h)$, and $V = (X, MUh(X)) \sim \mathrm{Unif}(A_{Mh})$. Since

$$MUh(X) \leq f(X) \iff (X \in \mathbb{R} \text{ and } 0 \leq MUh(X) \leq f(X)) \iff V \in A_f,$$

and by Lemma 13, the algorithm here returns V conditioned to fall into A_f, which is $\mathrm{Unif}(A_f)$, or rather it returns the first component X that by Lemma 11 has density f. □

From Lemma 13, we obtain:

Corollary 16 In the algorithm of Proposition 15, the number of trials (iterations of 1. and 2.) is geom$(1/M)$. In particular, the number of trials has mean M. Furthermore, the number of trials is independent of the simulated random variable X.

Proof: Where we apply Lemma 13 in the proof of the proposition, we only exploit the second bullet point in Lemma 13. The first and third yield the statements of this corollary, because

$$p = \mathbb{P}(V \in A_f) = \frac{\mathrm{area}(A_f)}{\mathrm{area}(A_{Mh})} = \frac{1}{M}.$$

□

Example 17 The beta distribution Beta(α, β) has probability density function

$$f(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1} \qquad \text{for } 0 < x < 1,$$

where $\alpha > 0$ and $\beta > 0$ are parameters. For $\alpha \geq 1$ and $\beta \geq 1$, f is bounded, and we can choose the uniform density $h(x) = 1$ as proposal density. To minimise the expected number of trials, we calculate $M = \sup\{f(x)/h(x) : x \in (0,1)\} = \sup\{f(x) : x \in (0,1)\}$. With

$$f'(x) = 0 \iff (\alpha-1)(1-x) + (\beta-1)x = 0 \iff x = \frac{\alpha-1}{\beta+\alpha-2}$$

and a smooth density vanishing at one or both boundaries (unless $\alpha = \beta = 1$, which is Unif$(0,1)$ itself), this is where the supremum is attained:

$$M = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \left(\frac{\alpha-1}{\beta+\alpha-2}\right)^{\alpha-1} \left(\frac{\beta-1}{\alpha+\beta-2}\right)^{\beta-1}.$$

Note that $MUh(X) \leq f(X) \iff U \leq \dfrac{f(X)}{Mh(X)} \iff U \leq \dfrac{X^{\alpha-1}(1-X)^{\beta-1}}{M'}$, where $M' = \left(\dfrac{\alpha-1}{\beta+\alpha-2}\right)^{\alpha-1} \left(\dfrac{\beta-1}{\alpha+\beta-2}\right)^{\beta-1}$, sees the ratio of Gamma functions, the normalisation constant of f, disappear from the algorithm. Indeed, normalisation constants of f can always be avoided in the code of a rejection algorithm, since it only involves f/Mh, and maximising cf/h for any c gives a maximum cM, and $cf/cMh = f/Mh$. Figure 3.1 shows an implementation and sample simulation of this algorithm.

```
Geoffs_beta<-function(a=1,b=1) {
    #simulate X~Beta(a,b) variate, defaults to U(0,1)
    if (a<1 || b<1) stop(’a<1 or b<1’);
    M<-(a-1)^(a-1)*(b-1)^(b-1)*(a+b-2)^(2-a-b)
    finished<-FALSE
    while (!finished) {
        Y<-runif(1)
        U<-runif(1)
        accept_prob<-Y^(a-1)*(1-Y)^(b-1)/M
        finished<-(U<accept_prob)
    }
    X<-Y
X}
a<-1.5; b<-2.5;
n<-100000
X<-rep(NA,n)        #fill X with NA
for (i in 1:n) {
    X[i]<-Geoffs_beta(a,b)
}
```

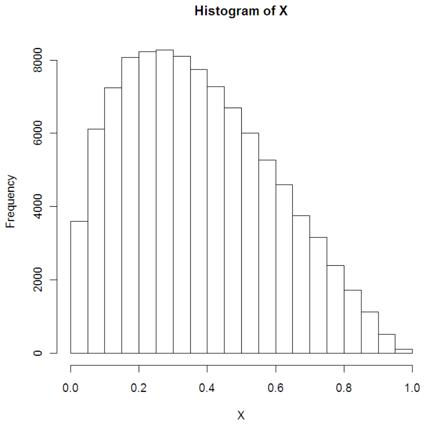


Figure 3.1: Simulation of Beta(α, β) with $\alpha = 1.5$ and $\beta = 2.5$

Lecture 4

Refining simulation methods

Reading: Ross Chapter 4
Further reading: Robert and Casella Section 2.2 and 2.3

4.1 More rejection

The rejection algorithm of Proposition 15 is in the most common framework of pdfs of one-dimensional distributions. In fact, the ideas equally apply on probability spaces $(E,\mathcal{E},\mu)$ and $(E,\mathcal{E},\nu)$ where $\mu(A)=\int_A g(x)\nu(dx)$ with bounded density g. Let us record here two instances of this.

Proposition 18 (Rejection II) *Let $f,h\colon\mathbb{R}^d\to[0,\infty)$ be probability density functions such that $f\le Mh$ for some $M\ge 1$. Then the algorithm*

1. generate a random vector $X=(X_1,\ldots,X_d)$ with density h and $U\sim\mathrm{Unif}(0,1)$;
2. if $MUh(X)\le f(X)$, go to 3., otherwise go to 1;
3. return X;

simulates a random vector X with density f.

Proof: The proof of Proposition 15 still applies, provided that Lemma 11 is also rewritten for d-dimensional X, which is straightforward. □

Proposition 19 (Rejection III) *Let $\mathcal{X}$ be a countable set and $\pi,\xi\colon\mathcal{X}\to[0,1]$ two probability mass functions such that $\pi\le M\xi$ for some $M\ge 1$. Then the algorithm*

1. generate a random variable X with probability mass function ξ and $U\sim\mathrm{Unif}(0,1)$;
2. if $MU\xi(X)\le\pi(X)$, go to 3., otherwise go to 1;
3. return X;

simulates a random variable X with probability mass function π.

Proof: Let us give a direct proof. For $p=\mathbb{P}(MU\xi(X)\le\pi(X))$, we have

$$\mathbb{P}(X=i|MU\xi(X)\le\pi(X))=\frac{\mathbb{P}(X=i,MU\xi(i)\le\pi(i))}{\mathbb{P}(MU\xi(X)\le\pi(X))}=\frac{\xi(i)\pi(i)/M\xi(i)}{p}=\frac{1}{pM}\pi(i).$$

Summing over i, we see that $p=1/M$ again. □

4.2 Random permutations

We have seen discrete uniform distributions as discretized $U \sim \text{Unif}(0,1)$, i.e. as $[kU] \sim \text{Unif}(\{0,\ldots,k-1\})$. A uniform random permutation of $\{1,\ldots,n\}$ is such that 1 is mapped to a uniform pick from $\{1,\ldots,n\}$, and there are many ways (essentially differing in their order) to sequentially map remaining numbers to remaining destinations. Here is one such algorithm.

1. Let $P_j = j$, $j = 1,\ldots,n$, and set $k = n$;
2. for k from n downto 2, generate $U \sim \text{Unif}(0,1)$, let $I = 1 + [kU]$ and interchange P_I and P_k.
3. Return the random permutation X, where $X(j) = P_j$, $j = 1,\ldots,n$.

We can now use the uniform distribution as proposal distribution in a rejection algorithm.

Example 20 (Gibbs distributions) Let $\mathcal{X}$ be the set of permutations of n. Let ξ be the uniform distribution on $\mathcal{X}$ and π a Gibbs distribution

$$\pi(\eta) = \frac{1}{Z_n} \exp\left(- \sum_{\gamma \text{ cycle of } \eta} \lambda_{\#\gamma} \right), \qquad \text{where } \#\gamma \text{ is the cycle length,}$$

for some parameters $\lambda_1,\ldots,\lambda_n \in \mathbb{R}$. The choice of parameters makes the system favour certain configurations and not others. Since Z_n is usually difficult to work out, we calculate

$$M = \max\{Z_n\pi(\eta), \eta \in \mathcal{X}\} = \min\left\{ \sum_{\gamma \text{ cycle of } \eta} \lambda_{\#\gamma}, \quad \eta \in \mathcal{X} \right\}$$

and use the rejection algorithm

1. generate a random variable X with probability mass function ξ and $U \sim \text{Unif}(0,1)$;
2. if $MU\xi(X) \leq \exp\left(- \sum_{\gamma \text{ cycle of } X} \lambda_{\#\gamma} \right)$, go to 3., otherwise go to 1;
3. return X;

to simulate from π.

More generally, one can define Gibbs distributions on finite sets $\mathcal{X}$ of possible states of a system giving rise to some structure of components or other local features (such as the cycle structure in the example). Such models are popular in statistical physics and include spatial spin systems (e.g. Ising model), mean-field models such as random partitions, graphs (e.g. trees) etc. Gibbs distributions often reflect some physical constraints on components or on that make certain states unlikely.

4.3 Discrete distributions and random parameters

For discrete distributions, inversion can be carried out as a sequential algorithm:

Example 21 (Poisson) The Poisson distribution with parameter $r > 0$ has probability mass function

$$p_n = \frac{r^n}{n!}e^{-r}, \qquad r \geq 0.$$

Note that $p_n = rp_{n-1}/n$. The inversion method based on $U \sim \text{Unif}(0,1)$ sets

$$X = n \iff F(n-1) = \sum_{k=0}^{n-1} p_k < U \leq \sum_{k=0}^{n} p_k = F(n).$$

Therefore, $X = \inf\{n \geq 0 : U < F(n)\}$. This leads to the following algorithm:

1. generate $U \sim \text{Unif}(0,1)$ and set $n = 0$, $p = e^{-r}$, $F = p$;
2. if $U < F$, set $X = n$ and go to 3., otherwise set $n := n+1$, $p := rp/n$, $F := F + p$ and repeat 2.;
3. return X.

The Poisson distribution has the property $\text{Var}(X) = \mathbb{E}(X) = r$, if $X \sim \text{Poisson}(r)$. A rich source of "overdispersed" distributions with $\text{Var}(X) > \mathbb{E}(X)$ is obtained by randomising the parameter:

Example 22 (Overdispersed Poisson) Let R have a distribution on $[0, \infty)$ that we can simulate. Consider the following algorithm

1. generate R and, independently, $U \sim \text{Unif}(0,1)$ and set $n = 0$, $p = e^{-r}$, $F = p$;
2. if $U < F$, set $X = n$ and go to 3., otherwise set $n := n+1$, $p := Rp/n$, $F := F + p$ and repeat 2.;
3. return X.

This trivial extension of the Poisson generator yields the following. In the case where R is discrete with $\mathbb{P}(R = r) = \pi_r$, $r \geq 0$, we obtain

$$\begin{aligned}\mathbb{P}(X = n) &= \sum_{r \geq 0} \mathbb{P}(X = n, R = r) = \sum_{r \geq 0} \mathbb{P}(R = r)\mathbb{P}(X = n | R = r) = \sum_{r \geq 0} \mathbb{P}(R = r)\frac{r^n}{n!}e^{-r} \\ &= \mathbb{E}\left(\frac{R^n}{n!}e^{-R}\right),\end{aligned}$$

by the definition of $\mathbb{E}(g(R))$, where here $g(r) = r^n e^{-r}/n!$. With appropriate care in the conditioning, this last expression holds in the general case of any $[0, \infty)$-valued random variable R.

Intuitively $\text{Var}(X) > \mathbb{E}(X)$ because $\mathbb{E}(X) = \mathbb{E}(\mathbb{E}(X|R)) = \mathbb{E}(R)$, (this can be written more carefully like the conditioning above), while for variances, consider the spread around $\mathbb{E}(R)$: first R spreads around $\mathbb{E}(R)$ and then X spreads around R and these two sources of spread add up. More formally, we calculate

$$\text{Var}(X) = \mathbb{E}(\mathbb{E}(X^2|R)) - (\mathbb{E}(X))^2 = \mathbb{E}(R + R^2) - (\mathbb{E}(R))^2 = \mathbb{E}(R) + \text{Var}(R) \geq \mathbb{E}(X)$$

with equality if and only if $\text{Var}(R) = 0$, i.e. R is deterministic.

Lecture 5

Importance sampling

Reading: Ross Section 7.1, Ripley Sections 5.1 and 5.2
Further reading: Ross Section 8.6

So far, we have developed many answers, applicable in different situations, to one question, essentially: given a distribution, how do we simulate from it? I.e., how can we generate a sample $X_1, \dots, X_n$ from this distribution? In practice, this will often just be a tool to help us answer more specific questions, in situations when analytic answers are not (or not easily) available:

- What is the mean of this distribution, $\mathbb{E}(X)$?
- How likely is it that this distribution produces values above a threshold, $\mathbb{P}(X > c)$?
- For some utility function u, what is the expected utility, $\mathbb{E}(u(X))$?

