Lectures on stochastic simulation

Matti Vihola

March 5, 2020

Copyright © 2020 by Matti Vihola
and University of Jyväskylä

Preface

This is a summary of the lectures of the Spring 2020 course "MATS442 Stochastic simulation" at Department of Mathematics and Statistics, University of Jyväskylä. These notes are inspired by the lecture notes of Antti Penttinen [22], Geoff Nicholls [19] Adam Johansen, Ludger Evers and Nick Whiteley [12], and by the textbooks [25, 4, 21].

The purpose of these notes is to support the lectures, so they may not be well suited for self study. Some important methods (and examples) are covered also in the problems classes. The "Monte Carlo Statistical Methods" book by Christian P. Robert and George Casella [24, 25] is a recommended supporting material. Other references to the literature are given during the course regarding more specific topics.

1 Introduction

Simulation of stochastic systems provides powerful tools to inspect complex models. Monte Carlo methods use simulations in order to approximate expectations and probabilities related to (nearly) arbitrary models. The methods have been used (in the modern sense) already from the 1950s, and by the increase of computational power and the methodological advances over the years, they have become central tools in many applications. The analysis, efficient implementation and development of simulation methods are all active research areas. The simulation methods tend to rely on a handful of elegant key ideas, many of which are touched within this course.

Prerequisities

The course requires background in

- basic (vector) calculus (differentiation, integration),
- basic probability (probabilities, expectation, conditioning, joint distributions...), and
- knowledge of standard limit theorems in probability (law of large numbers, central limit theorem).

Basic programming skills are also useful.

Learning outcomes

After taking this course, you will be able to:

- *apply* simulation methods in practice,
- *understand* why the methods work, (and why they sometimes work poorly),
- *modify* the methods and *combine* them for your needs (for some specific application).

Theory of stochastic simulation may be categorised as *applied probability*, and the application of the methods in practice as *computational statistics*.

The programming environment

In the lectures, we focus on methods (algorithms) and theory behind them, and in the exercises we also implement the algorithms and experiment them in practice.

We are using primarily the Julia programming language in the course. If you are not familiar with Julia, that is not a problem, as basic use is simple and similar to R/Matlab/Python, and no advanced programming skills will be needed. There is also plenty of online introductory material available for self-study. Solutions to implementation problems may be returned also using another programming language, such as R or Python.

Why Julia? Because it is fast, which is essential because many simulation-based methods are computationally intensive...

1.1 Conventions

We will generally use the symbols $\mathbb{P}$ and $\mathbb{E}$ for probability and expectation, denote random variables[1] with capital letters, and $\mathbf{1}(\cdot)$ stands for the characteristic function (e.g. $\mathbf{1}(X \in A) = 1$ if $X \in A$ and $\mathbf{1}(X \in A) = 0$ otherwise).

In this course, we focus on the two common types of distributions p:

continuous distribution defined by a probability density function (p.d.f.) p on a space $\mathbb{X} = \mathbb{R}^d$.

discrete distribution defined by a probability mass function (p.m.f.) p on a finite space $\mathbb{X} = \{x_1, \ldots, x_m\}$ or a countably infinite space $\mathbb{X} = \{x_1, x_2, \ldots, \}$.

The notation $X \sim p$ means that X is a random variable has distribution p and $X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} p$ or $(X_k)_{k\geq 1} \overset{\text{i.i.d.}}{\sim} p$ means $(X_k)_{k\geq 1}$ are independent and each $X_k \sim p$.

Remark 1.1. The $\mathbb{X} = \mathbb{R}^d$ then $X \sim p$ is a *random vector* with distribution p, that is, the coordinates $X^{(1)}, \ldots, X^{(d)}$ are *random numbers* with joint density $p(x^{(1)}, \ldots, x^{(d)})$.

For $f : \mathbb{X} \to \mathbb{R}$, we write $\mathbb{E}_p[f(X)]$ meaning the expectation of $f(X)$ when $X \sim p$. That is,

$$\mathbb{E}_p[f(X)] = \begin{cases} \int_{\mathbb{X}} f(x)p(x)\mathrm{d}x, & \text{if } X \text{ is continuous (and } p \text{ is a p.d.f.)} \\ \sum_{x\in\mathbb{X}} f(x)p(x), & \text{if } X \text{ is discrete (and } p \text{ is a p.m.f.)} \end{cases}$$

We also write similarly $\mathrm{Var}_p\big(f(X)\big) = \mathbb{E}_p\big[(f(X) - \mathbb{E}_p[f(X)])^2\big] = \mathbb{E}_p\big[f^2(X)\big] - (\mathbb{E}_p[f(X)])^2$.

Remark 1.2. If $\mathbb{X} = \mathbb{R}^d$, we mean that

$$\int_{\mathbb{X}} f(x)p(x)\mathrm{d}x = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(x^{(1)}, \ldots, x^{(d)})p(x^{(1)}, \ldots, x^{(d)})\mathrm{d}x^{(1)} \cdots \mathrm{d}x^{(d)}.$$

We shall omit the domain of integration in most cases if we integrate over the whole space $\mathbb{X}$. For instance, if $\mathbb{X} = \mathbb{R}$, we may write

$$\int f(x)p(x)\mathrm{d}x = \int_{\mathbb{X}} f(x)p(x)\mathrm{d}x = \int_{-\infty}^{\infty} f(x)p(x)\mathrm{d}x.$$

---

1. We use the term 'random variable' regardless of dimension and nature (vector, numbers, integers etc.).

(NB: Not to be confused with the indefinite integral!)

Remark 1.3 (*, starred sections, remarks etc. such as this are *non-examinable* extra material, which may be safely skipped). For those who are familiar with general probability theory, the integral above can also be taken with respect to an arbitrary (sigma-finite) measure "dx" on a general measurable space $\mathbb{X}$ (instead of the Lebesgue measure on $\mathbb{R}^d$ equipped with the Borel sets, or the counting measure on countable $\mathbb{X}$ equipped with the power set). Then, p is the density (Radon-Nikodym derivative) of the distribution of interest with respect to dx. Note also that if μ is a probability measure of interest on $\mathbb{X}$, we may also take itself as the dominating measure d$x = \mu(\mathrm{d}x)$, in which case $p \equiv 1$. Most of the techniques presented in the course generalise into such a setting directly.

The functions $f : \mathbb{X} \to \mathbb{R}$ for which expectations are defined must, of course, be measurable. We shall implicitly assume the required measurability of any such test function (or set), without explicit notification.

We do not explicitly state the probaiblity space where the random variables etc. are defined. Instead, we either work with countable sequences of independent and identically distributed random variables, or more generally, dependent sequences defined by conditional probabilities, such as discrete-time Markov chains. For such countable 'algorithmic' definitions, the existence of the underlying probability space is standard (using the Ionescu-Tulcea extension theorem).

1.2 The Monte Carlo method

Let us start by a very simple but illustrative example.

Example 1.4 (Finding approximation of π by simulation). Suppose 'rain drops' fall uniformly in a 2×2 metre square. Let us check how they could be used to determine an approximation of π.

Probability of one drop hitting a unit radius disc inside the square

$$\beta = \frac{\text{area of disc}}{\text{area of square}} = \frac{\pi}{4}$$

If $H_k = 1$ if the drop hit the unit disc and $H_k = 0$ otherwise[2], then

$$4\Big(\frac{1}{n}\sum_{k=1}^{n} H_k\Big) \xrightarrow{n\to\infty} \pi, \qquad \text{(almost surely)}$$

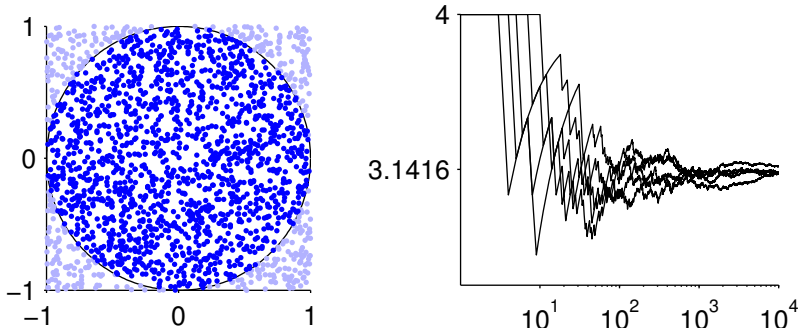
by the (strong) law of large numbers.

Definition 1.5 (Monte Carlo)**.** Assume $X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} p$. Then, for $f : \mathbb{X} \to \mathbb{R}$,

$$I_p^{(n)}(f) := \frac{1}{n}\sum_{k=1}^{n} f(X_k) \tag{1}$$

is the *Monte Carlo approximation* of $\mathbb{E}_p[f(X)]$.

2. $H_k \sim$ Bernoulli(β): $\mathbb{P}(H_k = 1) = \beta$, $\mathbb{P}(H_k = 0) = 1 - \beta$.



(a) (b)

Figure 1: The rain drops example: (a) 'rain drops' falling inside the unit disc highlighted (b) five realisations of the experiment: estimates of π converge as $n \to \infty$ (note the log scale on n).

Example 1.6. Example 1.4 corresponds to

- $\mathbb{X} = \mathbb{R}^2$
- $(X_k)_{k \geq 1} \overset{\text{i.i.d.}}{\sim} \mathcal{U}([-1, 1]^2)$,
- $f(x) = 4 \cdot \mathbf{1}\left(\|x\| \leq 1\right)$ or $f\big(x^{(1)}, x^{(2)}\big) = 4 \cdot \mathbf{1}\left((x^{(1)})^2 + (x^{(2)})^2 \leq 1\right)$,

or, equivalently, to simulated Bernoulli random variables,

- $\mathbb{X} = \{0, 1\}$
- $(H_k)_{k \geq 1} \overset{\text{i.i.d.}}{\sim}$ Bernoulli$(\pi/4)$ (NB: We simulate H_k above as $H_k = f(X_k)$, which does not require us to evaluate $\pi/4$!)
- $f(h) = 4h$.

In both cases, $\mathbb{E}_p[f(X)] = \pi$.

Remark 1.7. If $\mathbb{X}$ is finite, then $\mathbb{E}_p[f(X)]$ is a finite sum and can, in principle, be computed exactly. However, we might not be able to calculate $p(x)$ exactly (cf. Example 1.6), or the space $\mathbb{X}$ may have a huge number of elements, for example if $\mathbb{X}$ is the set of all 100×100 binary images $\mathbb{X} = \{0, 1\}^{100 \times 100}$, in which case the Monte Carlo method may still be relevant...

Example 1.8. Definition 1.5 allows for calculating:

- Probabilities: $\mathbb{P}(X \in A)$ for $X \sim p$, by choosing $f(x) = \mathbf{1}\left(x \in A\right)$ (cf. Example 1.4).
- Multiple expectations (or probabilities) simultanously: $E_p[f_k(X)]$ for a number of test functions $f_1, \ldots, f_m$. For example, the mean of random vector $X = \big(X^{(1)}, \ldots, X^{(d)}\big) \sim p$ is a vector of means of individual coordinates,

$$\mathbb{E}_p[X] = \big(\mathbb{E}_p[f_1(X)], \ldots, \mathbb{E}_p[f_d(X)]\big),$$

  where $f_k\big(x^{(1)}, \ldots, x^{(d)}\big) = x^{(k)}$.

  Note that we may simulate $X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} p$ and construct all $I_p^{(n)}(f_1), \ldots, I_p^{(n)}(f_d)$ using the *same* samples $X_1, \ldots, X_n$.

1.3 Properties of Monte Carlo estimators

We need the strong law of large numbers and the central limit theorem frequently, so we shall restate them here without proof.

Theorem 1.9 (Strong law of large numbers)**.** *Assume $Y_1, Y_2, \ldots$ are i.i.d. random numbers such that $\mu = \mathbb{E}[Y_1]$ is finite. Then,*

$$\frac{1}{n}\sum_{k=1}^{n} Y_k \xrightarrow{n\to\infty} \mu \qquad \text{a.s. (almost surely).} \tag{2}$$

Remark 1.10. Recall that $Z_n \to Z$ a.s. $\implies$ $Z_n \to Z$ in probability $\implies$ $Z_n \to Z$ in distribution;

$$\begin{aligned}
Z_n \to Z \text{ a.s.} \quad &\iff \quad \mathbb{P}(Z_n \xrightarrow{n\to\infty} Z) = 1 \\
Z_n \to Z \text{ in probability} \quad &\iff \quad \text{For all } \epsilon > 0,\ \mathbb{P}(|Z_n - Z| \le \epsilon) \xrightarrow{n\to\infty} 1 \\
Z_n \to Z \text{ in distribution} \quad &\iff \quad \text{For all continuity points } t \text{ of the mapping } t \mapsto \mathbb{P}(Z \le t), \\
&\qquad\qquad \mathbb{P}(Z_n \le t) \xrightarrow{n\to\infty} \mathbb{P}(Z \le t),
\end{aligned}$$

In particular, you may always replace "almost surely" in (2) by "in probability," but not vice versa. (Theorem 1.9 with "in probability" instead of "a.s." is known as the *weak* law of large numbers.)

Theorem 1.11 (Central limit theorem)**.** *Assume $Y_1, Y_2, \ldots$ are i.i.d. random numbers with $\sigma^2 := \mathrm{Var}\big(Y_1\big) \in (0, \infty)$, then with $\mu = \mathbb{E}[Y_k]$,*

$$\frac{1}{\sigma\sqrt{n}}\sum_{k=1}^{n}(Y_k - \mu) \xrightarrow{n\to\infty} N(0,1) \qquad \text{in distribution,}$$

in other words,

$$\mathbb{P}\left(\frac{1}{\sigma\sqrt{n}}\sum_{k=1}^{n}(Y_k - \mu) \le t\right) \xrightarrow{n\to\infty} \Phi(t) \qquad \text{for all } t \in \mathbb{R},$$

where Φ is the standard normal c.d.f., that is, $\Phi(t) := \mathbb{P}(Z \le t)$ with $Z \sim N(0,1)$.

Proposition 1.12. *The Monte Carlo estimators satisfy the following properties:*
Unbiasedness *If $\mathbb{E}_p[f(X)]$ is finite, then $\mathbb{E}[I_p^{(n)}(f)] = \mathbb{E}_p[f(X)]$ for all $n \ge 1$.*
Strong consistency *If $\mathbb{E}_p[f(X)]$ is finite, then $I_p^{(n)}(f) \xrightarrow{n\to\infty} \mathbb{E}_p[f(X)]$ almost surely.*
Variance *If $\mathrm{Var}_p[f(X)] < \infty$, then $\mathrm{Var}[I_p^{(n)}(f)] = \frac{1}{n}\mathrm{Var}_p[f(X)]$.*

Proof. Let $Y_k = f(X_k)$, then $\mathbb{E}[Y_k] = \mathbb{E}_p[f(X)]$. Now,

$$\mathbb{E}[I_p^{(n)}(f)] = \frac{1}{n}\sum_{k=1}^{n}\mathbb{E}[Y_k] = \mathbb{E}_p[f(X)].$$

Strong consistency follows from application of the strong law of large numbers, because $Y_k := f(X_k)$ are i.i.d. random variables with expectation $\mathbb{E}_p[f(X)]$. Finally,

$$\mathrm{Var}[I_p^{(n)}(f)] = \frac{1}{n^2}\mathrm{Var}\Big(\sum_{k=1}^{n} Y_k\Big) = \frac{1}{n}\mathrm{Var}(Y_1). \qquad \square$$

Proposition 1.13 (Asymptotic Monte Carlo error)**.** *Assume* $(X_k)_{k\geq 1} \overset{i.i.d.}{\sim} p$ *and* $f : \mathbb{X} \to \mathbb{R}$ *is such that with* $\sigma^2 := \mathrm{Var}_p\big(f(X_1)\big) \in (0, \infty)$*,*

(i) $\sqrt{n}\big[I_p^{(n)}(f) - \mathbb{E}_p[f(X)]\big] \xrightarrow{n\to\infty} N(0, \sigma^2)$ *in distribution.*

Furthermore, letting $\hat{\sigma}_n^2$ *stand for the sample variance:*

$$\hat{\sigma}_n^2 := \frac{1}{n-1}\sum_{k=1}^{n}\big(f(X_k) - I_p^{(n)}(f)\big)^2;$$

(ii) for any $\beta \in \mathbb{R}$*,*

$$\mathbb{P}\big(\sqrt{n}\big[I_p^{(n)}(f) - \mathbb{E}_p[f(X)]\big] \leq \beta\hat{\sigma}_n\big) \xrightarrow{n\to\infty} \Phi(\beta), \text{ and}$$

(iii) for any $\alpha \in (0, \infty)$*, the following confidence interval is consistent:*

$$\mathbb{P}\left(\mathbb{E}_p[f(X)] \in \left[I_p^{(n)}(f) \pm \alpha\frac{\hat{\sigma}_n}{\sqrt{n}}\right]\right) \xrightarrow{n\to\infty} 1 - 2\Phi(-\alpha).$$

Recall the following lemma for the proof:

Lemma 1.14 (Slutsky)**.** *Suppose the random numbers* $X_n \to X$ *in distribution and* $Y_n \to y$ *in probability, where* $y \in \mathbb{R}$ *is a constant, then:*

(i) $X_nY_n \to Xy$ *in distribution.*

(ii) If $y \neq 0$*, then* $X_n/Y_n \to X/y$ *in distribution.*

Proof of Proposition 1.13. (i) is an application of the CLT with $Y_k := f(X_k)$, and because $\hat{\sigma}^2 \to \sigma^2$ almost surely (and in probability), (ii) follows by Lemma 1.14.

Consider then (iii), and observe that

$$\begin{aligned}&\mathbb{P}\left(\mathbb{E}_p[f(X)] \in \left[I_p^{(n)}(f) \pm \alpha\frac{\hat{\sigma}_n}{\sqrt{n}}\right]\right)\\ &\quad= \mathbb{P}\left(I_p^{(n)}(f) - \mathbb{E}_p[f(X)] \leq \alpha\frac{\hat{\sigma}_n}{\sqrt{n}}\right) - \mathbb{P}\left(I_p^{(n)}(f) - \mathbb{E}_p[f(X)] < -\alpha\frac{\hat{\sigma}_n}{\sqrt{n}}\right).\end{aligned}$$

The first term converges to $\Phi(\alpha) = 1 - \Phi(-\alpha)$ by (ii). The second can be sandwiched between $\Phi(-\alpha - \epsilon)$ and $\Phi(-\alpha)$ for arbitrary $\epsilon > 0$. □

Remark 1.15. Proposition 1.13 is an asymptotic result, so it does not give any guarantees for a finite n. In practice, the approximation is often informative for moderate α and large n.

Remark 1.16. The variance expression of Proposition 1.12 can be used directly to build non-asymptotic upper bounds by Chebychev's inequality,

$$\mathbb{P}(|I_p^{(n)}(f) - \mathbb{E}_p[f(X)]| \geq \epsilon) \leq \frac{\mathrm{Var}[I_p^{(n)}(f)]}{\epsilon^2} = \frac{\mathrm{Var}_p[f(X)]}{n\epsilon^2} \qquad \text{for all } \epsilon > 0.$$

Note that we need to know $\mathrm{Var}_p[f(X)]$, or we need to be able to upper bound $\mathrm{Var}_p[f(X)]$, in order to use this bound.

Remark 1.17 (*). The Chebychev bound is rather pessimistic for the tails: if we set $\epsilon = t/\sqrt{n}$, then the bound is $O(t^{-2})$ for large t. If more is known about $f(X)$, tighter tail bounds are possible. For instance, in the bounded case $|f(X) - \mathbb{E}_p[f(X)]| \leq c$, a Hoeffding inequality implies

$$\mathbb{P}(|I_p^{(n)}(f) - \mathbb{E}_p[f(X)]| \geq \epsilon) \leq 2\exp\big(-2\epsilon^2 n/c^2\big),$$

and therefore for $\epsilon = t/\sqrt{n}$, we get $O(e^{-2t^2/c^2})$ bound.

1.4 About uniformly distributed pseudo-random numbers

During this course, we shall assume that we can access $(U_k)_{k\geq 1} \overset{\text{i.i.d.}}{\sim} \mathcal{U}(0,1)$, a sequence of independent random variables uniformly distributed on the interval $(0,1)$. All algorithms are based on these random variables, and all theoretical results given below rely on this (rather strong) assumption.

In practice, when the algorithms are implemented on a computer, the sequence $(U_k)_{k\geq 1}$ are not going to be random, but *pseudo-random*. That is, $(U_k)_{k\geq 1}$ are in fact produced by a *deterministic* recursive algorithm with a finite state, a pseudo-random number generator (PRNG). Setting a *seed* of the algorithm means that we set the state variables of the algorithm to given initial values. The sequence $(U_k)_{k\geq 1}$ is entirely determined by the seed. However, a good PRNG approximates 'true randomness' rather well (is indistinguishable by a wide range of statistical tests).

It is essential to use a good PRNG for stochastic simulation, such as the *Mersenne twister* [16], which is the default PRNG for many environments, including Julia, Matlab, R and Python, and there are free implementations for most other environments. Remember also to seed your algorithm, if your implementation does not do that automatically.

1.5 Monte Carlo vs. other numerical integration methods

There are several other numerical integration methods, which may be used to calculate expectations instead of the Monte Carlo method. It is not straightforward to say which method works the best for a given problem, but here are some thoughts about the strengths and weaknesses of the Monte Carlo method:

+ Monte Carlo methods are generally applicable. For instance, the functions f and p need not be continuous, differentiable etc.
+ Monte Carlo is often easy to implement.
+ Monte Carlo can work well in multiple dimensions, where grid-based methods can be inefficient/inapplicable. This is supported by the "$O(n^{-1/2})$ rate of convergence" which is independent of the dimension.
− Even though the MC rate is usually $O(n^{-1/2}$, the constants involved may grow exponentially in dimension. (That is, MC does not generally 'beat the curse of dimensionality')
− Deterministic methods may have better rate of convergence than the Monte Carlo rate $n^{-1/2}$ (but may also deteriorate faster when dimension increases).
− Monte Carlo estimate is always random, so we never have guaranteed tolerance, but only statistical evidence (consistent confidence intervals at best).

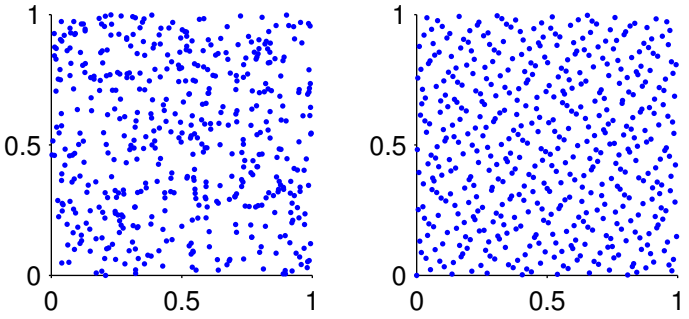


Figure 2: 500 points on $[0,1]^2$ which are (a) i.i.d. pseudo-random (b) from a low-discrepancy sequence (Halton).

Remark 1.18 (*). It may be good to know that there are also so-called *quasi Monte Carlo* methods, which may behave better in some applications (they often have a better rate of convergence). They are similar to Monte Carlo (based on averages), but instead of using i.i.d. (pseudo-)random variables $(U_k)_{k\geq 1}$, they use specifically designed 'low-discrepancy sequences' which 'fill' up the unit interval (or unit hypercube) in a deterministic way so that the points are scattered in a 'uniform' manner; see Figure 2.

We do not consider QMC methods further in the course, but note that QMC is also active in research, and succesful combinations of (randomised) QMC and MC have been suggested recently...

2 Variable transformation methods

Obviously, many interesting Monte Carlo problems assume that $(X_k)_{k\geq 1} \overset{\text{i.i.d.}}{\sim} p$, where p is not $\mathcal{U}(0,1)$. We need methods to transform $(U_k)_{k\geq 1} \sim \mathcal{U}(0,1)$ into $(X_k)_{k\geq 1}$. In this section, we consider methods that

- Transform single $U \sim \mathcal{U}(0,1)$ into a single $X \sim p$.
- Transform multiple $U_1, \ldots, U_n \overset{\text{i.i.d.}}{\sim} \mathcal{U}(0,1)$ into single or multiple $X_1, \ldots, X_m \overset{\text{i.i.d.}}{\sim} p$, where $1 \leq m \leq n$.

2.1 Inverse distribution function method

Recall that the (cumulative) distribution function (c.d.f.) F of a random variable X is defined as $F(x) := \mathbb{P}(X \leq x)$ for all $x \in \mathbb{R}$. Recall also that if X has density p, then

$$F(x) = \int_{-\infty}^{x} p(t)\mathrm{d}t.$$

Theorem 2.1. *Assume $U \sim \mathcal{U}(0,1)$ and let $F : A \to (0,1)$ be a c.d.f. on an open interval[3] $A \subset \mathbb{R}$, which is continuous and strictly increasing, with inverse $F^{-1} : (0,1) \to A$. Then, $X := F^{-1}(U) \sim F$, that is, X has the c.d.f. F.*

3. May be infinite: (a,b), (a,∞), $(-\infty,b)$ or $\mathbb{R}$.

Proof. A direct calculation shows that $\mathbb{P}(X \leq x) = F(x)$ for all $x \in A$:

$$\begin{aligned}\mathbb{P}(X \leq x) &= \mathbb{P}(F^{-1}(U) \leq x) \\ &= \int_0^1 \mathbf{1}\left(F^{-1}(u) \leq x\right) \mathrm{d}u \\ &= \int_0^1 \mathbf{1}\left(u \leq F(x)\right) \mathrm{d}u = F(x). \qquad \square\end{aligned}$$

Example 2.2. If we want $X \sim \mathrm{Exp}(r)$, that is, $X \sim p(x)$ with

$$p(x) = r \exp(-rx)\mathbf{1}\left(x \geq 0\right),$$

then the c.d.f. is for $x > 0$

$$F(x) = \int_0^x r \exp(-rt)\mathrm{d}t = 1 - \exp(-rx),$$

with inverse $F^{-1}(u) = -\log(1-u)/r$. The algorithm is

(i) $U \sim \mathcal{U}(0,1)$

(ii) $X := -\log(U)/r$,

because if $U \sim \mathcal{U}(0,1)$, then also $1 - U \sim \mathcal{U}(0,1)$.

```
n = 1000          # Number of samples to simulate
u = rand(n)       # Vector of n independent U(0,1)
x = -log.(u)/2    # Vector of n indepedent Exp(2)
```

Theorem 2.3. *Assume p is a p.m.f. on $\mathbb{X} = \{x_1, x_2, \ldots\}$. Suppose $U \sim \mathcal{U}(0,1)$ and define the random variable*

$$K := \min\left\{k \geq 1 \,:\, \sum_{j=1}^{k} p(x_j) \geq U\right\}.$$

Then, $X := x_K$ has distribution p.