Fundamentally, these are all instances of the following: estimate $\mathbb{E}(\phi(X))$, from simulated data, respectively with $\phi(x) = x$, $\phi(x) = 1_{(c,\infty)}(x)$, $\phi = u$. With $\phi\colon \mathcal{X} \to \mathbb{R}^d$, we can capture $\mathcal{X}$-valued random variables for quite arbitrary spaces $\mathcal{X}$, e.g. large discrete spaces or $\mathbb{R}^m$. However, let us first think $\mathcal{X} = \mathbb{R}$ and $d = 1$.

5.1 Monte-Carlo integration

We know how to estimate an unknown mean in the context of parameter estimation.

Proposition 23 *Let ϕ be such that $\theta = \mathbb{E}(\phi(X))$ exists and $X_1, \dots, X_n$ a sample of independent copies of X. Then*

$$\widehat{\theta} = \frac{1}{n} \sum_{j=1}^{n} \phi(X_j)$$

is an unbiased and consistent estimator of θ. If $\mathrm{Var}(\phi(X))$ exists, then the mean squared error of $\widehat{\theta}$ is

$$\mathbb{E}\left(\left(\widehat{\theta} - \theta\right)^2\right) = \mathrm{Var}\left(\widehat{\theta}\right) = \frac{\mathrm{Var}(\phi(X))}{n},$$

where $\mathrm{Var}(\phi(X))$ can be estimated by the (unbiased) sample variance

$$S^2_{\phi(X)} = \frac{1}{n-1} \sum_{j=1}^{n} \left(\phi(X_j) - \widehat{\theta}\right)^2.$$

Proof: With $Y_j = \phi(X_j)$, this is Mods Statistics (and further Part A Probability and Statistics). The following is a reminder. Unbiasednes holds by linearity of expectations

$$\mathbb{E}\left(\widehat{\theta}\right) = \frac{1}{n}\sum_{j=1}^{n}\mathbb{E}(\phi(X_j)) = \mathbb{E}(\phi(X)).$$

The variance expression follows from the variance rules $\mathrm{Var}(A+B) = \mathrm{Var}(A) + \mathrm{Var}(B)$ for independent A and B, and $\mathrm{Var}(cA) = c^2\mathrm{Var}(A)$:

$$\mathrm{Var}\left(\widehat{\theta}\right) = \frac{1}{n^2}\sum_{j=1}^{n}\mathrm{Var}(\phi(X_j)) = \frac{\mathrm{Var}(\phi(X))}{n}.$$

(Weak or strong) consistency is a consequence of the (Weak or Strong) Law of Large Numbers applied to $Y_j = \phi(X_j)$ – note that $\mathbb{E}(Y) = \mathbb{E}(\phi(X))$ is assumed to exist:

$$\widehat{\theta} = \frac{1}{n}\sum_{j=1}^{n}\phi(X_j) = \overline{Y} := \frac{1}{n}\sum_{j=1}^{n}Y_j \longrightarrow \mathbb{E}(Y), \quad \text{(in probability or a.s.) as } n \to \infty.$$

With $Y_j = \phi(X_j)$, the final statement is just unbiasedness of the sample variance of $Y_1, \ldots, Y_n$, which justifies the denominator $n-1$; we expand the squares and apply $\mathbb{E}(Y^2) = \mathrm{Var}(Y) + (\mathbb{E}(Y))^2$ to get

$$\mathbb{E}(S^2_{\phi(X)}) = \frac{1}{n-1}\left(\sum_{j=1}^{n}Y_j^2 - \overline{Y}^2\right) = \frac{\mathrm{Var}(Y) + \theta^2 - n^{-2}(n\mathrm{Var}(Y) + n^2\theta^2)}{n-1} = \mathrm{Var}(\phi(X)).$$

□

The variance is useful to decide on the size of the sample that we need to simulate to achieve a given level of accuracy. A conservative estimate can be obtained by Chebychev's inequality:

$$\mathbb{P}\left(\left|\widehat{\theta} - \theta\right| > \frac{c}{\sqrt{n}}\right) \le \frac{\mathrm{Var}(\widehat{\theta})}{c^2/n} = \frac{\mathrm{Var}(\phi(X))}{c^2} \approx \frac{S^2_{\phi(X)}}{c^2}.$$

A more realistic estimate follows from the Central Limit Theorem: for large n,

$$\frac{\widehat{\theta} - \theta}{\sqrt{\mathrm{Var}(\phi(X))/n}} \approx \mathrm{Normal}(0,1) \Rightarrow \mathbb{P}\left(\left|\widehat{\theta} - \theta\right| > c\sqrt{\frac{\mathrm{Var}(\phi(X))}{n}}\right) \approx 2(1 - \Phi(c)).$$

This can be used to not just give an estimate but an approximate $(1-\alpha)100\%$-confidence interval for θ, choosing $c = c_\alpha$ such that $2(1 - \Phi(c_\alpha)) = \alpha$:

$$\left(\widehat{\theta} - c_\alpha\sqrt{\frac{\mathrm{Var}(\phi(X))}{n}}, \widehat{\theta} + c_\alpha\sqrt{\frac{\mathrm{Var}(\phi(X))}{n}}\right) \approx \left(\widehat{\theta} - c_\alpha\frac{S_{\phi(X)}}{\sqrt{n}}, \widehat{\theta} + c_\alpha\frac{S_{\phi(X)}}{\sqrt{n}}\right).$$

Therefore, if we want to be $(1-\alpha)100\%$ confident that our estimate of θ is within δ of the true value of θ, we start with a moderate number of n and keep simulating more samples until $c_\alpha S_{\phi(X)}/\sqrt{n} < \delta$. If $\mathrm{Var}(\phi(X)) < \infty$, this algorithm terminates, because $S_{\phi(X)}$, which depends on n, will converge to $\sqrt{\mathrm{Var}(\phi(X))}$: by the Laws of Large Numbers

$$\frac{1}{n-1}S^2_{\phi(X)} = \frac{n}{n-1}\frac{1}{n}\sum_{j=1}^{n}(\phi(X_j))^2 - \left(\widehat{\theta}\right)^2 \longrightarrow \mathbb{E}\left((\phi(X_j))^2\right) - \theta^2 = \mathrm{Var}(\phi(X)).$$

The estimator $\widehat{\theta}$ is called *Monte-Carlo estimator* of θ. We can write the corresponding *Monte-Carlo algorithm* simply as:

1. Simulate $X_1, \dots, X_n$ from the given distribution.
2. Return $n^{-1}(\phi(X_1) + \cdots + \phi(X_n))$.

So the key in any such algorithm is, of course, step 1., which is often non-trivial – Monte-Carlo methods are most useful when analytic methods fail. We will, however, focus on simpler examples here. Particularly when X is continuously distributed (on $\mathbb{R}^d$) with some density f, the estimation is also called *Monte-Carlo integration*, but also more generally (appealing to the notion of integrals with respect to more general measures), because it is, fundamentally, a way to approximate the integral

$$\mathbb{E}(\phi(X)) = \int \phi(x) f(x) dx.$$

5.2 Variance reduction via importance sampling

In discussing Monte-Carlo integration, we drew on similarities to parameter estimation in data analysis, where we use the sample mean to estimate a population mean, and this is virtually always the best we can do. So it is perhaps surprising that we can do better here. The main difference is that in data analysis we are given the sample, whereas the trick here is to sample $Z_1, \dots, Z_n$ from a different distribution to estimate $\mathbb{E}(\phi(X))$:

Proposition 24 *Given two probability density functions f and h and a function ϕ such that $h(x) = 0$ only if $f(x)\phi(x) = 0$, and such that for $X \sim f$,*

$$\theta = \mathbb{E}(\phi(X)) = \int_{\mathbb{R}^d} \phi(x) f(x) dx$$

exists, consider a sample $Z_1, \dots, Z_n \sim h$. Then the importance sampling estimator

$$\widehat{\theta}^{\mathrm{IS}} = \frac{1}{n} \sum_{j=1}^{n} \phi(Z_j) \frac{f(Z_j)}{h(Z_j)}$$

is an unbiased and consistent estimator of θ. If $\mathrm{Var}(\phi(Z)f(Z)/h(Z))$ *exists, then*

$$\mathbb{E}\left(\left(\widehat{\theta}^{\mathrm{IS}} - \theta\right)^2\right) = \mathrm{Var}\left(\widehat{\theta}^{\mathrm{IS}}\right) = \frac{1}{n}\mathrm{Var}\left(\phi(Z)\frac{f(Z)}{h(Z)}\right) = \frac{1}{n}\int_{\mathbb{R}^d} \phi^2(x)\frac{f^2(x)}{h(x)} dx - \frac{1}{n}\theta^2.$$

Proof: This result follows from Proposition 23, considering $\phi^{\mathrm{IS}}(x) = \phi(x)f(x)/h(x)$ with

$$\mathbb{E}\left(\phi^{\mathrm{IS}}(Z)\right) = \int_{\mathbb{R}^d} \phi^{\mathrm{IS}}(x) h(x) dx = \int_{\mathbb{R}^d} \phi(x)\frac{f(x)}{h(x)} h(x) dx = \int_{\mathbb{R}^d} \phi(x) f(x) dx = \mathbb{E}(\phi(X)) = \theta.$$

For the integral expression of the variance note that $\theta = \mathbb{E}(\phi(Z)f(Z)/h(Z))$ and hence

$$\mathrm{Var}\left(\phi(Z)\frac{f(Z)}{h(Z)}\right) = \mathbb{E}\left(\phi^2(Z)\frac{f^2(Z)}{h^2(Z)}\right) - \theta^2 = \int_{\mathbb{R}^d} \phi^2(x)\frac{f^2(x)}{h^2(x)} h(x) dx - \theta^2.$$

□

Note that the mean squared error of the *importance sampling estimator* $\widehat{\theta}^{\mathrm{IS}}$ is small compared to the one of the Monte-Carlo estimator $\widehat{\theta}$, if $f(x)/h(x)$ is small where $\phi^2(x)f(x)$ is large. We can either use this to define h with these properties, or we can more formally minimise the mean squared error within a family of densities h that we can simulate.

The reason for the name importance sampling is that some x-values (those where $\phi(x)f(x)$ is relatively large) are more important than others (e.g. where $\phi(x) = 0$, they are irrelevant). Successful importance sampling generates more of the important sample values and adjusts by giving appropriate smaller weights to such values, as is best seen in examples. We can identify these weights $w_j = f(Z_j)/h(Z_j)$, the relative likelihood of Z_j being selected from f versus h.

5.3 Example: Importance sampling for Normal tails

Suppose we want to use importance sampling to estimate $\theta = \mathbb{P}(X > c) = \mathbb{E}(1_{\{X>c\}})$ for $X \sim \text{Normal}(0, \sigma^2)$ and $c > 3\sigma$. The Monte-Carlo estimator

$$\widehat{\theta} = \frac{1}{n}\sum_{j=1}^{n} 1_{\{X_j>c\}} = \frac{1}{n}\#\{j \in \{1, \ldots, n\} : X_j > c\}$$

will be quite poor, because the vast majority of sample values is completely irrelevant, only the very few that exceed c matter (well, it only matters onto which side of c they fall in either case, but the imbalance due to the very small probability to fall onto one of the two sides means that the variance is large compared to the probability estimated). We now use importance sampling to see more values above c. We use

$$h_\mu(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

Then $f(x) = h_0(x)$ and the weights will be

$$\frac{f(x)}{h_\mu(x)} = \exp\left(-\frac{x^2 - (x-\mu)^2}{2\sigma^2}\right) = \exp\left(\frac{\mu(\mu - 2x)}{2\sigma^2}\right),$$

that is, for $\mu > 0$, sample values above $\mu/2$ are sampled more often under h_μ than under $f = h_0$ and hence given a weight less than 1, while values below $\mu/2$ are sampled less often under h_μ and given a higher weight. The importance sampling estimator for $\mu \in \mathbb{R}$ is

$$\widehat{\theta}_\mu^{IS} = \frac{1}{n}\sum_{j=1}^{n} 1_{\{Z_j>c\}} \frac{f(Z_j)}{h_\mu(Z_j)} = \frac{1}{n}\sum_{j=1}^{n} 1_{\{Z_j>c\}} \exp\left(\frac{\mu(\mu - 2Z_j)}{2\sigma^2}\right).$$

In a simulation of $n = 10,000,000$ samples for $\sigma = 1$ and $c = 3$, we found sample standard deviations as in the following table, so we see that $\mu \approx 3.15$ gives the best accuracy, three significant digits estimating $\mathbb{P}(X > 3)$ by 0.00135, whereas only the first digit was significant for the Monte-Carlo estimator.