Proof. Define $F(k) := \sum_{j=1}^{k} p(x_j)$ with $F(0) := 0$, and note that

$$\mathbb{P}(K = k) = \mathbb{P}\big(F(k-1) < U \leq F(k)\big) = F(k) - F(k-1) = p(x_k),$$

and therefore $\mathbb{P}(X = x_k) = \mathbb{P}(K = k) = p(x_k)$. $\square$

Example 2.4. If $0 < \tilde{p} < 1$ and $\tilde{q} = 1 - \tilde{p}$, and we want to simulate $X \sim$ Geometric$(\tilde{p})$ then

$$p(k) = \tilde{p}\tilde{q}^{k-1}, \qquad k \in \mathbb{N} = \{1, 2, \ldots\}$$

with

$$F(k) = \sum_{i=1}^{k} p(i) = 1 - \tilde{q}^k.$$

Smallest k giving $1 - \tilde{q}^k \geq u$ is

$$k = \left\lceil \frac{\log(1-u)}{\log(\tilde{q})} \right\rceil$$

where $\lceil x \rceil$ rounds up (smallest integer not less than x).

```
n = 1000;  q = 3/4
u = rand(n)
x = ceil.(log.(u)/log(q))
```

In fact, both the continuous and the discrete case follow as special cases from the following general inverse c.d.f. result.

Theorem 2.5 (*). *Assume $U \sim \mathcal{U}(0,1)$ and let $F : \mathbb{R} \to [0,1]$ be a c.d.f.*[4]*. Define*

$$X := F^{-1}(U) \qquad \text{where}$$
$$F^{-1}(u) := \min\{x \in \mathbb{R} \ : \ F(x) \geq u\} \quad \text{for } 0 < u < 1.$$[6]

Then, $X \sim F$, that is, X has c.d.f. F.

Proof. Recall that a c.d.f. F is increasing and right-continuous (which implies that the the min above is well-defined). The proof follows as in the proof of Theorem 2.1, by noticing that

$$F^{-1}(u) \leq x \iff u \leq F(x) \qquad \text{for all } x \in \mathbb{R} \text{ and } u \in (0,1).$$

Namely, suppose $F^{-1}(u) \leq x$ and denote $x_u := F^{-1}(u) \leq x$, then $F(x) \geq F(x_u) \geq u$. Conversely, if $u \leq F(x)$, then $F^{-1}(u) = \min\{y \in \mathbb{R} : F(y) \geq u\} \leq x$, because x is included in the set which is minimised. □

Example 2.6 (*). Consider the following c.d.f.:

$$F(x) := \left(\frac{1}{2} + \frac{1}{2}\big(1 - \exp(-x)\big)\right)\mathbf{1}\,(x \geq 0)\,.$$

Its generalised inverse is

$$F^{-1}(u) = -\log\big(2(1-u)\big)\mathbf{1}\,(u > 1/2)\,.$$

We may replace U with $1 - U$ again, resulting in the following:

```
u = rand(1000)
x = -log.(2u) .* (u .<= 1/2)
```

2.2 Distribution of transformed random variables

The inverse c.d.f. method provides a general result to transform $\mathcal{U}(0,1)$ random variables into scalar random variables, provided that the (inverse) c.d.f. is accessible. In a multivariate setting, or when c.d.f. is inaccessible, other transformations can be useful.

---

4. Recall that F is a c.d.f. if it is increasing[5], right-continuous, $\lim_{x\to-\infty} F(x) = 0$ and $\lim_{x\to\infty} F(x) = 1$.
6. The function F^{-1} is called the *generalised inverse c.d.f.*.

Suppose that $X \sim p_X$, the function $g : \mathbb{R} \to \mathbb{R}$ is strictly increasing and continuously differentiable. Let $Y = g(X)$, then

$$F_Y(y) := \mathbb{P}(Y \le y) = \mathbb{P}\big(g(X) \le y\big) = F_X\big(g^{-1}(y)\big),$$

where $F_X(x) := P(X \le x)$ is the c.d.f. of X. Now, the p.d.f. of Y is

$$p_Y(y) = F_Y'(y) = F_X'\big(g^{-1}(y)\big)(g^{-1})'(y) = \frac{p_X\big(g^{-1}(y)\big)}{g'(g^{-1}(y))},$$

because $(g^{-1})'(y) = 1/g'\big(g^{-1}(y)\big)$.

Recall the following multivariate generalisation of the above, which we use without proof.

Theorem 2.7. *Suppose $X \sim p_X$ and $S := \operatorname{supp}(p) := \{x \in \mathbb{R}^d : p_X(x) > 0\}$ is an open set. If $g : S \to \mathbb{R}^d$ is one-to-one and continuously differentiable such that its Jacobian Dg is invertible, $\det(Dg(x)) \neq 0$ for all $x \in S$, then $Y = g(X)$ has density p_Y given as follows,*

$$p_Y(y) = \begin{cases} p_X(g^{-1}(y))|\det(Dg^{-1})(y)|, & y \in g(S) \\ 0, & y \notin g(S), \end{cases}$$

where Dg^{-1} stands for the Jacobian of g^{-1}.

Remark 2.8. By the inverse function theorem, for all $y \in g(S)$,

$$(Dg^{-1})(y) = [(Dg)(x)]^{-1},$$

where $y = g(x)$ (or $x = g^{-1}(y)$). Also, $\det(A^{-1}) = 1/\det(A)$, so we have

$$|\det(Dg^{-1})(y)| = \frac{1}{|\det(Dg)(x)|}.$$

Remark 2.9 (*). If $\operatorname{supp}(p)$ can be partitioned (up to set of volume (measure) zero) into disjoint open sets $S_1, S_2, \ldots$ such that g satisfies the conditions required in Theorem 2.7, then Theorem 2.7 can be applied piecewise, leading into

$$p_Y(y) = \sum_i p_X(g^{-1}(y))|\det(Dg^{-1}(y)|\mathbf{1}\left(y \in g(S_i)\right).$$

2.3 (Multivariate) normal random variables

Normal distribution is, of course, particularly important in applications. The inverse c.d.f. method is not (directly) applicable because the c.d.f. is not available in a closed form. However, it is possible to generate normal random variables by a simple bivariate transformation.

Recall that the standard normal $N(0,1)$ p.d.f. is

$$p(x) = \frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{x^2}{2}\Big),$$

and the general multivariate normal $N(\mu, \Sigma)$ p.d.f. is

$$p(x) = \frac{1}{(2\pi)^{d/2}\sqrt{\det(\Sigma)}} \exp\Big(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\Big).$$

The random vector $X := (X_1, \ldots, X_d)^T$, where $(X_k) \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ is distributed by $\mathcal{N}(0, I_d)$, that is, the standard multivariate Gaussian distribution with zero mean vector and identity covariance matrix.

Theorem 2.10 (Box-Muller transform)**.** *Let* $U_1, U_2 \overset{i.i.d.}{\sim} \mathcal{U}(0,1)$ *and define*

$$\begin{array}{ll} X_1 := R\cos(T) & \\ X_2 := R\sin(T), & \textit{where} \end{array} \qquad \begin{array}{ll} R := \sqrt{-2\ln U_1} & \\ T := 2\pi U_2. & \end{array}$$

Then, $X_1, X_2 \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$.

Proof. The density of (R, T) is (exercise)

$$p_{R,T}(r,t) = \begin{cases} \frac{1}{2\pi} r e^{-r^2/2}, & 0 < t < 2\pi, 0 < r < \infty, \\ 0, & \text{otherwise.} \end{cases}$$

Now, $(X, Y) = h(R, T)$ with $h(r,t) := (r\cos t, r\sin t)$ (polar-to-Cartesian transform), with

$$|\det(Dh)(r,t)| = \left| \det \begin{pmatrix} \cos t, & -r\sin t \\ \sin t, & r\cos t \end{pmatrix} \right| = r.$$

Now we may apply Theorem 2.7 and Remark 2.8 to deduce that

$$p_{X,Y}(x,y) = p_{R,T}(r(x,y), t(x,y)) \frac{1}{r(x,y)} = \frac{1}{2\pi} e^{-\frac{1}{2}(x^2+y^2)}, \quad (x,y) \neq 0,$$

where $r(x,y) := \sqrt{x^2+y^2}$ and $t(x,y) := \operatorname{atan2}(y,x)$. □

Proposition 2.11 (Generic multivariate Gaussian distribution)**.** *Let* $\mu \in \mathbb{R}^d$ *and* $\Sigma \in \mathbb{R}^{d\times d}$ *be a positive definite matrix, and let* $L \in \mathbb{R}^{d\times d}$ *be the Cholesky factor of* Σ *(lower-triangular matrix satisfying* $LL^T = \Sigma$*). Then, if* $Z \sim N(0, I_d)$*,*

$$X := \mu + LZ \qquad \textit{satisfies} \qquad X \sim \mathcal{N}(\mu, \Sigma). \tag{3}$$

Proof. The Jacobian of $g(z) = \mu + Lz$ is $|\det(L)| = \sqrt{\det(\Sigma)} > 0$ and the inverse $g^{-1}(x) = L^{-1}(x-\mu)$.

$$\begin{aligned} p_X(x) &= p_Z(L^{-1}(x-\mu))/\sqrt{\det(\Sigma)} \\ &= \frac{1}{(2\pi)^{d/2}\sqrt{\det(\Sigma)}} \exp\Big(-\frac{1}{2}(x-\mu)^T(L^{-1})^T L^{-1}(x-\mu)\Big), \end{aligned}$$

and $(L^{-1})^T L^{-1} = (L^T)^{-1}L^{-1} = (LL^T)^{-1} = \Sigma^{-1}$. □

Remark 2.12. We could use, of course, any matrix $L \in \mathbb{R}^{d\times d}$ satisfying $LL^T = \Sigma$, but the Cholesky factor is both easy to compute and the lower-triagular structure allows for some savings when computing the transform (3).

Example 2.13. Generating bivariate Gaussians.

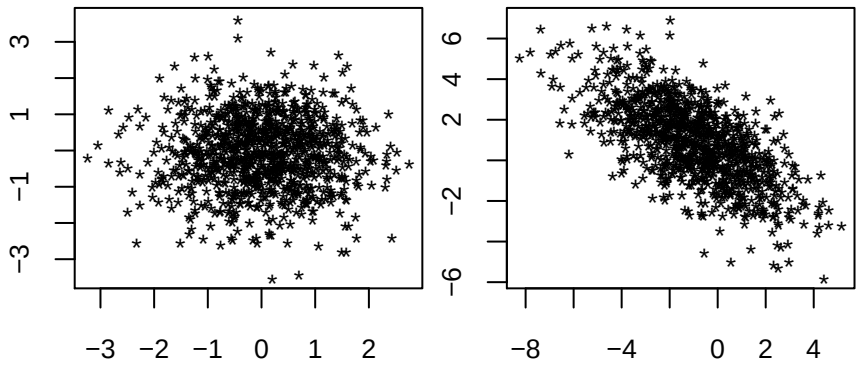


Figure 3: Standard bivariate samples $(Z_k)_{k\geq 1}$ (left) and $N(m, S)$ samples (X_k) generated in Example 2.13.

```
using LinearAlgebra
n = 1000; d = 2   # Number of samples & dimension
m = [-1,1]        # Mean vector
S = [5 -3; -3 4]  # Covariance matrix
L = cholesky(S).L # (Lower-triangular) Cholesky factor
X = zeros(d, n)   # Initialise output space
for k = 1:n
  X[:,k] = m + L*randn(d)
end
```

2.4 Relations of probability distributions (*)

Known relationships between probability distributions may yield useful transformations.

Example 2.14. [Gamma distribution with integer shape] Consider $\Gamma(\alpha, \beta)$ distribution with $\alpha \in \mathbb{N}$ and $\beta > 0$ with p.d.f.

$$p(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} \mathbf{1}\left(x \geq 0\right).$$

Inverse c.d.f. method is not easily applicable. Instead,

(a) Simulate $Y_1, \ldots, Y_\alpha \overset{\text{i.i.d.}}{\sim} \text{Exp}(1)$.
(b) Set $X := \frac{1}{\beta} \sum_{i=1}^{\alpha} Y_i$.

Then $X \sim p$.