μ	standard deviation	estimate	lower 95%CI	upper 95%CI
0.00	11.5827300e-06	1.343400e-03	1.320698e-03	1.366102e-03
1.00	2.9066300e-06	1.352702e-03	1.347005e-03	1.358398e-03
2.00	1.1767060e-06	1.351572e-03	1.349266e-03	1.353878e-03
3.00	0.7855584e-06	1.350020e-03	1.348480e-03	1.351559e-03
3.10	0.7797804e-06	1.349173e-03	1.347645e-03	1.350701e-03
3.15	0.7791837e-06	1.349363e-03	1.347835e-03	1.350890e-03
3.20	0.7793362e-06	1.348548e-03	1.347020e-03	1.350075e-03
4.00	0.9765308e-06	1.348453e-03	1.346539e-03	1.350367e-03

The confidence intervals are also displayed in Figure 5.1.

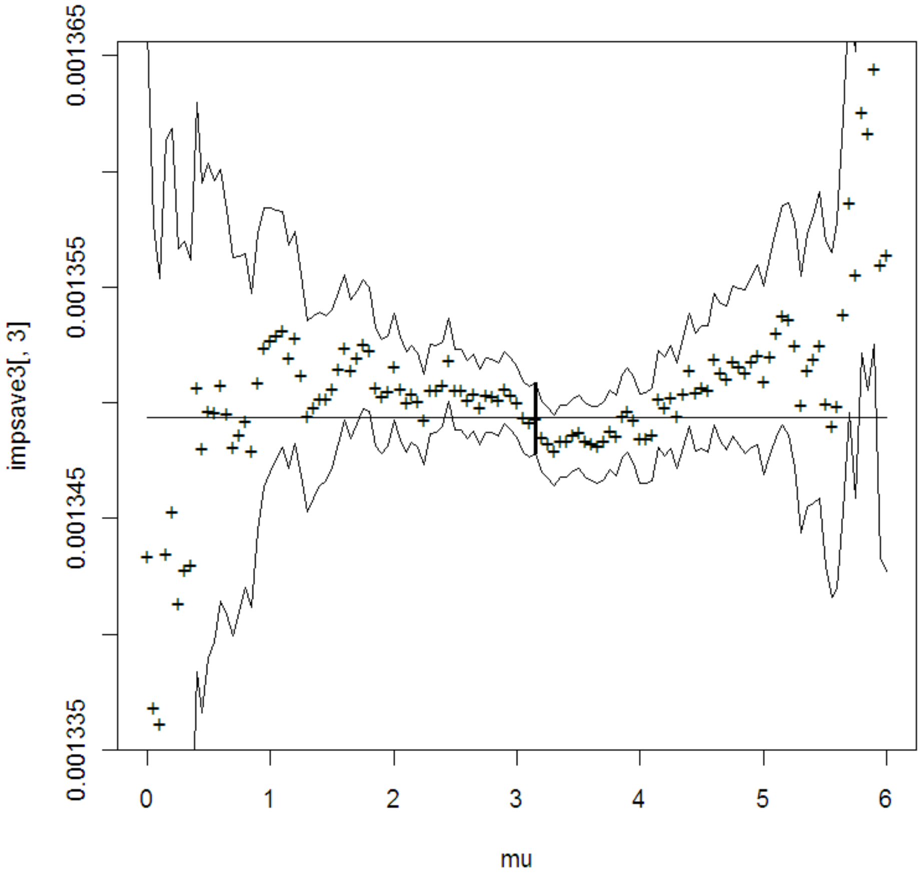


Figure 5.1: Confidence intervals for $\mathbb{P}(X > 3)$ as $0 \leq \mu \leq 6$, with $\mu = 3.15$ emphasized.

Going a bit further for $c = 4.5$, Monte-Carlo does not find any significant digits; importance sampling still finds two significant digits $\mathbb{P}(X > 4.5) \approx 0.0000034$:

μ	standard deviation	estimate	lower 95%CI	upper 95%CI
0.0	556.775600e-09	3.100000e-06	2.008740e-06	4.191260e-06
4.5	2.423704e-09	3.397638e-06	3.392887e-06	3.402388e-06
4.6	2.414115e-09	3.394935e-06	3.390203e-06	3.399666e-06
4.7	2.419696e-09	3.393120e-06	3.388377e-06	3.397862e-06

Here is the R code we used:

```
# Importance sampling for P(X>c) for N(0,1) based on N(mu,1)
# function imp returns entries for a row of our tables

imp<-function(mu=0,c=0,n=1000000,X=rnorm(n)){
     IS<-mean((mu+X>c)*exp(mu*(mu-2*(mu+X))/2))
     standev<-sd((mu+X>c)*exp(mu*(mu-2*(mu+X))/2))/sqrt(n)
     c(mu,standev,IS,IS-qnorm(0.975)*standev,IS+qnorm(0.975)*standev)
}
```

```
n<-10000000   #we fix n and X in advance
X<-rnorm(n)   #and use the same random numbers for all estimates

c<-3
imp(0,c,n,X)
imp(1,c,n,X)
imp(2,c,n,X)
imp(3,c,n,X)
imp(3.1,c,n,X)
imp(3.15,c,n,X)  #we find the (near-)optimal value of mu slightly above c
imp(3.2,c,n,X)
imp(4,c,n,X)

c<-4.5
imp(0,c,n,X)
imp(4.5,c,n,X)
imp(4.6,c,n,X)   #we find the (near-)optimal value of mu slightly above c
imp(4.7,c,n,X)

#To plot confidence intervals as mu varies

mu<-(0:120)/20                 #choose range from 0 to 6 in steps of 0.05
impsave3<-matrix(1:605,121,5)  #initialize impsave3, to save imp for c=3

for (i in 1:121){impsave3[i,]<-imp(mu[i],c,n,X)} #save imp in impsave3

plot(mu,impsave3[,3],pch="+")                        #estimates
lines(mu,impsave3[,4])                               #lower CI bound
lines(mu,impsave3[,5])                               #upper CI bound
lines(c(3.15,3.15),c(impsave3[64,4],impsave3[64,5])) #emphasize mu=3.15
lines(c(3.14,3.14),c(impsave3[64,4],impsave3[64,5])) #line too thin
lines(c(3.16,3.16),c(impsave3[64,4],impsave3[64,5])) #add a bit more
lines(c(0,6),c(impsave3[64,3],impsave3[64,3]))       #horizontal line
```

Lecture 6

Markov chain Monte-Carlo

Reading: Norris Section 5.5, Ross Sections 10.1-10.2
Further reading: Norris Chapter 1, Robert and Casella Sections 6.1-6.2

6.1 Simulation of Markov chains

Let $\mathcal{X}$ be a countable state space suitably enumerated such that we can think of a vector $\lambda = (\lambda_i, i \in \mathcal{X})$ as a row vector of initial probabilities $\mathbb{P}(X_0 = i) = \lambda_i$ – we suppose that $\lambda_i \geq 0$, $i \in \mathcal{X}$ and $\sum_{i \in \mathcal{X}} \lambda_i = 1$. For a transition matrix $P = (p_{i,j}, i, j \in \mathcal{X})$ – non-negative entries with unit row sums – we consider the (λ, P)-Markov chain $(X_0, \ldots, X_n)$ with distribution

$$\mathbb{P}(X_0 = i_0, \ldots, X_n = i_n) = \lambda_{i_0} \prod_{j=1}^{n} p_{i_{j-1}, i_j}, \qquad i_0, \ldots, i_n \in \mathcal{X}.$$

In other words, the probability of a path $i_0 \to i_1 \to \cdots \to i_{n-1} \to i_n$ is the probability λ_{i_0} of initial value i_0 successively multiplied into all one-step transition probabilities p_{i_{j-1}, i_j} from i_{j-1} to i_j.

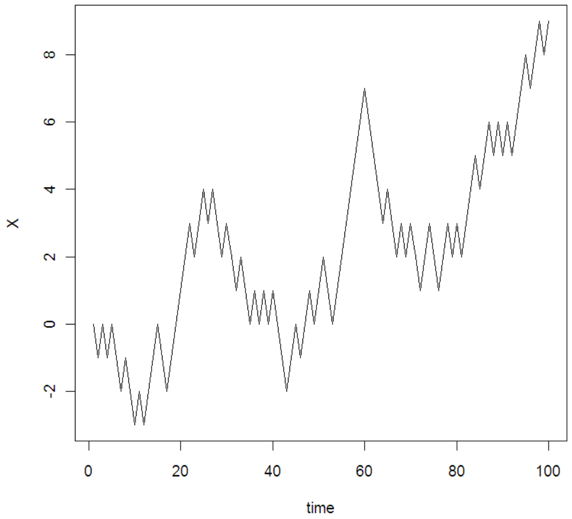
Each row $p_{i,\bullet} = (p_{i,j}, j \in \mathcal{X})$, of P, $i \in \mathcal{X}$, is a probability distribution on $\mathcal{X}$, and if we can simulate (from the initial distribution λ and) from each of these rows $p_{i,\bullet}$, $i \in \mathcal{X}$, e.g. using a sequential inversion algorithm as in Section 4.3, we can simulate the chain:

1. Generate $X_0 \sim \lambda$.
2. For k from 1 to n, generate $X_k \sim p_{X_{k-1},\bullet}$.
3. Return $(X_0, \ldots, X_n)$.

Example 25 (Simple random walk) A particle starting from 0 (set $\lambda_0 = 1$) is moving on $\mathcal{X} = \mathbb{Z}$, up with probability $p \in (0, 1)$, down with probability $1 - p$:

$$p_{i,i+1} = p, \qquad p_{i,i-1} = 1 - p, \qquad p_{i,j} = 0 \quad \text{otherwise.}$$

Figure 6.1 simulates n steps of this Markov chain. Recall that simple random walk is irreducible, but 2-periodic, and transient except when $p = 1/2$ in which case it is null-recurrent. In any case, there is no stationary distribution and hence no convergence to stationarity of $\mathbb{P}(X_n = k)$ as $n \to \infty$. Instead, it can be shown that $\mathbb{P}(X_n = k) \to 0$ as $n \to \infty$.



```
p<-0.5
n<-100
X<-rep(NA,n)

X[1]<-0
for (j in 2:n) {
  X[j]<-X[j-1]+2*(runif(1)<=p)-1
}

time<-1:n
plot(time,X,type="l")
```

Figure 6.1: Simple symmetric random walk

6.2 Monte-Carlo algorithm

The Monte-Carlo algorithm of Lecture 5, for the estimation of $\theta = \mathbb{E}(\phi(X))$, was based on *independent* $X_1, \ldots, X_n$ with the same distribution as X, setting

$$\widehat{\theta} = \frac{1}{n}\sum_{j=1}^{n} \phi(X_j) \to \mathbb{E}(\phi(X)) \qquad \text{in probability (or a.s.) as } n \to \infty.$$

For irreducible Markov chains $(X_0, \ldots, X_{n-1})$ with stationary distribution $\pi = \pi P$, the Ergodic Theorem makes a very similar statement

$$\frac{1}{n}\sum_{j=0}^{n-1} \phi(X_j) \to \mathbb{E}(\phi(X)) = \sum_{i\in\mathcal{X}} \phi(i)\pi_i, \qquad \text{in probability (or a.s.) as } n \to \infty,$$

where $X \sim \pi$. If furthermore $X_0 \sim \pi$, then

$$\mathbb{P}(X_1 = j) = \sum_{i\in\mathcal{X}} \mathbb{P}(X_1 = j | X_0 = i)\pi_i = (\pi P)_j = \pi_j, \qquad \text{so } X_1 \sim \pi,$$

and, inductively, $X_0, \ldots, X_{n-1}$ are identically distributed, but not (in general) independent. Writing $p_{i,\bullet}$ for the ith row of P, the associated *Monte-Carlo algorithm* is:

1. Generate $X_0 \sim \lambda$.
2. For k from 1 to $n-1$, generate $X_k \sim p_{X_{k-1},\bullet}$.
3. Return $n^{-1}(\phi(X_0) + \cdots + \phi(X_{n-1}))$.

Ideally, $\lambda = \pi$, which gives an unbiased estimator. In practice, it is often difficult to simulate from π, so other λ are often taken – the bias disappears asymptotically.

When applying a Markov chain Monte-Carlo algorithm, the quantity of interest is usually not the Markov chain, but (some weighted sum over) stationary probabilities. Indeed, the Markov chain Monte-Carlo method is a method to approximate a distribution π or weighted sum $\sum_{i\in\mathcal{X}} \phi(i)\pi_i$, which involves *finding/choosing* a suitable Markov chain.

Example 26 (Moran model with mutation) A population of size $N \geq 2$ consists of two types of individuals, initially $X_0 = N/2$ of type 1 and $N/2$ of type 2. At any time $n \geq 1$, two individuals are picked at random, one to give birth to a new individual, of the same type with probability $1-p$, and of the other type with the probability p; the other individual dies. Let X_n be the number of type-1 individuals at time n. This is a Markov chain with $\mathcal{X} = \{0, \ldots, N\}$, $\lambda_{N/2} = 1$, non-zero transitions

$$p_{k,k+1} = \frac{k(N-k)(1-p) + (N-k)^2 p}{N^2}, \quad p_{k,k-1} = \frac{k^2 p + k(N-k)(1-p)}{N^2},$$

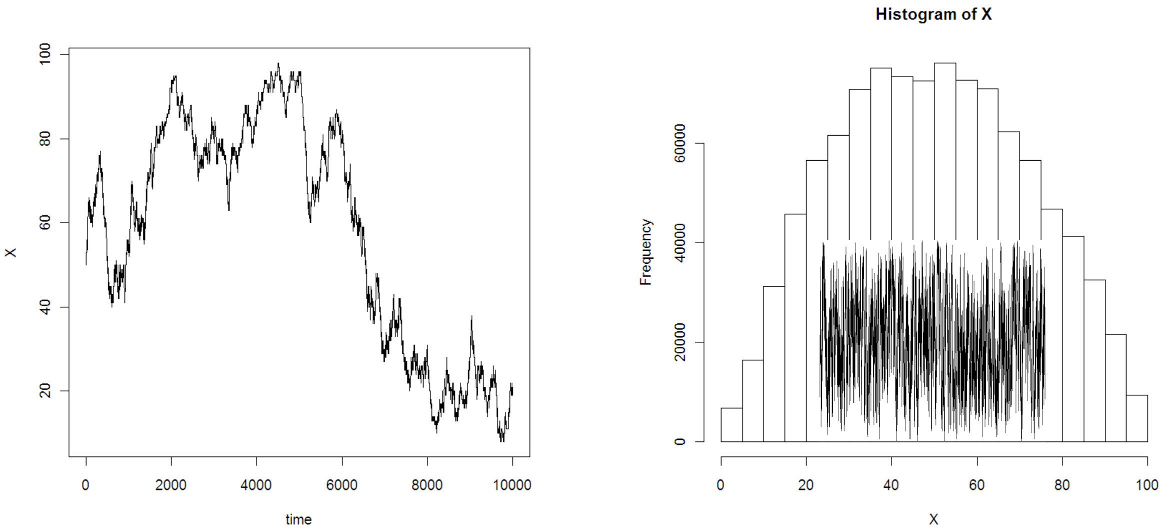
$$p_{k,k} = \frac{k^2(1-p) + (N-k)^2(1-p) + 2k(N-k)p}{N^2}.$$

The left-hand plot of Figure 6.2 shows a simulation path of length $n = 10,000$.

This model is clearly irreducible and aperiodic on a finite state space, so it converges to stationarity. We could work out the stationary distribution π analytically, but let us here use simulation techniques to display the stationary distribution. Of the two theorems, the Convergence Theorem $\mathbb{P}(X_n = k) \to \pi_k$ and the Ergodic Theorem

$$n^{-1}\#\{j \in \{1, \ldots, n\} : X_j = k\} \to \pi_k \qquad \text{a.s.}$$

the latter is more efficient – the Monte-Carlo method. We simulate $n = 1,000,000$ steps and plot a histogram, see the right-hand plot of Figure 6.2, where we also inserted the simulation path to demonstrate how the chain has explored the state space over time.



```
N<-100; p<-0.02; n<-10000; # n<-1000000 for stationary distribution
X<-rep(NA,n); X[1]<-N/2
for (j in 2:n) {
    k<-X[j-1]; X[j]<-X[j-1]; U<-runif(1)
    if (U<(k*(N-k)*(1-p)+(N-k)^2*p)/N^2) {X[j]<-X[j-1]+1}
    else {if (1-U<(k^2*p+k*(N-k)*(1-p))/N^2) {X[j]<-X[j-1]-1}}}
time<-1:n; plot(time,X,type="l")
hist(X)
```

Figure 6.2: Simulations of the Moran model with mutation

6.3 Reversible Markov chains

A Markov chain is called *reversible* if for all $n \geq 1$ and $i_0, i_1, \ldots, i_{n-1}, i_n \in \mathcal{X}$, we have

$$\mathbb{P}(X_0{=}i_0, X_1{=}i_1, \ldots, X_{n-1}{=}i_{n-1}, X_n{=}i_n) = \mathbb{P}(X_0{=}i_n, X_1{=}i_{n-1}, \ldots, X_{n-1}{=}i_1, X_n{=}i_0).$$

In particular, this implies that for $n = 1$ and $i, j \in \mathcal{X}$,

$$\lambda_i p_{i,j} = \mathbb{P}(X_0 = i, X_1 = j) = \mathbb{P}(X_0 = j, X_1 = i) = \lambda_j p_{j,i}.$$

These relations are called *detailed balance equations*. Summing over i, we see that

$$(\lambda P)_j = \sum_{i \in \mathcal{X}} \lambda_i p_{i,j} = \sum_{i \in \mathcal{X}} \lambda_j p_{j,i} = \lambda_j \sum_{i \in \mathcal{X}} p_{j,i} = \lambda_j, \qquad \text{i.e. } \lambda \text{ stationary.}$$

Furthermore, if the detailed balance equations hold, the chain is already reversible:

$$\begin{aligned}\mathbb{P}(X_0{=}i_0, X_1{=}i_1, \ldots, X_{n-1}{=}i_{n-1}, X_n{=}i_n) &= \lambda_{i_0} p_{i_0,i_1} \cdots p_{i_{n-1},i_n} = p_{i_1,i_0} p_{i_2,i_1} \cdots p_{i_n,i_{n-1}} \lambda_{i_n} \\ &= \mathbb{P}(X_0{=}i_n, X_1{=}i_{n-1}, \ldots, X_{n-1}{=}i_1, X_n{=}i_0).\end{aligned}$$

Example 27 Let $\mathcal{X} = \{0, \ldots, N\}$. Any transition matrix P with $p_{i,i-1} > 0$ and $p_{i-1,i} > 0$, $i \in \{1, \ldots, n\}$ and $p_{i,j} = 0$ for $|i - j| \geq 2$ gives rise to a reversible Markov chain. Specifically, note that $\pi_i p_{i,j} = \pi_j p_{j,i}$ holds trivially for $i = j$ and $|i - j| \geq 2$. The only non-trivial detailed balance equations are $\pi_i p_{i,i-1} = \pi_{i-1} p_{i-1,i}$, $i \in \{1, \ldots, n\}$. A straightforward induction shows that they are equivalent to

$$\pi_j = \pi_0 \prod_{i=1}^{j} \frac{p_{i-1,i}}{p_{i,i-1}}, \qquad j \in \mathcal{X},$$

and the normalisation condition $\sum_{j \in \mathcal{X}} \pi_j = 1$ yields a unique solution π. Note further, that this chain is irreducible and so π is the unique stationary distribution of this Markov chain.

Not every Markov chain that has a stationary distribution is reversible, but reversibility is easy to check and useful to calculate a stationary distribution (certainly easier than solving $\pi = \pi P$, usually). We will construct reversible Markov chains with a given stationary distribution.

6.4 The Metropolis-Hastings algorithm

Let $\mathcal{X}$ be a finite state space and π be a distribution on $\mathcal{X}$ with $\pi_i > 0$ for all $i \in \mathcal{X}$. For any initial distribution λ on $\mathcal{X}$ and irreducible transition matrix $R = (r_{i,j}, i, j \in \mathcal{X})$ on $\mathcal{X}$ with $r_{i,j} = 0 \iff r_{j,i} = 0$, consider the following *Metropolis-Hastings algorithm*:

1. Let $X_0 \sim \lambda$ and $k = 0$.
2. Set $k := k + 1$. Generate independent $Y_k \sim r_{X_{k-1},\bullet}$ and $U \sim \text{Unif}(0, 1)$.
3. If $U \pi_{X_{k-1}} r_{X_{k-1},Y_k} \leq \pi_{Y_k} r_{Y_k,X_{k-1}}$, set $X_k = Y_k$, otherwise set $X_k = X_{k-1}$.
4. If $k < n$, go to 2., otherwise return $(X_0, \ldots, X_n)$.

We will show that this algorithm simulates a reversible Markov chain with stationary distribution π.

Lecture 7

The Metropolis-Hastings algorithm

Reading: Norris Section 5.5

7.1 The basic Metropolis-Hastings algorithm

Recall the basic Metropolis-Hastings setting: a finite state space $\mathcal{X}$ and a distribution π on $\mathcal{X}$ with $\pi_i > 0$ for all $i \in \mathcal{X}$; for any initial distribution λ on $\mathcal{X}$ and irreducible transition matrix $R = (r_{i,j}, i, j \in \mathcal{X})$ on $\mathcal{X}$ with $r_{i,j} = 0 \iff r_{j,i} = 0$, consider:

1. Let $X_0 \sim \lambda$ and $k = 0$.
2. Set $k := k + 1$. Generate independent $Y_k \sim r_{X_{k-1},\bullet}$ and $U \sim \text{Unif}(0,1)$.
3. If $U\pi_{X_{k-1}}r_{X_{k-1},Y_k} \le \pi_{Y_k}r_{Y_k,X_{k-1}}$, set $X_k = Y_k$, otherwise set $X_k = X_{k-1}$.
4. If $k < n$, go to 2., otherwise return $(X_0, \ldots, X_n)$.

Proposition 28 *Under the assumptions above, the Metropolis-Hastings algorithm simulates a Markov chain with transition matrix $P = (p_{i,j}, i, j \in \mathcal{X})$, where*

$$p_{i,j} = \min\left\{r_{i,j}, \frac{\pi_j r_{j,i}}{\pi_i}\right\}, \quad i \neq j \qquad \text{and} \qquad p_{i,i} = 1 - \sum_{j\in\mathcal{X}:j\neq i} p_{i,j}.$$

The P-Markov chain is irreducible and reversible with stationary distribution π.

Proof: The Markov property follows directly from the explicit sequential nature of the algorithm that only draws on X_{k-1} and independent randomness from $U \sim \text{Unif}(0,1)$. For the transition probabilities, note that for $i \neq j$

$$p_{i,j} = \mathbb{P}(Y_k = j|X_{k-1} = i)\mathbb{P}\left(U \le \frac{\pi_{Y_k}r_{Y_k,X_{k-1}}}{\pi_{X_{k-1}}r_{X_{k-1},Y_k}}\middle| X_{k-1} = i, Y_k = j\right) = r_{i,j}\min\left\{1, \frac{\pi_j r_{j,i}}{\pi_i r_{i,j}}\right\},$$

any rejection leads to a transition $i \to i$ accumulating the remaining probability mass.

Irreducibility of P follows from irreducibility of R, because the hypotheses on R and π ensure that

$$p_{i,j} > 0 \iff r_{i,j} > 0, \pi_j > 0, r_{j,i} > 0 \iff r_{i,j} > 0.$$

For reversibility, note that for any $i, j \in \mathcal{X}$, $i \neq j$, we have $\pi_j r_{j,i} \ge \pi_i r_{i,j}$ or $\pi_j r_{j,i} \le \pi_i r_{i,j}$. In the first case, we have $p_{i,j} = r_{i,j}$ and $p_{j,i} = \pi_i r_{i,j}/\pi_j = \pi_i p_{i,j}/\pi_j$. The second case follows by symmetry. □

Corollary 29 *For any function $\phi\colon \mathcal{X} \to \mathbb{R}$, the Monte-Carlo estimator is consistent:*

$$\frac{1}{n}\sum_{j=0}^{n-1}\phi(X_j) \to \mathbb{E}(\phi(X)) = \sum_{i\in\mathcal{X}}\phi(i)\pi_i. \qquad \textit{in probability (a.s.) as } n \to \infty.$$

Proof: The Ergodic Theorem applies as the chain is irreducible and $\mathcal{X}$ finite. □

Example 30 Consider the distribution $\pi_j = j/Z_m$, $j \in \mathcal{X} = \{1,\ldots,m\}$, where $Z_m = m(m+1)/2$ is the normalisation constant. We use simple random walk $r_{i,i+1} = 1-r_{i,i-1} = p$ for $j \in \{2,\ldots,m-1\}$ with boundary transitions $r_{1,2} = r_{m,m-1} = 1$. Then the acceptance condition, for $j = i+1 \in \{3,\ldots,m-1\}$ can be written as

$$U\pi_{i+1}r_{i+1,i} \le \pi_i r_{i,i+1} \iff U \le \frac{ip}{(i+1)(1-p)}.$$

Similarly, for $j = i-1 \in \{2,\ldots,m-2\}$,

$$U\pi_{i-1}r_{i-1,i} \le \pi_i r_{i,i-1} \iff U \le \frac{i(1-p)}{(i-1)(1-p)}.$$

The boundary cases are similar. The thresholds are

$$2 \to 1 : \frac{1}{2(1-p)}, \quad 1 \to 2 : 2p, \quad (m-1) \to m : \frac{m}{(m-1)p}, \quad m \to (m-1) : \frac{(m-1)(1-p)}{m}.$$

The resulting Markov chain with stationary distribution π has transition probabililities

$$p_{i,j} = \min\left\{r_{i,j}, \frac{\pi_j}{\pi_i}r_{j,i}\right\} = \min\left\{r_{i,j}, \frac{j}{i}r_{j,i}\right\}, \quad p_{i,i} = 1 - \sum_{j\in\mathcal{X}:j\neq i} p_{i,j},$$

which we leave in this form. Since the stationary distribution puts more weight into higher states, we get fewer rejections and hence better mixing properties (faster convergence to stationarity) by choosing $p \ge 1/2$, but not too close to 1.