Proof. We can check that $X \sim p$ by inspecting moment generating functions. The m.g.f. of $Y \sim \text{Exp}(1)$ is

$$M_Y(t) = \mathbb{E}\left(e^{tY}\right) = \frac{1}{1-t}, \qquad t \in [0, 1),$$

so the m.g.f. of X is

$$M_X(t) = \mathbb{E}\left(e^{tX}\right) = \prod_{i=1}^{\alpha} \mathbb{E}\left(e^{tY_i/\beta}\right) = \prod_{i=1}^{\alpha} M_{Y_i}(t/\beta) = \frac{1}{(1-t/\beta)^\alpha},$$

for $t \in [0, \beta)$, which is the m.g.f. of $\Gamma(\alpha, \beta)$. □

2.5 Spherically/elliptically symmetric distributions (*)

Example 2.15 (Uniform distribution on a $(d-1)$-sphere). Suppose $X \sim N(0, I)$, a standard Gaussian distribution in $\mathbb{R}^d$. Then, $V = X/\|X\| \sim \mathcal{U}(S^{d-1})$, that is, V is uniformly distributed on the unit sphere $S^{d-1} := \{x \in \mathbb{R}^d : \|x\| = 1\}$, because the Gaussian distribution is spherically symmetric.

If p is a spherically symmetric distribution, then it is possible to draw independent 'direction vector' $V \sim \mathcal{U}(S^{d-1})$ and a 'radius' $R \geq 0$ so that $RV \sim p$. The density of the radius q can be found by polar integration.

Proposition 2.16. *Assume that p is a spherically symmetric probability density on $\mathbb{R}^d$, that is,*

$$p(x) = c\hat{p}(\|x\|) \qquad \text{for all} \qquad x \in \mathbb{R}^d,$$

where $c > 0$ is a constant. Suppose q is a probability density on $[0, \infty)$ satisfying

$$q(r) = c' r^{d-1} \hat{p}(r) \qquad \text{for all} \qquad r \in [0, \infty),$$

for some constant $c' > 0$. Then, if $V \sim \mathcal{U}(S^{d-1})$ and $R \sim q$, the random variable $X := RV \sim p$.

Proof. Let $A \subset [0, \infty)$, then by polar integration

$$\int_{\|x\| \in A} p(x) \mathrm{d}x = cC_d \int_{r \in A} r^{d-1} \hat{p}(r) \mathrm{d}r,$$

where C_d is the surface area of the $(d-1)$-sphere. That is, we know that the right density q of R should satisfy

$$q(r) = c' r^{d-1} \hat{p}(r),$$

where $c' = cC_d$. The constant is unique, because q is a probability density. In fact,

$$c' = \left(\int_0^\infty r^{d-1} \hat{p}(r) \mathrm{d}r \right)^{-1}.$$ □

Example 2.17 (Uniform distribution on a d-ball). If $V \sim \mathcal{U}(S^{d-1})$ and $U \sim \mathcal{U}(0, 1)$, then $Z = U^{1/d} V \sim \mathcal{U}(B^d)$, where $B^d := \{x \in \mathbb{R}^d : \|x\| \leq 1\}$.

```
n = 1000; d = 2
X = zeros(d, n)
for k = 1:n
    u = rand(); z = randn(d); v = z/sqrt(sum(z.^2))
    X[:,k] = u^(1/d) * v
end
```

Remark 2.18. Elliptically symmetric densities of the form $p(x) = c\hat{p}(\|L^{-1}(x - m)\|)$ with location $m \in \mathbb{R}^d$ and non-singular shape $LL^T \in \mathbb{R}^{d \times d}$ can be simulated by drawing X from the corresponding spherically symmetric distribution with radial decay $\hat{p}$ as in Proposition 2.16 and then transforming $Y = m + LX$; the argument is identical with Proposition 2.11.

3 Rejection sampling

When it is not possible (or efficient) to do transformations of variables to produce variables that are distributed according to a given distribution, rejection sampling (or the 'accept-reject' method) can make sampling possible (or more efficient).

Example 3.1 (Uniform distribution on a disc). Consider the raindrops in Example 1.4, and assume $(V_k)_{k\geq 1} \overset{\text{i.i.d.}}{\sim} \mathcal{U}([0,1]^2)$. Let $(\hat{V}_k)_{k\geq 1}$ consist of those V_k that fall within the unit disc $D := \{(w_1, w_2) \in \mathbb{R}^2 : w_1^2 + w_2^2 < 1\}$. Then, $(\hat{V}_k) \overset{\text{i.i.d.}}{\sim} \mathcal{U}(D)$, a uniform distribution on the unit disc D.

3.1 Rejection sampling algorithm

To give the general form of rejection sampling, assume that both p and q are p.d.f.s or p.m.f.s on a common space $\mathbb{X}$, and suppose that $M \in [1, \infty)$ is a constant such that

$$\boxed{\text{Assumption: } \frac{p(x)}{q(x)} \leq M \qquad \text{for all } x \in \mathbb{X},} \tag{4}$$

where by convention $0/0 = 0$ and $a/0 = \infty$ for $a > 0$.

Algorithm 3.2 (Rejection sampling)**.** Let $(Y_k)_{k\geq 1} \overset{\text{i.i.d.}}{\sim} q$ which are independent of $(U_k)_{k\geq 1} \overset{\text{i.i.d.}}{\sim} \mathcal{U}(0,1)$. Set $T = 1$ and

(A) If $U_T \leq \dfrac{p(Y_T)}{Mq(Y_T)}$, then output $X = Y_T$.

(R) Otherwise, increment $T = T + 1$ and retry (A).

Remark 3.3. The distribution q in rejection sampling is often called the *proposal distribution* (or the *instrumental distribution*).

Theorem 3.4. *Suppose* (4) *holds and consider* $X = Y_T$ *of Algorithm 3.2.*

(i) The running time $T \sim \text{Geometric}(1/M)$.

(ii) The simulated sample $X \sim p$.

Proof (discrete case). Define

$$h(x) := \begin{cases} \frac{p(x)}{Mq(x)}, & \text{whenever } q(x) > 0, \\ 1, & \text{otherwise.} \end{cases}$$

Denote the 'acceptance indicators' $B_k := \mathbf{1}\left(U_k \leq h(Y_k)\right)$, then B_k are independent Bernoulli random variables, with

$$\begin{aligned} \mathbb{P}(B_k = 1) &= \sum_{y\in\mathbb{X}} \mathbb{P}(B_k = 1, Y_k = y) = \sum_{y\in\mathbb{X}} \mathbb{P}(B_k = 1 \mid Y_k = y)\mathbb{P}(Y_k = y) \\ &= \sum_{y\in\mathbb{X}} \mathbb{P}(U_k \leq h(y))q(y) = \frac{1}{M}\sum_{y\in\mathbb{X}} p(y) = \frac{1}{M}. \end{aligned}$$

That is, $\mathbb{P}(T = t) = \mathbb{P}(B_t = 1)\mathbb{P}(B_1 = 0)\cdots\mathbb{P}(B_{t-1} = 0)$ which proves (i).

Let then $x \in \mathbb{X}$, and calculate for any $t \in \mathbb{N}$,

$$\begin{aligned}\mathbb{P}(X = x \mid T = t) &= \mathbb{P}(Y_t = x \mid B_1 = 0, \ldots, B_{t-1} = 0, B_t = 1) \\ &= \mathbb{P}(Y_t = x \mid B_t = 1) \\ &= \frac{\mathbb{P}(Y_t = x, B_t = 1)}{\mathbb{P}(B_t = 1)}.\end{aligned}$$

Similarly as above, for any $x \in \mathbb{X}$, we get

$$\mathbb{P}(Y_t = x, B_t = 1) = q(x)h(x) = p(x)/M.$$

We conclude that $\mathbb{P}(X = x \mid T = t) = p(x)$. □

Remark 3.5. Note that because $\mathbb{P}(X = x \mid T = t) = \mathbb{P}(X = x)$, the running time T and the sample X produced by Algorithm 3.2 are independent.

Because $T \sim \text{Geometric}(1/M)$, the expected running time (expected number of iterations before stopping) is $\mathbb{E}[T] = M$. Therefore, smaller M leads to a more efficient algorithm.

Remark 3.6. The proof in the continuous case is essentially identical, by considering $A \subset \mathbb{X}$ (or cylindrical sets) and calculating $\mathbb{P}(X \in A \mid T = t)$. In particular, notice that

$$\mathbb{P}(Y_t \in A, B_t = 1) = \int_A q(y)h(y)\mathrm{d}y = \frac{1}{M}\int_A p(y)\mathrm{d}y,$$

from which with $A = \mathbb{X}$ we also deduce that $\mathbb{P}(B_t = 1) = 1/M$.

Remark 3.7 (*). It is not difficult to see that the proof of rejection sampling generalises directly into general state spaces. A similar idea, called *thinning* is used in a point process context, in order to simulate a non-homogeneous Poisson process by discarding some points of a homogeneous Poisson process.

Example 3.8. Suppose we want to use rejection sampling to simulate from $N(0, 1)$ using standard Cauchy proposals. We have

$$\frac{p(x)}{q(x)} = \sqrt{\frac{\pi}{2}}(1 + x^2)\exp\Big(-\frac{x^2}{2}\Big) \le \sqrt{\frac{2\pi}{e}} =: M,$$

because the ratio is maximised with $x = \pm 1$ (derivative zero also at $x = 0$).

```
using Distributions  # Package w/ all 'standard' distributions; install by:
                     # using Pkg; Pkg.add("Distributions")
function cauchy_normal(n)
    M = sqrt(2pi)*exp(-.5)
    x = zeros(n)
    while n>0
        y = rand(Cauchy()); u = rand()
        if M*u < exp(logpdf(Normal(),y) - logpdf(Cauchy(), y))
            x[n] = y;  n = n-1
        end
    end
    x
end
x = cauchy_normal(10_000)
```

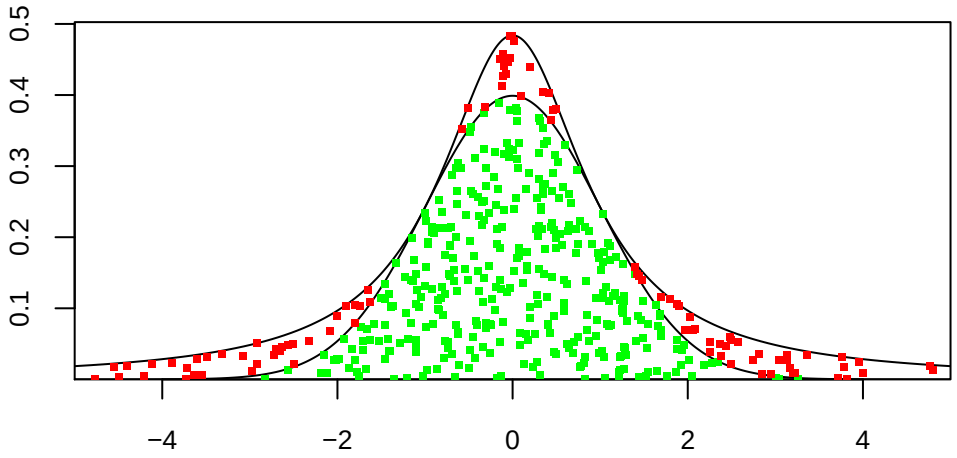


Figure 4: All points $(Y_k, U_k Mq(Y_k))$ simulated in the rejection algorithm of Example 3.8 are *distributed uniformly* in beteeen the x-axis and the function $Mq(x)$ (the upper curve). The points that fall below the curve $p(x)$ are accepted (green), others are rejected (red).

3.2 Unnormalised distributions and rejection sampling

A p.d.f. $p(x)$ on $\mathbb{X}$ (resp. p.m.f. $p(x)$ on $\mathbb{X}$) must satisfy

$$\int_{\mathbb{X}} p(x)\mathrm{d}x = 1 \qquad \left(\text{resp.} \quad \sum_{x\in\mathbb{X}} p(x) = 1\right).$$

We can specify a p.d.f (resp. p.m.f.) by just giving a non-negative function $p_u(x)$, which is propotional to $p(x)$. More specifically, if

$$p(x) \propto p_u(x) \qquad \text{then} \qquad p(x) = \frac{p_u(x)}{Z_p},$$

with the *normalising constant*

$$Z_p := \int_{\mathbb{X}} p_u(x)\mathrm{d}x. \qquad \left(\text{resp.} \quad Z_p = \sum_{x\in\mathbb{X}} p_u(x)\right).$$

The distribution $p(x)$ is fully determined by $p_u(x)$, even though we could not calculate values of $p(x)$. (Of course, we must have $Z_p \in (0, \infty)$.)

Example 3.9. Suppose we know $p(x)$, the density of random variable X, and we are interested in the conditional density of X given $X \geq t$, of the following form:

$$p_t(x) = \frac{p(x)\mathbf{1}\left(x \geq t\right)}{\int_t^\infty p(t)\mathrm{d}t} \propto p(x)\mathbf{1}\left(x \geq t\right).$$

It is clear that we could sample from $(X_k)_{k\geq 1} \overset{\text{i.i.d.}}{\sim} p$, and accept only those for which $X_k \geq t$, which would be samples from p_t.

Example 3.10. In Bayesian inference, we are interested in a conditional distribution (the posterior)

$$p(x) = f_{X|Y}(x \mid y^*) = \frac{f_{Y|X}(y^* \mid x) f_X(x)}{\int_{\mathbb{X}} f_{Y|X}(y^* \mid \hat{x}) f_X(\hat{x}) \mathrm{d}\hat{x}} \propto f_{Y|X}(y^* \mid x) f_X(x),$$

where y^* stands for the observed value of random variable Y and random variable X is the unknown. (Above, $p(x)$ is the conditional density of $X \mid (Y = y^*)$ and $f_{X|Y}$ stands for the conditional density of X given Y.) We can only calculate $p_u(x) = f_{Y|X}(y^* \mid x) f_X(x)$.

We would like an algorithm to simulate $X \sim p$ and use only the unnormalised density $p_u(x)$, without need to calculate $p(x)$. The rejection algorithm can be used in such a case.

Algorithm 3.11 (Rejection sampling with unnormalised distributions)**.** Suppose q and p are p.d.f.s (or p.m.f.s) such that $q \propto q_u$ and $p \propto p_u$, with

$$\boxed{\text{Assumption: } \frac{p_u(x)}{q_u(x)} \leq M \qquad \text{for all } x \in \mathbb{X},} \tag{5}$$

and that $(Y_k)_{k \geq 1} \overset{\text{i.i.d.}}{\sim} q$ independent of $(U_k)_{k \geq 1} \overset{\text{i.i.d.}}{\sim} \mathcal{U}(0,1)$. Set $T = 1$ and

(A) If $U_T \leq \dfrac{p_u(Y_T)}{M q_u(Y_T)}$, then output $X = Y_T$.

(B) Otherwise, increment $T = T + 1$ and retry (A).

Algorithm 3.11 is valid by the proof of Theorem 3.4, with minor adjustments. Namely,

$$\mathbb{P}(Y_t = x, B_t = 1) = \frac{1}{M} q(x) \frac{p_u(x)}{q_u(x)} = \left(\frac{1}{M} \cdot \frac{Z_p}{Z_q} \right) p(x),$$

from which we notice also that $T \sim \text{Geometric}(1/\hat{M})$ where $\hat{M} = M Z_q / Z_p$.

(In fact, $\frac{p_u(y)}{M q_u(y)} = \frac{p(y)}{\hat{M} q(y)}$, so Algorithm 3.11 coincides with Algorithm 3.2 with $\hat{M} = M$.)

Example 3.12. Consider the probability density

$$p(x) \propto p_u(x) := \frac{\sin^2(x)}{x^2} \mathbf{1}\left(x \neq 0\right), \qquad -\infty < x < \infty, \, (x \neq 0)$$

and the standard Cauchy distribution $q(x) \propto q_u(x) = (1 + x^2)^{-1}$, which can be simulated with the inverse c.d.f. method (exercise). We have

$$\frac{p_u(x)}{q_u(x)} = \frac{\sin^2 x (1 + x^2)}{x^2} \leq \min\left\{ \frac{1 + x^2}{x^2}, 1 + x^2 \right\} \leq 2,$$

because $|\sin x| \leq \min\{1, x\}$. (Optimal bound is slightly less than 1.5.)

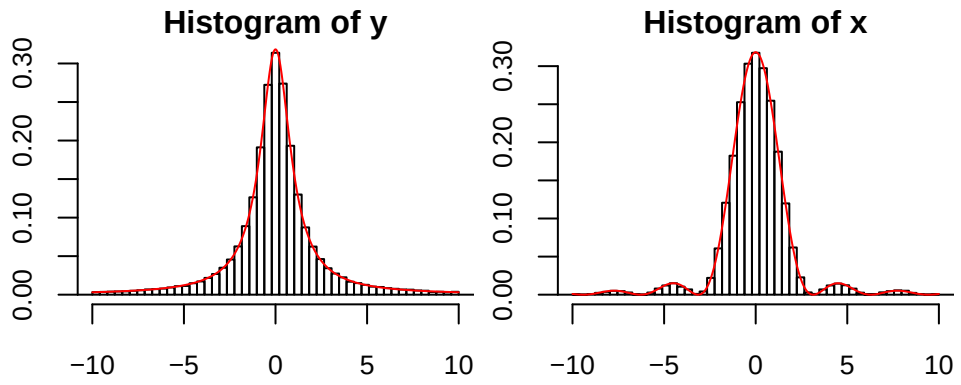


Figure 5: Simulated samples from the standard Cauchy distribution (left) and samples from $p(x) \propto \sin^2(x)/x^2$ (right) with corresponding densities.

```
using Distributions
max_n = 100_000; x = zeros(0) # Empty (zero-length) vector
for k = 1:max_n
    y = rand(Cauchy())
    ratio_pu_qu_M = sin(y)^2*(1+y^2) / (2y^2)
    if rand() <= ratio_pu_qu_M
        push!(x, y)  # Append y to the end of vector x
    end
end
```

4 Importance sampling

All methods up to this point have aimed at simulating i.i.d. random variables $(X_k)_{k\geq 1} \overset{\text{i.i.d.}}{\sim} p$. It is possible to use an auxiliary distribution q for Monte Carlo integration similar to rejection sampling, but without an explicit accept-reject mechanism.

This can be of interest from different reasons, for instance:

- Being less wasteful by 'recycling' samples that would be rejected in rejection sampling.
- Reducing Monte Carlo variance.
- Use when M in (4) or (5) is unknown, or even when no such finite M exists.

4.1 Unbiased importance sampling

Definition 4.1 (Importance sampling)**.** Suppose p and q are two p.d.f.s or p.m.f.s on $\mathbb{X}$ and $f : \mathbb{X} \to \mathbb{R}$.

$$\boxed{\text{Assumption: } q(x) = 0 \implies p(x)f(x) = 0.} \tag{6}$$

Define

$$w(x) := \begin{cases} \frac{p(x)}{q(x)}, & \text{if } q(x) > 0, \\ 0, & \text{otherwise.} \end{cases}$$

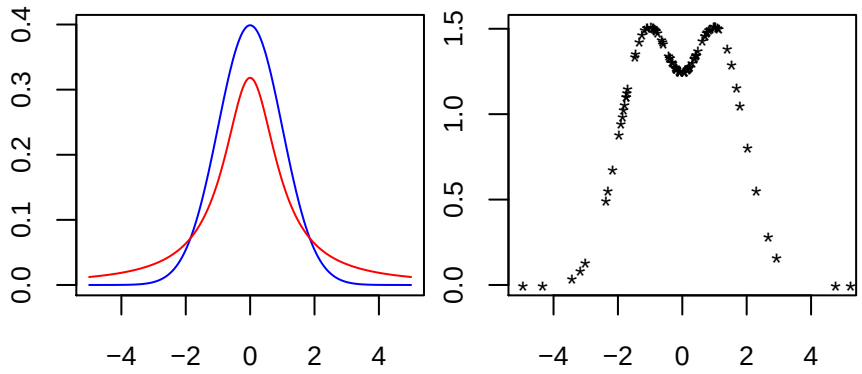


Figure 6: Importance sampling with p standard Normal (blue) and q Cauchy (red), as in Example 3.8). The importance weights $w(Y_k)$ are shown on the right.

Then, if $Y_1, Y_2, \ldots, Y_n \overset{\text{i.i.d.}}{\sim} q$, the estimator

$$I_{p,q}^{(n)}(f) := \frac{1}{n}\sum_{k=1}^{n} f(Y_k)w(Y_k) \tag{7}$$

is the (unbiased) importance sampling (IS) approximation of $\mathbb{E}_p[f(X)]$.

Remark 4.2. The distribution q is called the *proposal distribution* (sometimes also *importance* or *instrumental*). The term $w(Y_k)$ is called the *(importance) weight* related to the sample Y_k.

Theorem 4.3. *Assuming* (6) *holds, then the IS estimator is*

(a) Unbiased: $\mathbb{E}[I_{p,q}^{(n)}(f)] = \mathbb{E}_p[f(X)]$, *for all* $n \in \mathbb{N}$

(b) Consistent: $I_{p,q}^{(n)}(f) \xrightarrow{n\to\infty} \mathbb{E}_p[f(X)]$ *(almost surely).*

Proof. Because the random variables $Z_k := f(Y_k)w(Y_k)$ are i.i.d., it is sufficient for (a) to check that $\mathbb{E}[Z_1] = \mathbb{E}_p[f(X)]$. In the discrete case,

$$\mathbb{E}[Z_1] = \sum_{y\in\mathbb{X}\,:\,q(y)>0} f(y)\frac{p(y)}{q(y)}q(y) = \sum_{y\in\mathbb{X}} f(y)p(y)\mathrm{d}y = \mathbb{E}_p[f(X)],$$

and similarly in the continuous case, changing the sum to an integral. The almost sure convergence (b) follows from the strong law of large numbers. □

Remark 4.4 (*). In terms of general probability, importance sampling is a *change of measure*, and the function w is the related *Radon-Nikodym derivative*.

Example 4.5 (Gamma distribution). Example 2.14 showed how to simulate $Y \sim \Gamma(a, b)$ for $a \in \mathbb{N}_+$ and $b > 0$ by summing exponentials.

Suppose we have simulated $Y_1, \ldots, Y_n \overset{\text{i.i.d.}}{\sim} \Gamma(a, b)$, but want to estimate the expectation of $f(X)$ where $X \sim \Gamma(\alpha, \beta)$, with some other parameters $\alpha, \beta > 0$.

Recall that the density of $\Gamma(\alpha, \beta)$ is

$$p(x) = \frac{\beta^\alpha}{\Gamma(\alpha)}x^{\alpha-1}\exp(-\beta x)\mathbf{1}\,(x > 0)$$

so the importance weights are given as

$$w(y) = \frac{p(y)}{q(y)} = \frac{\Gamma(a)\beta^\alpha}{\Gamma(\alpha)b^a} y^{\alpha-a} \exp\big(-(\beta-b)y\big), \qquad \text{for } y > 0.$$

The importance sampling estimator is (NB: $\mathbb{P}(q(Y_i) = 0) = 0$!)

$$\begin{aligned} I_{p,q}^{(n)}(f) &= \frac{1}{n}\sum_{i=1}^n f(Y_i)w(Y_i) \\ &= \frac{\Gamma(a)\beta^\alpha}{\Gamma(\alpha)b^a} \cdot \frac{1}{n}\sum_{i=1}^n f(Y_i) Y_i^{\alpha-a} \exp\big(-(\beta-b)Y_i\big). \end{aligned}$$

We know that this is unbiased and (strongly) consistent estimator of $\mathbb{E}_p[f(X)]$.

Remark 4.6. In fact, we can 'recycle' the samples the $Y_1, \ldots, Y_n \overset{\text{i.i.d.}}{\sim} q$ in Example 4.5 to obtain estimates of $\mathbb{E}_{p_{\alpha,\beta}}[f(X)]$ with $p_{\alpha,\beta}$ corresponding to $\Gamma(\alpha, \beta)$, for a *range* of values α and β...

Theorem 4.3 showed that IS is consistent with minimal conditions. How about the variance of IS?

Proposition 4.7. *Suppose that* (6) *holds. Then, the variance of the IS estimator can be given as*

$$\mathrm{Var}\big(I_{p,q}^{(n)}(f)\big) = \frac{\sigma_{p,q}^2}{n} \qquad \textit{where} \qquad \sigma_{p,q}^2 := \mathbb{E}_p[f^2(X)w(X)] - \mathbb{E}_p[f(X)]^2.$$

Note that this permits the case $\sigma_{p,q}^2 = \infty \implies \mathrm{Var}\big(I_{p,q}^{(n)}(f)\big) = \infty \; \forall\, n \in \mathbb{N}$.

Proof. Denote $Z_k := f(Y_k)w(Y_k)$, then in the discrete case

$$\mathbb{E}[Z_1^2] = \sum_{y \in \mathbb{X}\,:\, q(y)>0} f^2(y)\frac{p^2(y)}{q^2(y)}q(y) = \sum_{y\in\mathbb{X}} f^2(y)w(y)p(y)\mathrm{d}y = \mathbb{E}_p[f^2(X)w(X)].$$

Now, $\sigma_{p,q}^2 = \mathrm{Var}(Z_1) = \mathbb{E}Z_1^2 - (\mathbb{E}Z_1)^2$ and $\mathbb{E}Z_1 = \mathbb{E}_p[f(X)]$, and as (Z_k) are i.i.d., $\mathrm{Var}(I_{p,q}^{(n)(f)}) = \sigma_{p,q}^2/n$. The continuous case follows similarly. □

Because $I_{p,q}^{(n)}(f)$ is a sum of i.i.d. random variables, the proof of Proposition 4.7 implies the following:

Corollary 4.8. *Suppose* (6) *holds and*

$$\mathbb{E}_p[f^2(X)w(X)] < \infty. \tag{8}$$

Then, $\sqrt{n}[I_{p,q}^{(n)}(f) - \mathbb{E}_p[f(X)]] \xrightarrow{n\to\infty} N(0, \sigma_{p,q}^2)$ *in distribution.*

Remark 4.9. Because IS is just usual Monte Carlo approximating $\mathbb{E}_q[g(X)]$ with $g(x) = f(x)w(x)$, Proposition 1.13 holds, and gives confidence intervals also for the IS estimator.