Here is an example that gives rise to a Markov chain Monte-Carlo algorithm that is not a Metropolis-Hastings algorithm, i.e. a non-example in our context here.

Example 31 Consider the Markov chain with transition matrix $\widetilde{P} = (\widetilde{p}_{i,j}, i,j \in \mathcal{X})$, where $\mathcal{X} = \{1,\ldots,m\}$ and

$$\widetilde{p}_{i,i-1} = \frac{i-1}{i}, \quad \widetilde{p}_{i,m} = \frac{1}{i}, \quad i \in \mathcal{X}.$$

It is easy to check that $\pi_i = i/Z_m$ is the stationary distribution of this Markov chain. How did we find the chain? We exploited the special structure of the problem, with increasing probability mass function and solved equations $\pi P = \pi$ for P, together with $p_{i,j} = 0$ except for $j = m$ or $j = i-1$. This chain is not reversible, because $\pi_i p_{i,i-1} > 0$ while $\pi_{i-1}p_{i-1,i} = 0$ for $i < m$. This example of constructing a Markov chain with given stationary distribution does not generalise to general state space. The strength of the Metropolis-Hastings algorithm is that it applies in extremely general situations. There may well be "better" Markov chains.

7.2 The Metropolis-Hastings algorithm in a more general setting

In the last section we made a number of assumption that can be relaxed.

- If $\mathcal{X}$ is countably infinite, the proof of Proposition 28 remains valid. The proof of Corollary 29 now relies on a different version of the Ergodic Theorem. In fact, irreducibility and the existence of a stationary distribution are enough for the conclusion to hold, and these are both provided by Proposition 28.
- If we allow $\pi_i = 0$ for some $i \in \mathcal{X}$, the Metropolis-Hastings algorithm still simulates a reversible Markov chain with transition matrix P. Since $p_{i,j} = 0$ if $\pi_i > 0$ and $\pi_j = 0$, we can restrict the state space of the P-chain (but not of the R-chain) to $\widetilde{\mathcal{X}} = \{i \in \mathcal{X} : \pi_i > 0\}$. Irreducibility is not automatic and needs to be checked case by case. This is important, because in the reducible case, the stationary distribution is not unique and the Ergodic Theorem will apply on each communicating class, converging to different stationary distributions and not to π. If the P-chain is irreducible, the Monte-Carlo estimator is consistent.
- If we allow one-way transitions $r_{i,j} > 0$ while $r_{j,i} = 0$, we will have $p_{i,j} = p_{j,i} = 0$ (unless $\pi_i = 0$, but in that case we restrict the state space of the P-chain to $\widetilde{\mathcal{X}}$, so these transitions are irrelevant). Again, the Metropolis-Hastings algorithm simulates a reversible Markov chain, but irreducibility does not follow, in general. If we modify the proposal transition matrix R into a matrix $\widetilde{R}$ by disallowing one-way transitions in $\widetilde{R}$, i.e.

$$\widetilde{r}_{i,j} = \begin{cases} r_{i,j} & \text{if } r_{j,i} > 0, \\ 0 & \text{otherwise,} \end{cases} \qquad \text{and} \quad \widetilde{r}_{i,i} = r_{i,i} + \sum_{j \in \mathcal{X} : r_{j,i} = 0} r_{i,j},$$

  we find $P = \widetilde{P}$ for the associated Metropolis-Hastings chains. Hence, we can assume $r_{i,j} > 0 \iff r_{j,i} > 0$ without loss of generality.

Example 32 Here are (trivial) examples to demonstrate the irreducibility problems:

(a) Consider a random walk on $\mathcal{X} = \{1, 2, 3\}$ arranged as $1 \leftrightarrow 2 \leftrightarrow 3$:

$$R = \begin{pmatrix} 0 & 1 & 0 \\ 1/2 & 0 & 1/2 \\ 0 & 1 & 0 \end{pmatrix}, \qquad \pi = (1/2, 0, 1/2).$$

R is irreducible, but 1 and 3 only communicate via 2, and $\pi_2 = 0$ makes $p_{1,2} = p_{3,2} = 0$. With $p_{1,3} = p_{3,1} = 0$ inherited from $r_{1,3} = r_{3,1} = 0$, we find

$$P = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 0 & 1/2 \\ 0 & 0 & 1 \end{pmatrix}$$

has a very different class structure: closed classes $\{1\}$ and $\{3\}$ and an open class $\{2\}$ that is irrelevant for the Monte-Carlo limits.

(b) Consider a (deterministic) walk on $\mathcal{X} = \{1, 2, 3\}$ arranged in a circle $1 \to 2 \to 3 \to 1$

$$R = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \qquad \pi = (1/3, 1/3, 1/3).$$

R is irreducible, but all transitions are one-way; indeed

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

has a different class structure: three closed classes $\{1\}$, $\{2\}$ and $\{3\}$.

7.3 Example: the Ising model

Most of our applications and examples have been rather elementary simulations that do not bear justice to the power of simulation to penetrate beyond the bounds of analytic tractability. To give a flavour of such applications (albeit still analytically tractable), consider the Ising model: on a finite integer lattice $\Lambda = (\mathbb{Z}/N\mathbb{Z})^2 = \{0, \dots, N-1\}^2$, equipped with the usual (horizontal and vertical, but not diagonal) neighbour relation $\lambda \sim \mu$, each lattice point $\lambda = (i, j) \in \Lambda$ has one of two types $s(\lambda) = s(i, j) \in \{0, 1\}$. The space of possible states of this system is

$$\mathcal{X} = \{0, 1\}^\Lambda = \{s \colon \Lambda \to \{0, 1\}\}$$

Physicists consider the following Gibbs distributions on $\mathcal{X}$

$$\pi(s) = \frac{1}{Z_\beta} \exp\left(-\beta \sum_{\lambda,\mu \in \Lambda : \lambda \sim \mu} (s(\lambda) - s(\mu))^2 \right).$$

Let us set up a Metropolis-Hastings algorithm to simulate from this distribution. As proposed transition mechanism we use single spin flips (turning 1 into 0 and vice versa) chosen uniformly among the N^2 lattice points of Λ. Now suppose that the proposed flip is at $\lambda = (i, j)$. Then a configuration s is turned into t, where $t(i, j) = 1 - s(i, j)$ and $t(\mu) = s(\mu)$ for $\mu \neq \lambda$. Hence $r_{s,t} = r_{t,s}$ for all $s, t \in \mathcal{X}$ and

$$\frac{\pi(t)}{\pi(s)} = \exp\left(2\beta(2 - \#\{\mu \in \Lambda : \mu \sim \lambda, s(\mu) = s(\lambda))\}\right).$$

Therefore, the algorithm is

1. Let $X_0 \sim \mathrm{Unif}(\mathcal{X})$ and $k = 0$.

2. Set $k := k + 1$. Generate independent $V = \mathrm{Unif}(\Lambda)$ and $U \sim \mathrm{Unif}(0, 1)$.

3. If $U \leq \exp\left(2\beta(2 - \#\{\mu \in \Lambda : \mu \sim V, s(\mu) = s(V))\}\right)$, set X_k equal to X_{k-1} with the V-bit flipped, otherwise set $X_k = X_{k-1}$.

4. If $k < n$, go to 2., otherwise return $(X_0, \dots, X_n)$.

Figure 7.1 intends to display typical configurations under the stationary distribution for various choices of $\beta \in \mathbb{R}$. For negative values of β different types attract, while equal types repel, while for positive values of β the different types repel, while equal types attract. The result is plausible for $\beta \in \{-1, 0, 0.5, 1\}$, possibly $\beta = 2$, but not $\beta = 10$.

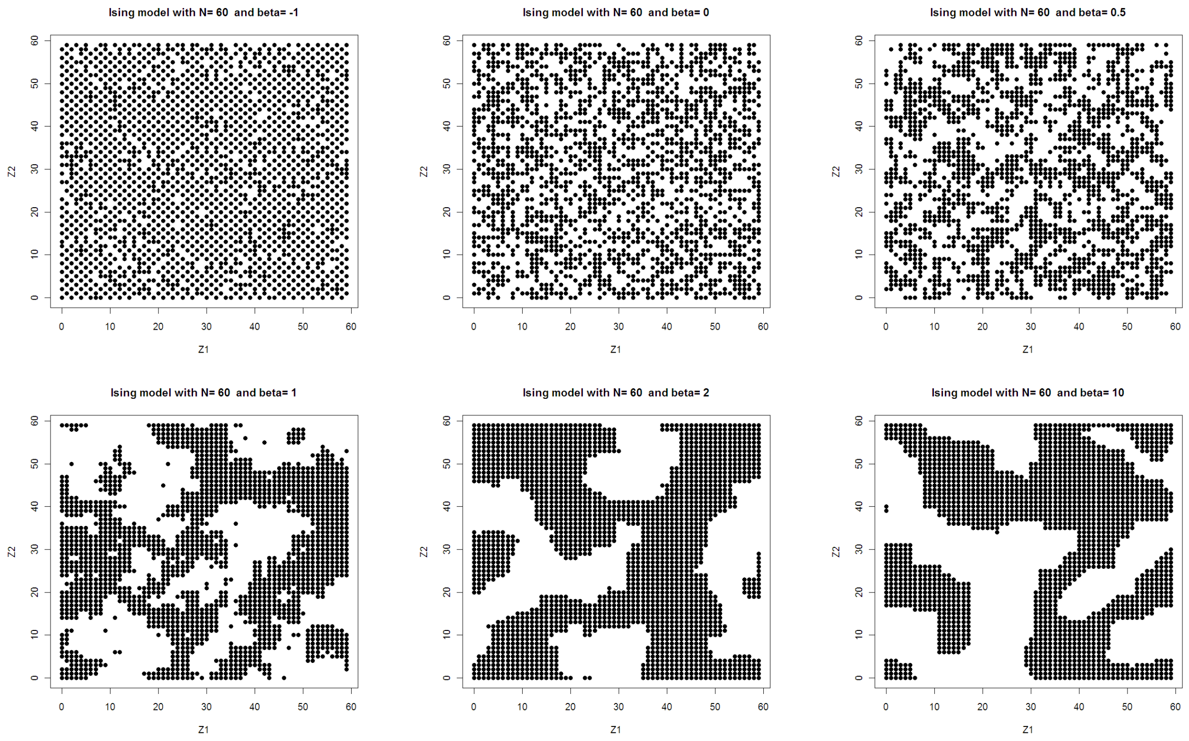


Figure 7.1: Simulations of the Ising model

```
N<-60
beta<--1
X<-matrix(rbinom((N+2)*(N+2),1,1/2),N+2,N+2)

connect<-function(N,A){           #copy neighbours from far boundary
    A[1,]<-A[N+1,];A[,1]<-A[,N+1];A[N+2,]<-A[2,];A[,N+2]<-A[,2];A}

n<-100000
for (k in 1:n) {
    X<-connect(N,X)
    V1<-floor(N*runif(1))+2;V2<-floor(N*runif(1))+2
    U<-runif(1)
    if (U<exp(2*beta*(2-(X[V1,V2]==X[V1-1,V2])-(X[V1,V2]==X[V1+1,V2])
                     -(X[V1,V2]==X[V1,V2-1])-(X[V1,V2]==X[V1,V2+1])))) {
       X[V1,V2]<-1-X[V1,V2]}
    }

Z1<-rep(NA,N^2);Z2<-rep(NA,N^2)
k<-1
for (i in 1:N) {
    for (j in 1:N){
        if (X[i+1,j+1]==1){
           Z1[k]<-i-1;Z2[k]<-j-1;k<-k+1}}}

plot(Z1,Z2,pch=16,main=paste("Ising model with N=",N," and beta=",beta))
```