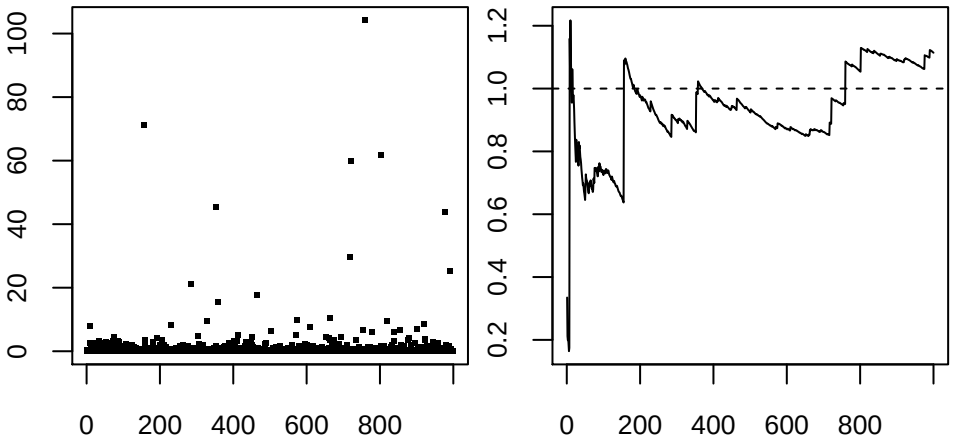


Figure 7: Example 4.10 with $a = 2$, $b = 2$ and $\beta = 2.5$, $\alpha = 0.5$ (NB $\alpha < a$) and $f(x) \equiv 1$. Values of the weights $w(Y_n)$ (left) and the sequence of estimates $I_{p,q}^{(n)}(f)$ (right) for $n = 1, 2, \ldots, 1000$.

Example 4.10 (Gamma distribution (cont.)). Let us consider the variance of the IS estimator for the Gamma distributions in Example 4.5. We may write

$$w(x)f^2(x) = \frac{\Gamma(a)\beta^\alpha}{\Gamma(\alpha)b^a} x^{\alpha-a} \exp(-(\beta - b)x) f^2(x),$$

so

$$\mathbb{E}_p[w(X)f^2(X)] = c_{a,b,\alpha,\beta} \mathbb{E}_p\big[X^{\alpha-a} \exp(-(\beta - b)X) f^2(X)\big].$$

If $\alpha \geq a$ and $\beta > b$, then

$$\sup_{x>0} \big[x^{\alpha-a} \exp(-(\beta - b)x)\big] < \infty.$$

In this case $\mathbb{E}_p[w(X)f^2(X)] \leq c\mathbb{E}_p[f^2(X)]$, so if also $\mathrm{Var}_p(f(X)) < \infty \iff \mathbb{E}_p[f^2(X)] < \infty$, then we have $\mathbb{E}_p[w(X)f^2(X)] < \infty$ and the importance sampling estimator is guaranteed to have finite variance.

Figure 7 shows an example simulation where $\mathrm{Var}(I_{p,q}^{(n)}(f)) = \infty$. Exercise: What would happen if we used $f(x) = x$ instead?

We formalise the sufficient condition found in the Gamma example above.

Proposition 4.11. *Suppose* (6) *holds and*

$$M := \sup_x w(x) = \sup_x \frac{p(x)}{q(x)} < \infty, \tag{9}$$

where the supremum is taken over all $x \in \mathbb{X}$ such that $p(x)f(x) > 0$. Then, if $\mathrm{Var}_p(f(X)) < \infty$, the variance of the IS estimator is finite, and can be upper bounded by

$$\begin{aligned}\sigma_{p,q}^2 &\leq M\mathbb{E}_p[f^2(X)] - \mathbb{E}_p[f(X)]^2 \\ &= M\mathrm{Var}_p(f(X)) + (M-1)\mathbb{E}_p[f(X)]^2.\end{aligned}$$

Remark 4.12. If $\mathbb{E}_p[f(X)]^2 \ll \mathbb{E}_p[f^2(X)]$, Proposition 4.11 indicates that the IS estimator is (roughly) at most M times worse than the classical Monte Carlo estimate. How does this result relate with using rejection sampling instead of IS?

> Rule of thumb: Try to make sure that (9) holds (unless you have a specific f in mind).

What is the best possible proposal density q for a specific f?

Proposition 4.13. *Suppose that $f : \mathbb{X} \to \mathbb{R}$ satisfies $\mathbb{E}_p[|f(X)|] > 0$. Then, the proposal distribution*

$$q_*(x) := \frac{p(x)|f(x)|}{\mathbb{E}_p[|f(X)|]} \propto p(x)|f(x)|$$

admits the minimum variance among all distributions q satisfying (6).

Proof. In the discrete case, we have with $w_*(x) = p(x)/q_*(x)$,

$$\mathbb{E}_p[f^2(X)w_*(X)] = \sum_{x\in\mathbb{X}\,:\,q_*(x)>0} f^2(x)\frac{p^2(x)}{q_*(x)} = \big(\mathbb{E}_p[|f(X)|]\big)^2$$

On the other hand, for any q satisfying (6),

$$\big(\mathbb{E}_p[|f(X)|]\big)^2 = \Big(\mathbb{E}_q[|f(X)|w(X)]\Big)^2 \le \mathbb{E}_q[f^2(X)w^2(X)] = \mathbb{E}_p[f^2(X)w(X)],$$

by Jensen's inequality. This implies $\sigma^2_{p,q_*} \le \sigma^2_{p,q}$ by Proposition 4.7. □

Remark 4.14. The result of Proposition 4.13 is, of course, mostly theoretical, but leads to:

> Rule of thumb: Try to find q that is approximately proportional to $p(x)|f(x)|$.

In particular, if f is zero (or has very small absolute values) in some regions of the space, we avoid putting any (or put less) mass of q to such regions.

Remark 4.15. Note in particular that IS can have, in fact, a (significantly) smaller variance than the classical Monte Carlo estimate. We restate the main reasons to use IS rather than classical Monte Carlo:

- Use IS when we cannot sample (efficiently) from p.
- Use IS to reduce variance over the classical Monte Carlo estimator.
- Rejection sampling is not applicable (because we do not know $M < \infty$, or $M = \infty$)

4.2 Application: Rare event estimation

One important class of applications of IS as variance reduction is problems in which we estimate the probability of a rare event. In such scenarios, we may be able to sample from p directly but this leads to high variance.

For example, suppose $X \sim p$ and we want to estimate

$$\mathbb{P}(X \geq x_0) = \mathbb{E}_p[\mathbf{1}\,(X \geq x_0)]$$

with x_0 in the extreme upper tail of $p(x)$. If $X_1, \ldots, X_n \overset{\text{i.i.d.}}{\sim} p$, we may not get any samples $X_i \geq x_0$ and the usual Monte Carlo estimate

$$I_p^{(n)}(f) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\,(X_i \geq x_0)$$

is zero with high probability. We can take an proposal density q that puts more probability at large Y, and then reweight to get expectations in X. By using IS, we can reduce the variance significantly.

Example 4.16. Say $p(x)$ is the standard normal density and we want to estimate $\theta = \mathbb{P}(X \geq x_0)$ for some $x_0 \geq 3$.

Take q as the shifted exponential,

$$q(y) := r \exp\big(-r(y - x_0)\big)\mathbf{1}\,(y \geq x_0)\,.$$

Let us determine r so that q approximates the optimal distribution (the conditional tail of p) locally: $(\log p)' = (\log q)'$ at x_0, that is,

$$r = g'(x_0), \qquad g(x) = -\log p(x) = \frac{x^2}{2} \qquad \Longrightarrow \qquad r = x_0.$$

The weights are, for $y \geq x_0$,

$$\begin{aligned} w(y) &= \frac{p(y)}{q(y)} \\ &= \frac{1}{r\sqrt{2\pi}} \exp\left(-\frac{y^2}{2} + r(y - x_0)\right) \end{aligned}$$

and the IS estimator of θ is $\frac{1}{n}\sum_{i=1}^{n} w(Y_i)\mathbf{1}\,(Y_i \geq x_0)$; See Figure 8.

4.3 Self-normalised importance sampling

The rejection sampling algorithm is straightforward to apply in case of unknown normalising constants, that is, when only the unnormalised densities $p_u(x) \propto p(x)$ and $q_u(x) \propto q(x)$ are available.

In importance sampling, this means that we can access the *unnormalised* importance weights

$$w_u(x) := \frac{p_u(x)}{q_u(x)} = \frac{Z_p}{Z_q} w(x), \qquad q(x) > 0,$$

and $w_u(x) := 0$ when $q(x) = 0$. In order to apply (unbiased) importance sampling, we would need w. We can get around by *simultaneously* estimating the ratio Z_p/Z_q, with a cost of introducing a bias (which is asymptotically vanishing).

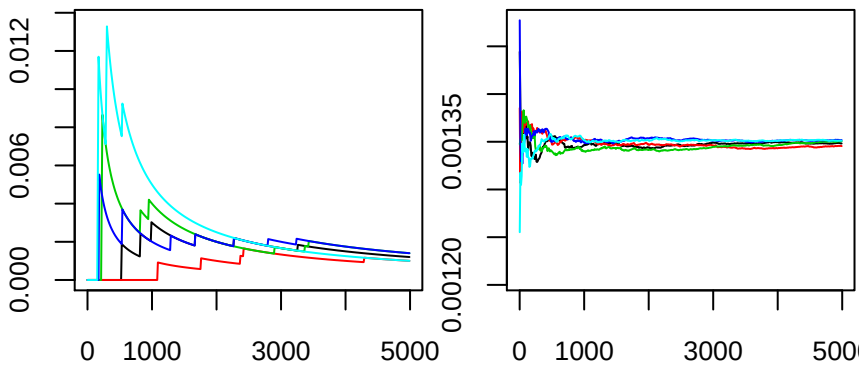


Figure 8: Five trajectories of classical Monte Carlo (left) and IS of Example 4.16 (right) with $x_0 = 3$. Number of samples in x-axis and value of estimate in y-axis.

Definition 4.17 (Self-normalised importance sampling)**.** Suppose p and q are p.d.f.s or p.m.f.s, such that

$$\boxed{\text{Assumption: } q(x) = 0 \implies p(x) = 0.} \tag{10}$$

Then, if $Y_1, Y_2, \ldots \overset{\text{i.i.d.}}{\sim} q$,

$$\hat{I}^{(n)}_{p,q}(f) := \sum_{k=1}^{n} f(Y_k) W^{(n)}_k, \tag{11}$$

$$\text{where} \quad W^{(n)}_k := \begin{cases} \frac{w_u(Y_k)}{\sum_{j=1}^n w_u(Y_j)}, & \text{if } w_u(Y_j) > 0 \text{ for some } 1 \le j \le n \\ \mathbf{1}(k = 1), & \text{otherwise} \end{cases}$$

is the *self-normalised* (or *rescaled*) IS approximation of $\mathbb{E}_p[f(X)]$.

Remark 4.18. Note that

(a) $\beta = \mathbb{P}_q(w_u(Y_j) > 0) = \mathbb{P}_q(p(Y_j) > 0) > 0$, and therefore

$$\mathbb{P}_q\big(w_u(Y_j) > 0 \text{ for some } 1 \le j \le n\big) = 1 - (1-\beta)^n \xrightarrow{n\to\infty} 1.$$

(b) We always have $\sum_{k=1}^n W^{(n)}_k = 1$.

The drawback of the self-normalised IS is that the estimator $\hat{I}^{(n)}_{p,q}(f)$ is generally biased for finite n. However, the estimator is (strongly) consistent.

Theorem 4.19. *Suppose* (10) *holds. Then,* $\hat{I}^{(n)}_{p,q}(f) \xrightarrow{n\to\infty} \mathbb{E}_p[f(X)]$ *(almost surely).*

Proof. Because $w_u(Y_j) > 0$ for some $1 \le j \le n$ eventually (almost surely; cf. Remark 4.18), we may consider only such n.

$$\hat{I}^{(n)}_{p,q}(f) = \frac{\sum_{k=1}^n f(Y_k) w_u(Y_k)}{\sum_{k=1}^n w_u(Y_k)} = \frac{\frac{1}{n}\sum_{k=1}^n f(Y_k) w(Y_k)}{\frac{1}{n}\sum_{k=1}^n w(Y_k)} = \frac{I^{(n)}_{p,q}(f)}{I^{(n)}_{p,q}(1)}.$$

Theorem 4.3 (b) implies that $I^{(n)}_{p,q}(f) \xrightarrow{n\to\infty} \mathbb{E}_p[f(X)]$ almost surely and $I^{(n)}_{p,q}(1) \xrightarrow{n\to\infty} \mathbb{E}_p[1] = 1$ almost surely. □

Remark 4.20. In the proof of Theorem 4.19, we need the condition $q(x) = 0 \implies p(x) = 0$ in order to ensure $I_{p,q}^{(n)}(1) \to 1$. This is more stringent than with unbiased IS, where we only need $q(x) = 0 \implies p(x)f(x) = 0$ which ensures $I_{p,q}^{(n)}(f) \to \mathbb{E}_p[f(X)]$.

Remark 4.21. Note that

$$\mathbb{E}_q[w_u(Y)] = \frac{Z_p}{Z_q}\mathbb{E}_q[w(Y)] = \frac{Z_p}{Z_q},$$

so the mean of unnormalised SNIS weights is unbiased and (strongly) consistent estimator of the ratio of normalising constants,

$$\frac{1}{n}\sum_{k=1}^{n} w_u(Y_k) \xrightarrow{n\to\infty} \frac{Z_p}{Z_q} \qquad \text{(almost surely)}.$$

This is important in certain applications.