For $\beta = 0$, all proposed swaps are accepted. For $\beta \to \infty$, there are two best configurations, namely "all type 0" and "all type 1" that, asymptotically, take probability 1/2 each under the stationary distribution – there are only finitely many configurations and the next likely configurations are the ones with only one site of different type, but these are already separated by a multiplicative factor of $\exp(-4\beta) \to 0$. This factor is already quite small and means that it is extremely unlikely to create new islands of one type within another, and if so they disappear again quickly. However, the algorithm first creates regions of both types that compete, with lots of activity back and forth at their boundary. It takes a long time for the last regions of the losing type to get smaller and disappear.

```
N<-60
beta<-10
X<-matrix(rbinom((N+2)*(N+2),1,1/2),N+2,N+2)

Z1<-rep(NA,N^2);Z2<-rep(NA,N^2)
k<-1
for (i in 1:N) {
    for (j in 1:N){
        if (X[i+1,j+1]==1){
           Z1[k]<-i-1;Z2[k]<-j-1;k<-k+1}}}

plot(Z1,Z2,pch=16,main=paste("Ising model with N=",N," and beta=",beta))

connect<-function(N,A){          #copy neighbours from far boundary
    A[1,]<-A[N+1,];A[,1]<-A[,N+1];A[N+2,]<-A[2,];A[,N+2]<-A[,2];A}

n<-100000
for (k in 1:n) {
    X<-connect(N,X)
    V1<-floor(N*runif(1))+2;V2<-floor(N*runif(1))+2
    U<-runif(1)
    if (U<exp(2*beta*(2-(X[V1,V2]==X[V1-1,V2])-(X[V1,V2]==X[V1+1,V2])
                    -(X[V1,V2]==X[V1,V2-1])-(X[V1,V2]==X[V1,V2+1])))) {
       X[V1,V2]<-1-X[V1,V2]
   if (X[V1,V2]==1) {points(V1-2,V2-2,col="black",pch=16)}
   else {points(V1-2,V2-2,col="white",pch=16)}
    }
    else {points(V1-2,V2-2,col=k)} #coloured open circles display rejections
}
```

This was all on a torus (i.e. with 0 neighbouring $N-1$ in both directions). Instead of connecting this way, we can fix boundary conditions, say all type 0, or type 0 on the left half and type 1 on the right. This influences the behaviour. In the code, just remove the connect line and set up the initial matrix as

```
X<-matrix(rbinom((N+2)*(N+2),1,1/2),N+2,N+2)
X[1,]<-0;X[N+2,]<-1;X[,1]<-0;X[,N+2]<-1
```

For the case of fixed boundary, Figure 7.2 shows three simulations with uniform random initial conditions (top) and three simulations (bottom) from the initial condition that is most likely under the stationary distribution, which has an upper triangle of type 1 and a lower triangle of type 0, so we see the fluctuations away from the optimal configuration, all for $\beta = 10$ here. This demonstrates, in particular, that convergence to stationarity has not happened in the simulations that started from uniform random initial conditions.

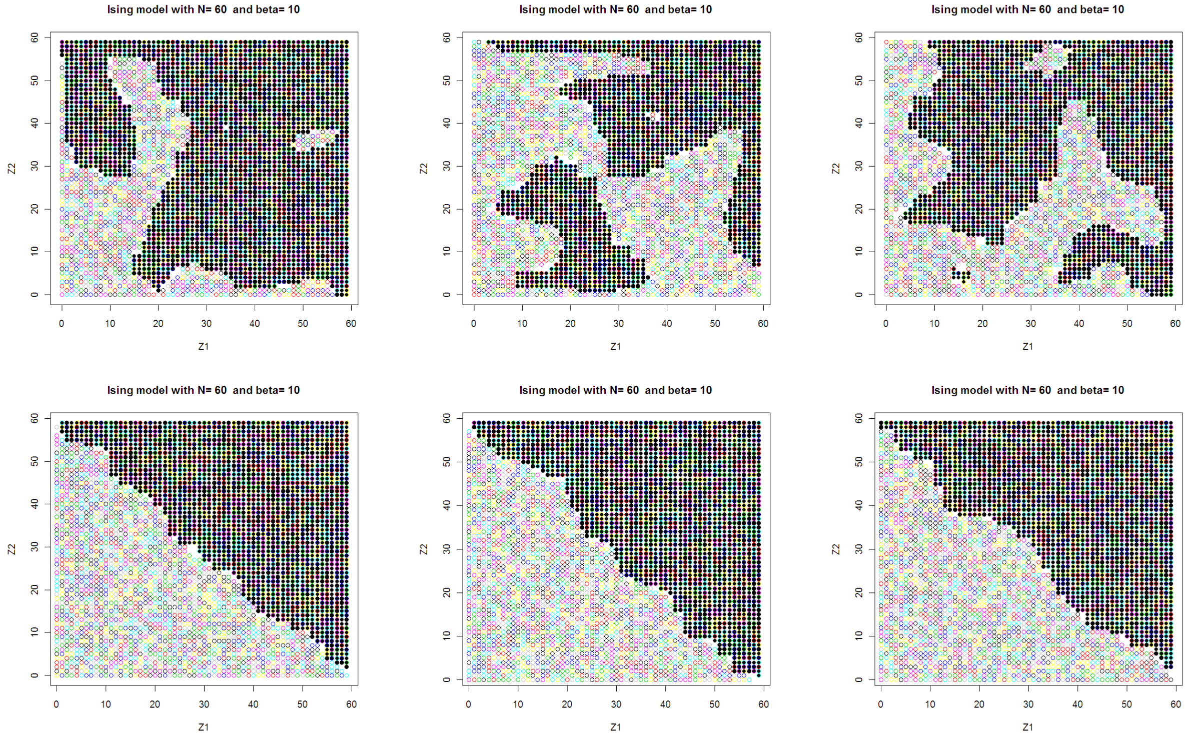


Figure 7.2: Simulations of the Ising model for $\beta = 10$ with fixed boundary

Here is the code for the upper triangular initial conditions:

```
for (i in 1:N+2) {
    for (j in 1:N+2) {
        if (i+j>N+2) {X[i,j]<-1}
        else {X[i,j]<-0}}}
```

Lecture 8

Unknown normalisation constants

Reading: Casella and Berger Section 3.3

8.1 Normalisation constants

All common probability distributions are based on a convergent integral or series, because we require probability density functions $f\colon \mathbb{R}^d \to [0,\infty)$ to integrate to 1 and probability mass functions $(p_j)_{j\in\mathcal{X}}$ to sum up to 1. This nearly always means that they are *naturally* written in the form

$$f(x) = \frac{\widetilde{f}(x)}{C}, \quad \text{where } C = \int_{\mathbb{R}^d} \widetilde{f}(x)dx, \qquad \text{or} \qquad p_j = \frac{\widetilde{p}_j}{C}, \qquad \text{where } C = \sum_{j\in\mathcal{X}} \widetilde{p}_j,$$

where $\widetilde{f}$ or $\widetilde{p}$ is "simpler".

Example 33 (i) Uniform distribution $\widetilde{f}(x) = 1$, $x \in D$, for some $D \subset \mathbb{R}^d$ of finite volume $\mathrm{Vol}(D)$; here $C = \mathrm{Vol}(D)$.

(ii) Uniform distribution $\widetilde{p}_j = 1$, $j \in \mathcal{X}$, for some finite set $\mathcal{X}$; here $C = \#\mathcal{X}$.

(iii) Poisson distribution $\widetilde{p}_j = \lambda^j/j!$, $j \geq 0$, for some $\lambda \in (0,\infty)$; here $C = e^\lambda$, by the exponential series.

(iv) Gamma distribution $\widetilde{f}(x) = x^{\alpha-1}e^{-\lambda x}$, $x > 0$, for some $\alpha,\lambda \in (0,\infty)$; here $C = \Gamma(\alpha)/\lambda^\alpha$, by the Gamma integral (and a linear change of variables $y = \lambda x$).

(v) Beta distribution $\widetilde{f}(x) = x^{\alpha-1}(1-x)^{\beta-1}$; here $C = \Gamma(\alpha)\Gamma(\beta)/\Gamma(\alpha+\beta)$, by the Beta integral.

(vi) Normal distribution $\widetilde{f}(x) = \exp(-(x-\mu)^2/2\sigma^2)$, for some $\mu \in \mathbb{R}$, $\sigma^2 > 0$; here $C = \sqrt{2\pi\sigma^2}$.

(vii) Gibbs distribution $\widetilde{p}_j = \exp(-H(j)) = \exp(-\sum_{i\in\Lambda} h(j_i))$, $j = (j_i, i \in \Lambda) \in \mathcal{X} = S^\Lambda$ for some finite sets S and Λ and some function $h\colon S \to \mathbb{R}$.

8.2 Rejection method

As mentioned before, normalisation constants of proposal and target distribution always cancel in the rejection method, because the (optimal) acceptance condition is

$$\sup\{f(x)/h(x) : x \in \mathbb{R}^d\}Uh(X) \leq f(X) \iff \sup\{\widetilde{f}(x)/\widetilde{h}(x) : x \in \mathbb{R}^d\}U\widetilde{h}(X) \leq \widetilde{f}(X),$$

or, respectively in the discrete setting

$$\sup\{\pi(j)/\xi(j) : j \in \mathcal{X}\}U\xi(X) \leq \pi(X) \iff \sup\{\widetilde{\pi}(j)/\widetilde{\xi}(j) : j \in \mathcal{X}\}U\widetilde{\xi}(X) \leq \widetilde{\pi}(X).$$

We can actually go one step further and use the rejection algorithm to estimate normalisation constants. Specifically, recall that the number of trials up to and including the first acceptance is geom$(1/M)$, where (in the continuous case, say)

$$\begin{aligned} M &= \sup\{f(x)/h(x) : x \in \mathbb{R}^d\} = \sup\{\widetilde{f}(x)C_{\widetilde{h}}/\widetilde{h}(x)C_{\widetilde{f}} : x \in \mathbb{R}^d\} \\ &= \frac{C_{\widetilde{h}}}{C_{\widetilde{f}}}\sup\{\widetilde{f}(x)/\widetilde{h}(x) : x \in \mathbb{R}^d\}. \end{aligned}$$

The maximum likelihood estimator for geom(p) based on a sample $N_1, \ldots, N_n$ is

$$\hat{p} = \frac{n}{N_1 + \cdots + N_n},$$

and we need $\widetilde{M} = \sup\{\widetilde{f}(x)/\widetilde{h}(x) : x \in \mathbb{R}^d\}$ to implement the algorithm, so we can use

$$\widehat{C_{\widetilde{h}}/C_{\widetilde{f}}} = \frac{1}{\widetilde{M}\hat{p}} = \frac{1}{\widetilde{M}n}\sum_{i=1}^{n} N_i.$$

This estimator is unbiased as $\mathbb{E}(N_i) = 1/M$, and it is independent from the accepted sample. If the proposal constant $C_{\widetilde{h}}$ is known, we obtain an unbiased estimator for the (reciprocal of the) target constant $1/C_{\widetilde{f}}$:

$$\widehat{1/C_{\widetilde{f}}} = \frac{1}{\widetilde{M}C_{\widetilde{h}}n}\sum_{i=1}^{n} N_i.$$

8.3 Importance sampling

In order to implement importance sampling as in Lecture 5 we have to know normalisation constants, since

$$\widehat{\theta}^{\text{IS}} = \frac{1}{n}\sum_{j=1}^{n}\phi(Z_j)\frac{f(Z_j)}{h(Z_j)},$$

cf. Proposition 24. There is an Importance Sampling estimator for un-normalized densities too, and it is in this form that Importance Sampling estimation is usually conducted.

Let $\widetilde{f}, \widetilde{h} \colon \mathbb{R}^d \to [0, \infty)$ be unnormalised probability density functions and $\phi \colon \mathbb{R}^d \to \mathbb{R}$ a function. Denote by $C_{\widetilde{f}} = \int_{\mathbb{R}^d} \widetilde{f}(x)dx$ and $C_{\widetilde{h}} = \int_{\mathbb{R}^d} \widetilde{h}(x)dx$ the normalisation constants that lead to probability density functions $f = \widetilde{f}/C_{\widetilde{f}}$ and $h = \widetilde{h}/C_{\widetilde{h}}$. Suppose that we wish to estimate $\theta = \mathbb{E}(\phi(X)) = \mathbb{E}(\phi(Z)f(Z)/h(Z))$ for a random variable $X \sim f$ using importance sampling based on a sample $Z_1, \ldots, Z_n \sim h$.

If the normalizing constants $C_{\widetilde{f}}$ and $C_{\widetilde{h}}$ are not available, we separate them out and write

$$\mathbb{E}\left(\phi(Z)\frac{f(Z)}{h(Z)}\right) = \frac{C_{\widetilde{h}}}{C_{\widetilde{f}}}\mathbb{E}\left(\phi(Z)\frac{\widetilde{f}(Z)}{\widetilde{h}(Z)}\right).$$

Note that (taking $\phi \equiv 1$), we have an unbiased estimator for $C_{\widetilde{f}}/C_{\widetilde{h}}$:

$$\mathbb{E}\left(\frac{\widetilde{f}(Z)}{\widetilde{h}(Z)}\right) = \frac{C_{\widetilde{f}}}{C_{\widetilde{h}}}\mathbb{E}\left(\phi(Z)\frac{f(Z)}{h(Z)}\right) = \frac{C_{\widetilde{f}}}{C_{\widetilde{h}}}\mathbb{E}(\phi(X)) = \frac{C_{\widetilde{f}}}{C_{\widetilde{h}}}.$$

It follows that (for general $\phi\colon \mathbb{R}^d \to \mathbb{R}$)

$$\mathbb{E}(\phi(X)) = \frac{\mathbb{E}(\phi(Z)\widetilde{f}(Z)/\widetilde{h}(Z)}{\mathbb{E}(\widetilde{f}(Z)/\widetilde{h}(Z))},$$

and we can estimate the numerator (A, say) and denominator (B, say) separately.

The resulting algorithm for the estimation of $\mathbb{E}(\phi(X))$ is:

1. Simulate independent $Z_1, \dots, Z_n \sim h$.
2. Set $\widetilde{w}(Z_j) = \widetilde{f}(Z_j)/\widetilde{h}(Z_j)$, for all $j = 1, \dots, n$.
3. Set $\widehat{A} = \frac{1}{n}\sum_{j=1}^{n}\widetilde{w}(Z_j)\phi(Z_j)$ and $\widehat{B} = \frac{1}{n}\sum_{j=1}^{n}\widetilde{w}(Z_j)$.
4. Return the ratio $\widehat{\theta}_R^{\mathrm{IS}} = \widehat{A}/\widehat{B}$ as modified Importance Sampling estimate for $\mathbb{E}(\phi(X))$.

While $\mathbb{E}(\widehat{A}) = \mathbb{E}(\phi(Z)\widetilde{f}(Z)/\widetilde{h}(Z)) = A$ and $\mathbb{E}(\widehat{B}) = B$, we have $\mathbb{E}(\widehat{\theta}_R^{\mathrm{IS}}) \neq \theta = \mathbb{E}(\phi(X))$, in general, since the expectation of a ratio is not (usually) the ratio of expectations. In our algorithm we use the same sample $Z_1, \dots, Z_n$ to estimate numerator and denominator, but even if we used independent samples, we would still end up with $\mathbb{E}(1/\widehat{B}) \neq 1/\mathbb{E}(\widehat{B}) = 1/B$. In fact, this latter inequality is always "strictly greater than" (by a form of Jensen's Inequality). In other words, while $\widehat{A}$ and $\widehat{B}$ are unbiased, $\widehat{\theta}_R^{\mathrm{IS}}$ may be biased.

Proposition 34 *The modified Importance Sampling estimator $\widehat{\theta}_R^{\mathrm{IS}}$ is consistent:*

$$\widehat{\theta}_R^{\mathrm{IS}} \to \theta \qquad \textit{(in probability or a.s.) as } n \to \infty.$$

Proof: As in Proposition 23, (weak or strong) consistency of $\widehat{A}$, and also of $\widehat{B}$, follows from the (Weak or Strong) Law of Large Numbers, applied to $Y_j = \phi(Z_j)\widetilde{f}(Z_j)/\widetilde{h}(Z_j)$, with $\phi \equiv 1$ in the case of $\widehat{B}$. Strong consistency means

$$\begin{array}{l}\mathbb{P}(\widehat{A} \to A) = 1\\ \mathbb{P}(\widehat{B} \to B) = 1\end{array} \quad \Rightarrow \quad \mathbb{P}(\widehat{A} \to A \text{ and } \widehat{B} \to B) = 1 \quad \Rightarrow \quad \mathbb{P}\left(\frac{\widehat{A}}{\widehat{B}} \to \frac{A}{B}\right) = 1.$$

But since almost sure convergence and the Strong Law of Large Numbers are not Part A material, we give a rather more cumbersome direct proof of weak consistency of $\widehat{\theta}_R^{\mathrm{IS}}$, which establishes the ratio part of the algebra of limits for convergence in probability.

Let us write $\widehat{A}_n$ and $\widehat{B}_n$ for $\widehat{A}$ and $\widehat{B}$ based on sample size n. Let $\varepsilon > 0$ and $\delta > 0$. Then we can find $n_0 \geq 0$ such that for all $n \geq n_0$

$$\mathbb{P}\left(\left|\widehat{B}_n - B\right| > \frac{B}{2}\right) < \frac{\delta}{3} \quad \text{and} \quad \mathbb{P}\left(\left|\widehat{A}_n - A\right| > \frac{\varepsilon B}{4}\right) < \frac{\delta}{3} \quad \text{and} \quad \mathbb{P}\left(A\left|\widehat{B}_n - B\right| > \frac{\varepsilon B^2}{4}\right) < \frac{\delta}{3}.$$

Then, we will also have for all $n \geq n_0$

$$\begin{aligned}\mathbb{P}\left(\left|\frac{\widehat{A}_n}{\widehat{B}_n} - \frac{A}{B}\right| > \varepsilon\right) &\leq \mathbb{P}\left(\left|\widehat{B}_n - B\right| > \frac{B}{2}\right) + \mathbb{P}\left(\left|\widehat{B}_n - B\right| \leq \frac{B}{2}, \left|\widehat{A}_n B - A\widehat{B}_n\right| > \varepsilon \widehat{B}_n B\right) \\ &< \frac{\delta}{3} + \mathbb{P}\left(\left|\widehat{A}_n B - AB\right| > \frac{\varepsilon B^2}{4}\right) + \mathbb{P}\left(\left|AB - A\widehat{B}_n\right| > \frac{\varepsilon B^2}{4}\right) < \delta,\end{aligned}$$

where the middle step uses $\widehat{B}_n > B/2$, and $|D_1 + D_2| > c \Rightarrow |D_1| > c/2$ or$|D_2| > c/2$. □

Example 35 We wish to estimate $\mathbb{E}(\phi(X))$ for the target distribution $X \sim \text{Gamma}(\alpha, \beta)$ using the proposal distribution $X \sim \text{Gamma}(a, b)$, e.g. $a \in \mathbb{N}$, i.e. where

$$\widetilde{f}(x) = x^{\alpha-1}e^{-\beta x}, \quad \widetilde{h}(x) = x^{a-1}e^{-bx}, \quad x > 0, \quad \text{and} \quad C_{\widetilde{f}} = \beta^{-\alpha}\Gamma(\alpha), \quad C_{\widetilde{h}} = b^{-a}\Gamma(a).$$

Suppose we do not know $C_{\widetilde{f}}$ and $C_{\widetilde{h}}$ (or we are not confident we can calculate it correctly or we cannot be bothered calculating it!). Now $\widetilde{f}(x)/\widetilde{h}(x) = x^{\alpha-a}\exp(-(\beta - b)x)$, so:

1. Simulate independent $Z_1, \ldots, Z_n \sim \text{Gamma}(a, b)$.
2. Set $\widetilde{w}_j = Z_j^{\alpha-a}\exp(-(\beta - b)Z_j)$, for all $j = 1, \ldots, n$.
3. Return the modified Importance Sampling estimate $\widehat{\theta}_R^{\text{IS}} = \dfrac{\sum_j \widetilde{w}_j \phi(Z_j)}{\sum_j \widetilde{w}_j}$.

Simulation studies show that often the ratio estimator $\widehat{\theta}_R^{\text{IS}}$ gives more stable estimates of $\mathbb{E}(\phi(X))$ than the basic Importance Sampling estimator $\widehat{\theta}^{\text{IS}}$. Some authors recommend using $\widehat{\theta}_R^{\text{IS}}$ even when you know and can easily calculate all the normalizing constants. This is surprising as we have an extra quantity to estimate for the ratio estimator.

8.4 Metropolis-Hastings algorithm

For $\pi = \widetilde{\pi}/C$, we can also estimate C in the Metropolis-Hastings algorithm , e.g. $C = \#\mathcal{X}$ with $\widetilde{\pi} \equiv 1$, : naively, take any $i \in \mathcal{X}$ and note $\widehat{\pi}_i = \#\{j \in \{0, \ldots, n-1\} : X_j = i\}/n \to \widetilde{\pi}_i/C$, so $1/C \approx \widehat{\pi}_i/\widetilde{\pi}_i$. However, more sophisticated techniques are needed for big $\mathcal{X}$.

8.5 Conclusion

In this lecture course, we have studied the main simulation techniques of inversion, transformation, rejection, importance sampling, and Markov chain Monte-Carlo methods such as the Metropolis-Hastings algorithm. For illustration of the techniques we have mostly used the common families of discrete and continuous probability distributions. It must be stressed, however, that the techniques apply much more widely, and that is where their full power can be exploited – however, the introduction to application areas would use a disproportionate amount of time in an 8-hour lecture course, so we have largely stayed clear of such developments and refer to the literature for further reading.

Appendix A

Assignments

Assignment sheets are issued on Mondays of weeks 2-4. They are made available on the website of the course at

http://www.stats.ox.ac.uk/∼winkel/ASim.html.

Two sets of three one-hour classes take place in room 102 of 1 South Parks Road (Lecture Room, Department of Statistics) in weeks 2 to 4, on Fridays, as follows:

- Fridays 11.00am-12.00pm
- Fridays 12.00-1.00pm

The class allocation can be accessed from the course website shortly after the first lecture, as soon as I have entered the details, soon after the first lecture.

Scripts are to be handed in at the Department of Statistics, 1 South Parks Road.

Exercises on the problem sheets vary in style and difficulty. If you find an exercise difficult, please do not deduce that you cannot solve the following exercises, but aim at giving each exercise a serious try. **Solutions will be provided on the course website.**

There are lecture notes available. Please print these so that we can make best use of the lecture time. Please let me know of any typos or other comments that you may have, e.g. by sending an email to winkel@stats.ox.ac.uk.

Below are some comments on the recommended Reading and Further Reading literature.

Ross: Simulation 2006

This is our main reference. The style is somewhat more elementary than our course, but our syllabus is well-coverered. There are lots of examples and exercises, so this book is the ideal place to find more detailed explanations and illustrations, where our course takes bigger steps or focusses on the overall picture. We use material from most of the 11 chapters, so it is best used as a reference book in conjunction with rather than instead of these lecture notes.

Ripley: Stochastic Simulation 1987

This book is now almost 25 years and some of the detailed discussions of (old) random number generators and efficiency considerations tend to play a much less prominent part nowadays. However, the issues are still there in principle, and it is still instructive to study them. We occasionally refer to this book. The style is concise. Some topics receive a briefer treatment than in our course, but there is much more in the book.

Robert and Casella: Monte Carlo Statistical Methods 1999

This book is more advanced than our course. Many of our topics are treated in much more depth, so the main ideas for our course are well-spread over the book. I recommend this book for further developments and techniques that are specific to certain applications.

Norris: Markov Chains 1997

This is a book on the theory and applications of Markov chains, written mainly for mathematicians with an interest in probability theory, but not necessarily statistics. Only Section 5.5 discusses a simulation topic: Markov chain Monte Carlo. However, Norris is our main reference for that part of the course, i.e. for Lectures 6 and 7.

A.1 Inversion, transformation and rejection

Please hand in scripts by Wednesday 11 May 2011, 10am, Department of Statistics. 2.(c) and 4. require material from Lecture 3, the others are on week 1 material only.

1. (a) Consider a random variable X with continuous and strictly increasing cumulative distribution function $F_X\colon \mathbb{R} \to (0,1)$. Show that

$$F_X(X) \sim \text{Unif}(0,1)$$

and that

$$F_X^{-1}(U) \sim X, \qquad \text{where } U \sim \text{Unif}(0,1).$$

(b) Consider a random variable X with any (right-continuous and non-decreasing) cumulative distribution function $F_X\colon \mathbb{R} \to [0,1]$. Let $A_u = \{z \in \mathbb{R}\colon F_X(z) > u\}$ for all $u \in (0,1)$. Show that

$$u < F_X(x) \;\Rightarrow\; \inf A_u \le x \;\Rightarrow\; u \le F_X(x) \qquad \text{for all } x \in \mathbb{R} \text{ and } u \in (0,1).$$

Hence, or otherwise, show that

$$\inf A_U \sim X, \qquad \text{where } U \sim \text{Unif}(0,1).$$

Define the generalised inverse $F_X^{-1}(u) = \inf A_u = \inf\{z \in \mathbb{R} : F_X(z) > u\}$ and compare with (a); also find a necessary and sufficient condition for

$$F_X(X) \sim \text{Unif}(0,1).$$

2. (a) Let $Y \sim \text{Exp}(\lambda)$. Fix $a > 0$ and consider a random variable X with

$$F_X(x) = \mathbb{P}(X \le x) = \mathbb{P}(Y \le x | Y > a), \qquad x \in \mathbb{R}.$$

For $u \in (0,1)$, calculate $F_X^{-1}(u)$ and hence give an algorithm simulating X based on $U \sim \text{Unif}(0,1)$.

(b) Let Z be any real-valued random variable. Fix $a < b$ with $\mathbb{P}(a < Z \le b) > 0$ and consider a random variable W with

$$\mathbb{P}(W \le w) = \mathbb{P}(Z \le w | a < Z \le b), \qquad w \in \mathbb{R}.$$

Show that

$$F_Z^{-1}(F_Z(a)(1-U) + F_Z(b)U) \sim W, \qquad \text{where } U \sim \text{Unif}(0,1).$$

Apply this formula to simulate the random variable X from (a) above.