Example 4.22. We saw in Example 4.5 that if $Y_i \sim \Gamma(a, b)$ and

$$w(y) = \frac{\Gamma(a)\beta^\alpha}{\Gamma(\alpha)b^a} y^{\alpha-a}\exp(-(\beta - b)y)$$

then

$$I_{p,q}^{(n)}(f) = \frac{1}{n}\sum_{i=1}^{n} f(Y_i)w(Y_i)$$

is unbiased and consistent estimator of $\mathbb{E}_p[f(X)]$ with $p = \Gamma(\alpha, \beta)$.

To avoid calculating $\Gamma(a)/\Gamma(\alpha)$, we can use

$$w_u(y) = y^{\alpha-a}\exp(-(\beta - b)y)$$

and then the self-normalised IS estimator

$$\hat{I}_{p,q}^{(n)}(f) := \frac{\sum_{i=1}^n f(Y_i)w_u(Y_i)}{\sum_{i=1}^n w_u(Y_i)}$$

is a consistent estimator of $\mathbb{E}_p[f(X)]$.

```
function snis_gamma(n, alpha, beta, f)
   y = -log.(rand(n))                          # y ~ Exp(1) = Gamma(1,1)
   w_u = y.^(alpha-1) .* exp.(-(beta-1)*y) # Unnormalised w
   w = w_u/sum(w_u)                            # Normalised w
   sum(f.(y) .* w)                             # SNIS estimate
end
# Use the function f(x)=x to estimate mean:
snis_gamma(1000, 2, 4, x -> x)
```

The self-normalised IS satisfies a CLT with same variance as the unbiased IS for zero mean functions, in which case they are asymptotically equally efficient. A consistent confidence interval can also be easily constructed.

Theorem 4.23. *Suppose* (10) *holds and* $\bar{\sigma}^2_{p,q} := \mathbb{E}_p[w(X)\bar{f}^2(X)] < \infty$, *where* $\bar{f}(x) = f(x) - \mathbb{E}_p[f(X)]$.

(i) $\sqrt{n}\big(\hat{I}^{(n)}_{p,q}(f) - \mathbb{E}_p[f(X)]\big) \xrightarrow{n\to\infty} N(0, \bar{\sigma}^2_{p,q})$ *in distribution.*

(ii) If also $\mathbb{E}_p[w(X)] < \infty$, *then the following hold:*

- $nv^{(n)}_{p,q} \xrightarrow{n\to\infty} \bar{\sigma}^2_{p,q}$ *(a.s.), where* $v^{(n)}_{p,q} := \sum_{k=1}^n (W^{(n)}_k)^2\big[f(Y_k) - \hat{I}^{(n)}_{p,q}(f)\big]^2$, *and*
- $\mathbb{P}\Big(\mathbb{E}_p[f(X)] \in \Big[\hat{I}^{(n)}_{p,q}(f) \pm \alpha\sqrt{v^{(n)}_{p,q}}\Big]\Big) \to 1 - 2\Phi(-\alpha)$ *for any* $\alpha \in (0, \infty)$.

Proof. (i) Because $\sum_{k=1}^n W^{(n)}_k = 1$, $\hat{I}^{(n)}_{p,q}(f) - \mathbb{E}_p[f(X)] = \hat{I}^{(n)}_{p,q}(\bar{f})$. Now, as in the proof of Theorem 4.19, $\sqrt{n}\hat{I}^{(n)}_{p,q}(\bar{f}) = \sqrt{n}I^{(n)}_{p,q}(\bar{f})/I^{(n)}_{p,q}(1)$. Corollary 4.8 implies that the numerator converges in distribution to $N(0, \bar{\sigma}^2_{p,q})$ and the denominator converges to 1 almost surely. Slutsky's theorem (Lemma 1.14) concludes the proof. The first part of (ii), that is, $nv^{(n)}_{p,q} \to \bar{\sigma}^2_{p,q}$ is an exercise, and the second claim follows from (i), as in the proof of Proposition 1.13 (iii). □

Remark 4.24 (*). The quantity $n_{\text{eff}} = \big(\sum_{k=1}^n (W^{(n)}_k)^2\big)^{-1} \in [1, n]$ is widely known as the *effective sample size* of (self-normalised) IS.

This may be (loosely) justified when the function is of the form $f(x) := c\mathbf{1}(x \in A)$ with $c > 0$ and A such that $\mathbb{E}_p[\mathbf{1}(X \in A)] = 1/2$. In this case, $\bar{f}(x) \equiv \frac{c}{2}$, and standard Monte Carlo estimator $I^{(n)}_p(f)$ would have variance $\mathrm{Var}_p(f(X))/n = (c/2)^2/n$, but the corresponding limiting CLT variance of the SNIS estimator is $\mathbb{E}_p[w(X)\bar{f}^2(X)]/n$. It is not hard to see (cf. the proof of Theorem 4.23 (ii)) that then

$$\frac{n}{n_{\text{eff}}} \xrightarrow{n\to\infty} \mathbb{E}_p[w(X)],$$

so $\mathbb{E}_p[w(X)]/n \approx \mathrm{Var}_p(f(X))/n_{\text{eff}}$ for large n. Therefore, the self-normalised IS with n samples may be (loosely) thought of as having n_{eff} 'effective independent samples'.

Remark 4.25 (*). It is sometimes useful to consider the SNIS as an empirical approximation of the distribution p. That is,

$$\hat{\mu}^{(n)}_{p,q}(A) := \sum_{k=1}^n W^{(n)}_k \mathbf{1}(Y_k \in A) \approx \mathbb{P}(X \in A), \qquad A \subset \mathbb{X},$$

where $X \sim p$. The approximation is consistent assuming (10), in the following sense:

$$\hat{\mu}^{(n)}_{p,q}(A) \xrightarrow{n\to\infty} \mathbb{P}(X \in A) \qquad \text{almost surely,}$$

for any (measurable) $A \subset \mathbb{X}$.

With unbiased IS, we have

$$\mu^{(n)}_{p,q}(A) := \frac{1}{n}\sum_{k=1}^n w(Y_k)\mathbf{1}(Y_k \in A).$$

Given (10) this is consistent and also unbiased $\mathbb{E}[\mu^{(n)}_{p,q}(A)] = \mathbb{P}(X \in A)$, but unlike self-normalised IS and plain MC, $\mu^{(n)}_{p,q}$ is not a probability distribution, because $\mu^{(n)}_{p,q}(\mathbb{X}) \neq 1$ in general.

5 Variance reduction techniques

Small variance is vital with Monte Carlo methods, because m-fold reduction of variance means that we may use m-fold less samples to get an estimator with same variance. We saw above that importance sampling can be used to reduce variance of the Monte Carlo estimate. There are other useful techniques which we consider next.

5.1 Rao-Blackwellisation

Recall the *law of total variance*.

Proposition 5.1. *If* $\mathrm{Var}(Z) < \infty$, *then*

$$\mathrm{Var}(Z) = \mathbb{E}[\mathrm{Var}(Z \mid Y)] + \mathrm{Var}(\mathbb{E}[Z \mid Y]),$$

where $\mathrm{Var}(Z \mid Y) = \mathbb{E}[Z^2 \mid Y] - (\mathbb{E}[Z \mid Y])^2 \geq 0$.

Corollary 5.2. *If* $\mathrm{Var}(Z) < \infty$, *then*

$$\mathrm{Var}(\mathbb{E}[Z \mid Y]) = \mathrm{Var}(Z) - \mathbb{E}[\mathrm{Var}(Z \mid Y)] \leq \mathrm{Var}(Z).$$

That is, conditioning can only decrease variance.

Example 5.3 (Rao-Blackwellisation in $\mathbb{R}^2$). Suppose that p is a p.d.f. in $\mathbb{R}^2$, and we would like to compute

$$\mathbb{E}_p[f(X,Y)] = \iint f(x,y)p(x,y)\mathrm{d}x\mathrm{d}y.$$

Simple Monte Carlo would be to simulate $(X_k, Y_k) \overset{\text{i.i.d.}}{\sim} p$ and then compute the average $I_p^{(n)}(f) = n^{-1}\sum_{k=1}^n f(X_k, Y_k)$.

However, if the conditional law $p_{X|Y}(x \mid y)$ is available, and we can calculate the conditional expectation

$$h(y) := \mathbb{E}_p[f(X,y) \mid Y = y],$$

(that is, with $Z = f(X,Y)$, we have $\mathbb{E}[Z \mid Y] = h(Y)$), we may use instead

$$I_{p,\mathrm{RB}}^{(n)}(f) := \frac{1}{n}\sum_{k=1}^n h(Y_k), \tag{12}$$

which approximates the desired quantity $\mathbb{E}_p[f(X,Y)]$ and has smaller variance than $I_p^{(n)}(f)$ (and the improvement can be significant).

Remark 5.4. In Example 5.3, we need only the samples $(Y_k)_{k\geq 1}$ which are distributed according to the marginal disrtribution $p_Y(y) := \int p(x,y)\mathrm{d}x$. We have a choice to simulate either $(X_k, Y_k)_{k\geq 1} \overset{\text{i.i.d.}}{\sim} p$ and throwing away X_k, or simulating directly from the marginal distribution $(Y_k) \overset{\text{i.i.d.}}{\sim} p_Y$, whichever is easier.

Remark 5.5. Rao-Blackwellisation applies similarly also with importance sampling, and with other Monte Carlo methods, such as Markov chain Monte Carlo introduced later.

Remark 5.6 (*). The term *Rao-Blackwellisation* is used, because the method is often associated with sufficient statistics and the Rao-Blackwell theorem. *Marginalisation* or *conditioning* might be more appropriate, but Rao-Blackwellisation is widely used for historical reasons.

Remark 5.7 (*). Sometimes, it may be useful to employ some (biased) approximations $\hat{h}(y) \approx \mathbb{E}[f(X) \mid Y = y]$ in place of the true conditional expectation. Theoretical guarantees for such 'approximate Rao-Blackwellisation' are usually not available, but empirically this type of schemes may be useful.

5.2 Stratification

Example 5.8. Suppose we are interested to estimate $\mathbb{E}_p[f(X)]$ with

$$p(x) = \frac{1}{2}p_1(x) + \frac{1}{2}p_2(x),$$

where p_1 and p_2 are distributions on $\mathbb{X}$.

(a) We know how to sample $X_1, \dots, X_n \sim p$ using $Z_k^{(i)} \overset{\text{i.i.d.}}{\sim} p_i$ and $U_k \overset{\text{i.i.d.}}{\sim} \mathcal{U}(0,1)$:

$$\begin{aligned} I_p^{(n)}(f) &= \frac{1}{n}\left[\sum_{k=1}^{n} \mathbf{1}\left(U_k \le \frac{1}{2}\right) f(Z_k^{(1)}) + \mathbf{1}\left(U_k > \frac{1}{2}\right) f(Z_k^{(2)})\right] \\ &\overset{d}{=} \frac{1}{n}\sum_{k=1}^{N_1} f(\tilde{Z}_k^{(1)}) + \frac{1}{n}\sum_{k=1}^{N_2} f(\tilde{Z}_k^{(2)}), \end{aligned}$$

where $(\tilde{Z}_k^{(i)}) \overset{\text{i.i.d.}}{\sim} p_i$, $N_1 = \sum_{k=1}^{n} \mathbf{1}\left(U_k \le \frac{1}{2}\right) \sim \text{Binom}(n, 1/2)$ and $N_2 = n - N_1 \sim \text{Binom}(n, 1/2)$ (note that N_1 and N_2 are not independent though!).

(b) Notice that $\mathbb{E}_p[f(X)] = (1/2)\mathbb{E}_{p_1}[f(X)] + (1/2)\mathbb{E}_{p_2}[f(X)]$, so we may use

$$\begin{aligned} I_p^{(n/2,n/2)}(f) &= \frac{1}{2}I_{p_1}^{(n/2)}(f) + \frac{1}{2}I_{p_2}^{(n/2)}(f) \\ &= \frac{1}{n}\sum_{k=1}^{n/2} f(Z_k^{(1)}) + \frac{1}{n}\sum_{k=1}^{n/2} f(Z_k^{(2)}). \end{aligned}$$

Which estimator should we use? The estimator $I_p^{(n/2,n/2)}(f)$, because it turns out that $\text{Var}\big(I_p^{(n/2,n/2)}(f)\big) \le \text{Var}(I_p^{(n)}(f))$. This is an example of *stratification* (with proportional allocation).

Theorem 5.9. *Suppose the distribution p is of the following mixture form:*

$$p(x) = \sum_{i=1}^{m} w_i p_i(x),$$

where $w_i > 0$ and $\sum_i w_i = 1$ and $p_1, \dots, p_m$ are distributions.

Let $\ell_1, \dots, \ell_m \in \mathbb{N}$ with $\sum_i \ell_i = n$, and define the stratified estimator

$$I_{\text{p}}^{(\ell_1,\dots,\ell_m)}(f) := \sum_{i=1}^{m} w_i \left(\frac{1}{\ell_i}\sum_{j=1}^{\ell_i} f(X_j^{(i)})\right),$$

where $(X_j^{(i)})_{i,j}$ *are all independent and* $(X_j^{(i)}) \overset{i.i.d.}{\sim} p_i$*. The estimator satisfies*

(i) Unbiasedness: $\mathbb{E}[I_{\mathrm{p}}^{(\ell_1,\dots,\ell_m)}(f)] = \mathbb{E}_p[f(X)]$.