(c) Here is another algorithm to simulate the random variable X from (a) above:

```
1. Simulate $Y \sim \text{Exp}(\lambda)$;
2. if $Y > a$, then stop and return $X = Y$, otherwise go to 1.
```

Show that this is a rejection algorithm and identify the proposal and target probability density functions. Calculate the expected number of trials before X is simulated. Why is inversion as in (a) to be preferred for $a \gg 1/\lambda$?

3. Consider the family of distributions with probability density function

$$f_{\mu,\lambda}(x) = \lambda \exp(-2\lambda|x-\mu|), \quad x \in \mathbb{R},$$

where $\lambda > 0$ and $\mu \in \mathbb{R}$ are parameters.

(a) Given $U \sim \text{Unif}(0,1)$, use the inversion method to simulate from $f_{\mu,\lambda}$.

(b) Let X have probability density function $f_{\mu,\lambda}$. Show that $a+bX$ has probability density function $f_{\mu',\lambda'}$. Find the parameters μ' and λ'.

(c) Let $Y, Z \sim \text{Exp}(r)$ independent. Show that $Y-Z$ has probability density function $f_{\mu',\lambda'}$. Find μ' and λ'. Hence, use the transformation method to simulate from $f_{\mu,\lambda}$ for any $\mu \in \mathbb{R}$ and $\lambda > 0$, given $U_1, U_2 \sim \text{Unif}(0,1)$ independent.

(d) Let $Y, Z \sim \text{Exp}(r)$ independent. Find the joint probability density function of $Y-Z$ and $Y+Z$. Are $Y-Z$ and $Y+Z$ independent?

(e) Let $Y \sim \text{Exp}(r)$ and $B \sim \text{Bernoulli}(1/2)$. Show that $a+(2B-1)Y$ has probability density function $f_{\mu',\lambda'}$. Find μ' and λ'. Hence, use the transformation method to simulate from $f_{\mu,\lambda}$ for any $\mu \in \mathbb{R}$ and $\lambda > 0$, given $U_1, U_2 \sim \text{Unif}(0,1)$ independent.

(f) (Optional `R` exercise) Implement the algorithms of (a), (c) and (e) and demonstrate statistically that they simulate from the same distribution.

4. Consider the standard normal density $f(x) = (2\pi)^{-1/2}\exp(-x^2/2)$ and $h(x) = \frac{1}{2}\exp(-|x|)$, $x \in \mathbb{R}$.

(a) Find M to bound $f(x)/h(x)$ for all $x \in \mathbb{R}$.

(b) Give a rejection algorithm simulating $X \sim f$ based on $Y_j \sim h$, $j \geq 1$.

(c) Could we reverse things and simulate $Y \sim h$ by rejection based on $X_j \sim f$, $j \geq 1$?

(d) (Optional `R` exercise) Implement your rejection algorithm for X in `R`.

A.2 Rejection and importance sampling

Please hand in scripts by Wednesday 18 May 2011, 10am, Department of Statistics.

1. Consider a discrete distribution on $\mathcal{X} = \{1, 2, \ldots, m\}$ with probability mass function $\pi(i)$, $i \in \mathcal{X}$. Let $\xi(i) = 1/m$ be the probability mass function of the uniform distribution on $\mathcal{X}$. Give a rejection algorithm using ξ as proposal distribution and π as target distribution. Calculate the expected number of trials from ξ per returned value from π if $m = 4$ and $(\pi(i), 1 \leq i \leq 4) = (0.5, 0.25, 0.125, 0.125)$.

2. The negative binomial distribution $\text{NegBin}(m, p)$ is the distribution of the number of failures in a sequence of independent $\text{Bernoulli}(p)$ trials run until the mth success.

   (a) Show that for $X \sim \text{NegBin}(m, p)$,
   $$\mathbb{P}(X = n) = \binom{m+n-1}{n}(1-p)^n p^m, \qquad n \geq 0.$$

   (b) Find a sequential inversion algorithm for the negative binomial distribution.

   (c) Show that there is no rejection algorithm for the negative binomial distribution that uses the Poisson distribution as proposal distribution.

   (d) Show that the negative binomial distribution is an overdispersed Poisson distribution of the form
   $$\mathbb{P}(X = n | R = r) = \frac{r^n}{n!}e^{-r} \qquad \text{where } R \sim \text{Gamma}\left(m, \frac{p}{1-p}\right).$$

   (e) Use (d) to set up a simulation algorithm for the negative binomial distribution based on $m+1$ independent uniform random variables $U_0, \ldots, U_m \sim \text{Unif}(0, 1)$.

   (f) Which of the two algorithms in (b) and (e) is to be preferred? Can you think of a situation where you might prefer the other algorithm?

3. Consider the following algorithm

   ```
   1. Generate independent $U_1 \sim \text{Unif}(0,1)$ and $U_2 \sim \text{Unif}(0,1)$.
   2. Set $(V_1, V_2) = (2U_1 - 1, 2U_2 - 1)$ and $S = V_1^2 + V_2^2$.
   3. If $S \leq 1$, go to 4., else go to 1.
   4. Set $P = \sqrt{-2(\ln(S))/S}$.
   5. Return the pair $(X, Y) = (PV_1, PV_2)$.
   ```

   This algorithm is called the "polar method".

   (a) Show that when the algorithm goes to 4., $V = (V_1, V_2) \sim \text{Unif}(D)$, where $D = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 < 1\}$ is the disk of radius 1.

   (b) Show that (X, Y) is a pair of independent standard normal random variables.

   (c) Compare this algorithm with the Box-Muller algorithm. How many $\text{Unif}(0, 1)$ variables are needed per $\text{Normal}(0, 1)$ variable, and what functions need to be evaluated?

4. Suppose $X \sim \text{Normal}(0, \sigma^2)$ and we want to estimate $\mu_g = \mathbb{E}(g(X))$ for some function g known to have finite mean $\mathbb{E}(g(X))$ and variance $\text{Var}(g(X))$. Suppose we have a random sample $Y_1, \ldots, Y_n \sim \text{Normal}(0, 1)$. Here are two estimators for μ_g given in terms of $Y_1, \ldots, Y_n$:

$$\widehat{\mu}_g^{(1)} = \frac{1}{n} \sum_{i=1}^{n} g(\sigma Y_i)$$

and

$$\widehat{\mu}_g^{(2)} = \frac{1}{n\sigma} \sum_{i=1}^{n} e^{-Y_i^2(1/2\sigma^2 - 1/2)} g(Y_i).$$

(a) Show that $\widehat{\mu}_g^{(1)}$ and $\widehat{\mu}_g^{(2)}$ are unbiased, and calculate their variances.

(b) Show that $\widehat{\mu}_g^{(1)}$ is a Monte-Carlo estimator of $\mathbb{E}(g(X))$, and that $\widehat{\mu}_g^{(2)}$ is an importance sampling estimator of $\mathbb{E}(g(X))$ – identify associated densities f and h.

(c) For what range of values of σ has $\widehat{\mu}_g^{(2)}$ finite variance for all g as above? Can you give a weaker condition if it is known that $\int_{-\infty}^{\infty} g^2(x)dx < \infty$?

(d) Why might we prefer $\widehat{\mu}_g^{(2)}$ to $\widehat{\mu}_g^{(1)}$, for some values of σ^2 and functions g? [*Hint: consider estimating* $\mathbb{P}(X > 1)$ *with* $\sigma \ll 1$.]

A.3 Importance sampling and MCMC

Please hand in scripts by Wednesday 25 May 2011, 10am, Department of Statistics.

1. Consider a random variable X with probability density function $f_0 : \mathbb{R} \to [0, \infty)$. We are seeking to estimate $\mathbb{E}(\phi(X)) = \int_{\mathbb{R}} \phi(x) f_0(x) dx$ for a function $\phi \colon \mathbb{R} \to \mathbb{R}$ with $\mathbb{E}(|\phi(X)|) < \infty$.

   (a) Use the Monte-Carlo method to estimate $\mathbb{E}(\phi(X))$.

   (b) Suppose that X has a moment generating function $G \colon (-a, b) \to (0, \infty)$, $G(t) = \mathbb{E}(e^{tX})$. Show that there is a family of probability distributions with probability density function $x \mapsto f_r(x)$ proportional to $x \mapsto e^{rx} f_0(x)$. Formulate associated importance sampling estimators of $\mathbb{E}(\phi(X))$.

   (c) Suppose that $X > 0$ has moments $M \colon (-c, d) \to (0, \infty)$, $M(t) = \mathbb{E}(X^t)$. Use the moments to find more importance sampling estimators of $\mathbb{E}(\phi(X))$.

   (d) Let $f_0(x) = (\Gamma(\alpha))^{-1} \lambda^\alpha x^{\alpha-1} \exp(-\lambda x)$ be the Gamma(α, λ) density. Apply (b) and (c) to find importance sampling estimators for $\mathbb{E}(\phi(X))$. In each case, identify the proposal distribution that needs simulating.

   (e) (Optional `R` exercise) Implement the algorithms in those cases where you can simulate the proposal distribution and where either $\phi = 1_{(h,\infty)}$ or $\phi = 1_{(0,h)}$.

2. A *contingency table* M is an $k \times \ell$ matrix with integer entries $m_{i,j} \geq 0$, $1 \leq i \leq k$, $1 \leq j \leq \ell$, and fixed row sums $r_1, \ldots, r_k$ and fixed column sums $c_1, \ldots, c_\ell$. Denote by $\mathcal{X} = \mathcal{X}_{r,c}$ the set of all such tables for given vectors $r = (r_1, \ldots, r_k)$ and $c = (c_1, \ldots, c_\ell)$.

   An *index table* H has zero row and column sums and entries $h_{i,j} \in \{-1, 0, 1\}$. Denote by $\mathcal{I} = \mathcal{I}_{k,\ell}$ the set of all $k \times \ell$ index tables. In the sequel, you may assume that you can simulate $U \sim \text{Unif}(\mathcal{I})$.

   Given $X_0 \in \mathcal{X}$, consider the following algorithm to modify contingency tables:

   1. Simulate $U_1 \sim \text{Unif}(\mathcal{I})$.
   2. Set $X_1 = X_0 + U_1$.

   Here are examples of contingency tables and index tables as in the algorithm:

$$M = \begin{pmatrix} 7 & 2 & 0 \\ 5 & 6 & 1 \\ 1 & 6 & 3 \\ 0 & 1 & 2 \end{pmatrix} \in \mathcal{X}, I = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{pmatrix} \in \mathcal{I}, M' = M + I = \begin{pmatrix} 7 & 2 & 0 \\ 5 & 7 & 0 \\ 1 & 5 & 4 \\ 0 & 1 & 2 \end{pmatrix} \in \mathcal{X}.$$

   (a) Is $+ \colon \mathcal{X} \times \mathcal{I} \to \mathcal{X}$ well-defined?

   (b) Modify the algorithm by rejecting disallowed transitions (i.e. staying at the same state instead) and repeatedly apply it to simulate a Markov chain $(X_n)_{n\geq 0}$ in $\mathcal{X}$ starting from $X_0 = M \in \mathcal{X}$. Specify its transition probabilities $p_{M,M'}$, $M, M' \in \mathcal{X}$.

   (c) Show that the Markov chain in (b) is reversible with stationary distribution Unif$(\mathcal{X})$.

(d) Assuming without proof that for any two $M, M' \in \mathcal{X}$ there are $I_1, \ldots, I_n \in \mathcal{I}$ with $M + I_1 + \cdots + I_n = M'$, use standard results to show that for all $M \in \mathcal{X}$

$$\mathbb{P}(X_n = M) \to 1/\#\mathcal{X}, \qquad \text{as } n \to \infty,$$

and

$$n^{-1}\#\{m \in \{0, \ldots, n-1\} : X_m = M\} \to 1/\#\mathcal{X},$$

in probability (or a.s.) as $n \to \infty$.

(e) Explain why the algorithm in (b) can be thought of as a Metropolis-Hastings algorithm with proposal transitions given by the original algorithm above.

3. (a) Give a Metropolis-Hastings algorithm to sample according to the binomial probability mass function

$$p_k = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}, \qquad k \in \{0, 1, \ldots, n\},$$

with parameters $n \in \mathbb{N}$ and $p \in (0, 1)$. Use $\mathrm{Unif}(\{0, 1, \ldots, n\})$ as proposal distribution, independent of the previous states.

(b) Give a Metropolis-Hastings algorithm to sample according to the geometric probability mass function

$$p_k = p(1-p)^{k-1}, \qquad k \in \{1, 2, \ldots\},$$

with parameter $p \in (0, 1)$. Use simple symmetric random walk as the proposal transition matrix.

(c) (Optional `R` exercise) Implement the algorithms of (a) and (b) and check that they work by comparing estimated means and variances with the known theoretical means and variances of theses distributions.