(ii) If $\boxed{\ell_i = w_i n}$ *(proportional allocation), then*

$$\mathrm{Var}\big(I_{\mathrm{p}}^{(\ell_1,\dots,\ell_m)}(f)\big) \le \mathrm{Var}\big(I_p^{(n)}(f)\big).$$

Proof. Unbiasedness (i) is direct, and

$$\begin{aligned}\mathrm{Var}\big(I_{\mathrm{p}}^{(\ell_1,\dots,\ell_m)}(f)\big) &= \sum_{i=1}^m w_i^2 \mathrm{Var}\left(\frac{1}{\ell_i}\sum_{j=1}^{\ell_i} f(X_j^{(i)})\right)\\ &= \sum_{i=1}^m \frac{w_i^2}{\ell_i}\mathrm{Var}_{p_i}\big(f(X)\big)\\ &= \frac{1}{n}\sum_{i=1}^m w_i \mathrm{Var}_{p_i}\big(f(X)\big),\end{aligned}$$

because $\ell_i = w_i n$. Consider then $X = \sum_{i=1}^m \mathbf{1}\left(s_{i-1} \le U < s_i\right) X^{(i)}$, where $U \sim \mathcal{U}(0,1)$, $s_0 = 0$, $s_i = \sum_{k=1}^i w_i$ and $X^{(i)} \sim p_i$, then $X \sim p$ (exercise!) and we notice that

$$\sum_{i=1}^m w_i \mathrm{Var}_{p_i}\big(f(X)\big) = \mathbb{E}[\mathrm{Var}(f(X) \mid U)] \le \mathrm{Var}_p\big(f(X)\big). \qquad \square$$

Example 5.10 (Stratification with inverse c.d.f.). Suppose F^{-1} is the (generalised) inverse c.d.f. corresponding to a distribution p, and we try to approximate $\mathbb{E}_p[f(X)]$. We may use the following stratified estimator

$$I_{p,\mathrm{strat}}^{(n)}(f) := \frac{1}{n}\sum_{k=1}^n f(X_k), \qquad X_k := F^{-1}(\tilde{U}_k), \qquad \tilde{U}_k := \frac{k-1+U_k}{n},$$

where $(U_k) \overset{\text{i.i.d.}}{\sim} \mathcal{U}(0,1)$.

This is, in fact, proportionally allocated stratification, which follows by writing $\mathbb{E}_p[f(X)] = \mathbb{E}_u[f(F^{-1}(U)]$, where the uniform density can be written as

$$u(t) := \mathbf{1}\left(0 < t \le 1\right) = \sum_{k=1}^n w_k \tilde{u}_k(t),$$

where $w_k = 1/n$ and $\tilde{u}_k(t) = n\mathbf{1}\left(\frac{k-1}{n} < t \le \frac{k}{n}\right)$ are the densities of $\tilde{U}_k$.

Remark 5.11 (*). Stratification with proportional allocation is guaranteed to provide at least as good estimates as without stratification, but optimal allocation strategy would be $\ell_i \propto w_i\sqrt{\mathrm{Var}_{p_i}(f(X))}$. Because this depends on f and we may be interested in several f, and because $\mathrm{Var}_{p_i}(f(X))$ is usually not known, proportional allocation is often a safe choice.

5.3 Introducing dependence: antithetic variables

In some cases, it is possible to use the dependence of random variables to help decrease the variance. First such technique is so-called 'antithetic' variables.

Definition 5.12 (Antithetic variables)**.** Suppose $\hat{p}(x, y)$ is a joint distribution with p as its marginals[7]. In the discrete case, this means that for all $x, y \in \mathbb{X}$,

$$p(x) = \sum_{z \in \mathbb{X}} \hat{p}(x, z) \quad \text{and} \quad p(y) = \sum_{z \in \mathbb{X}} \hat{p}(z, y).$$

Suppose that $(X_n, Y_n)_{n \geq 1} \overset{\text{i.i.d.}}{\sim} \hat{p}$, then clearly $(X_n)_{n \geq 1} \overset{\text{i.i.d.}}{\sim} p$ and $(Y_n)_{n \geq 1} \overset{\text{i.i.d.}}{\sim} p$, but each X_k typically depends on the corresponding 'pair' Y_k. The antithetic variable estimator

$$I^{(n)}_{\hat{p},\text{anti}}(f) := \frac{1}{2n} \sum_{k=1}^{n} [f(X_k) + f(Y_k)]$$

is clearly unbiased and strongly consistent estimator of $\mathbb{E}_p[f(X)]$.

Proposition 5.13. *The variance of the antithetic variable estimator is*

$$\operatorname{Var}\big(I^{(n)}_{\hat{p},anti}(f)\big) = \frac{1}{2n}\big[\operatorname{Var}_p f(X) + \operatorname{Cov}_{\hat{p}}\big(f(X), f(Y)\big)\big].$$

Therefore, if $\operatorname{Cov}_{\hat{p}}\big(f(X), f(Y)\big) = \operatorname{Cov}\big(f(X_1), f(Y_1)\big) \leq 0$ *then* $\operatorname{Var}\big(I^{(n)}_{\hat{p},anti}(f)\big) \leq \operatorname{Var}\big(I^{(2n)}_{p}(f)\big)$.

Note that $I^{(2n)}_p(f)$ has the same total number of samples as $I^{(n)}_{\hat{p},\text{anti}}(f)$, so they have roughly equal computational complexity.

Useful antithetic variables can be found with the inverse c.d.f. method.

Proposition 5.14. *Suppose* F^{-1} *is a generalised inverse c.d.f. of* p*, and* $f : \mathbb{R} \to \mathbb{R}$ *is monotonic. Define* $X_k = F^{-1}(U_k)$ *and* $Y_k = F^{-1}(1 - U_k)$ *where* $(U_k)_{k \geq 0} \overset{i.i.d.}{\sim} \mathcal{U}(0, 1)$*. Then,* $\operatorname{Cov}(f(X_1), f(Y_1)) \leq 0$*.*

Proof. (*) Without loss of generality, we may assume f increasing. Then also $\bar{f}(x) = f(x) - \mathbb{E}_p[f(X)]$ and $g := \bar{f} \circ F^{-1}$ are increasing. If $\operatorname{Var}(g(U)) = 0$, the claim is trivial, so assume $\operatorname{Var}(g(U)) > 0$.

Because of symmetry

$$\operatorname{Cov}(f(X_1), f(Y_1)) = \int_0^1 g(u)g(1-u)\mathrm{d}u = 2\int_0^{1/2} g(u)g(1-u)\mathrm{d}u.$$

Recall $\mathbb{E}[g(U)] = 0$, so there exists $u_0 \in (0, 1)$ such that $g(u) \leq 0$ for $u < u_0$ and

7. Such $\hat{p}$ is also known as a *coupling* of p with itself.

$g(u) \geq 0$ for $u > u_0$. Assume $u_0 < 1/2$ and notice that then

$$\begin{aligned}\int_0^{1/2} g(u)g(1-u)\mathrm{d}u &= \int_0^{u_0} g(u)g(1-u)\mathrm{d}u + \int_{u_0}^{1/2} g(u)g(1-u)\mathrm{d}u \\ &\leq \int_0^{u_0} g(u)g(1-u_0)\mathrm{d}u + \int_{u_0}^{1/2} g(u)g(1-u_0)\mathrm{d}u \\ &\leq g(1-u_0)\int_0^1 g(u)\mathrm{d}u = 0.\end{aligned}$$

The case $u_0 = 1/2$ is easy, and if $u_0 > 1/2$, then we may use the proof above with $\tilde{g}(u) := -g(1-u)$. □

5.4 Control variates (*)

Definition 5.15 (Control variates)**.** Suppose $(X_k, W_k)_{k\geq 1} \overset{\text{i.i.d.}}{\sim} \hat{p}$ with $X_k \sim p$ ($\mathbb{X}$-valued) and W_k is a zero-mean random number. Let $\beta \in \mathbb{R}$, then

$$I^{(n)}_{p,\mathrm{ctrl}}(f) := \frac{1}{n}\sum_{k=1}^{n}\big[f(X_k) + \beta W_k\big].$$

is an unbiased and strongly constent estimator of $\mathbb{E}_p[f(X)]$.

Example 5.16. Suppose that we are interested in estimation of $\mathbb{E}_p[f(X)]$, where p is $N(\mu, \sigma^2)$, but f is a complicated function. Then $X_k \sim N(\mu, \sigma^2)$ and we may use $W_k = X_k - \mu$ as a control variate.

Example 5.17. Suppose that $X_k = F^{-1}(U_k)$, where $U_k \sim \mathcal{U}(0,1)$. We can always use $W_k = U_k - 0.5$ as control variates.

Proposition 5.18. *We have the expression of the variance*

$$\mathrm{Var}\big(I^{(n)}_{p,ctrl}(f)\big) = \frac{1}{n}\big[\mathrm{Var}_p(f(X)) + \beta^2\mathrm{Var}(W_1) + 2\beta\mathrm{Cov}(f(X_1), W_1)\big].$$

If $\mathrm{Cov}(f(X_1), W_1) \neq 0$, it is possible (in principle) to choose β such that the variance is reduced.

Remark 5.19. Theoretically, the best value is

$$\beta_* = -\mathrm{Cov}(f(X_1), W_1)/\mathrm{Var}(W_1),$$

which leads into

$$\mathrm{Var}\big(I^{(n)}_{p,\mathrm{ctrl}}(f)\big) = \frac{1}{n}\big[(1 - \mathrm{Corr}(f(X_1), W_1)^2)\mathrm{Var}_p(f(X))\big].$$

Remark 5.20. The value β_* is often unknown, but β may be chosen as an empirical approximation of β_* based on preliminary simulation of (X_k, W_k). Finding suitable control variates is problem-specific.

6 Markov chain Monte Carlo

Up to this point, we have considered only methods based on i.i.d. random sequences. Sometimes it is useful to construct non-i.i.d. sequence $X_1, X_2, \ldots$ such that we can approximate as before

$$\frac{1}{n}\sum_{k=1}^{n} f(X_k) \approx \mathbb{E}_p[f(X)].$$

In this section, we will focus on Markov chains like this.

Intuitively, X_k are going to be 'approximately from p' for large k and X_k will be 'approximately independent' of X_j if $|k-j|$ is large.

6.1 Recap of some Markov chain theory

We will restate some concepts and key results related to (time-homogeneous) Markov chains, which you may have seen in earlier courses[8]. We focus here on countable or finite $\mathbb{S}$.

Definition 6.1 (Markov chain)**.** The random variables $(X_k)_{k\geq 0}$ form a Markov chain, if for all $k \in \mathbb{N}$ and $x_0, \ldots, x_k \in \mathbb{S}$,

$$\mathbb{P}(X_k = x_k \mid X_0 = x_0, \ldots, X_{k-1} = x_{k-1}) = \mathbb{P}(X_k = x_k \mid X_{k-1} = x_{k-1}).$$

Definition 6.2 (Transition probability, initial distribution)**.** The *transition probability* or *transition matrix* P of a (time-homogeneous) Markov chain $(X_k)_{k\geq 0}$ on $\mathbb{S}$ consists of

$$P(x,y) = \mathbb{P}(X_{k+1} = y \mid X_k = x) \qquad \text{for all } k \in \mathbb{N} \text{ and } x, y \in \mathbb{S}.$$

The distribution of $(X_k)_{k\geq 0}$ is called *initial distribution* $\lambda(x) = \mathbb{P}(X_0 = x)$ for all $x \in \mathbb{S}$.

Recall that λ and P characterise the distribution of $(X_k)_{k\geq 0}$.

Taking λ as a row vector and P as a matrix (you can think of finite, but the same ideas work with countable case), then

$$(\lambda P)(x) = \sum_{y\in\mathbb{S}} \lambda(y)P(y,x) = \sum_{y\in\mathbb{S}} \mathbb{P}(X_1 = x, X_0 = y) = \mathbb{P}(X_1 = x).$$

This argument can be iterated to find out that $(\lambda \overbrace{P \cdots P}^{k \text{ times}})(x) = (\lambda P^k)(x) = \mathbb{P}(X_k = x)$.

Definition 6.3 (Invariant distribution)**.** If $\pi = \big(\pi(x)\big)_{x\in\mathbb{S}}$ is a p.m.f. on $\mathbb{S}$ taken as a row vector, and if

$$\pi P = \pi, \qquad (\text{that is, } (\pi P)(x) = \pi(x) \text{ for all } x \in \mathbb{S}),$$

then π is the *invariant* or *stationary distribution* of P.

8. MATA271 Stochastic Models.

Definition 6.4 (Irreducibility)**.** Markov chain, or equivalently its transition probability, is *irreducible* if for any $x, y \in \mathbb{S}$ there exists $n = n(x, y) \in \mathbb{N}$ such that

$$\mathbb{P}(X_n = y \mid X_0 = x) > 0.$$

We state the following well-known theorem without proof:

Theorem 6.5 (Markov chain strong law of large numbers)**.** *Suppose π is a p.m.f. on $\mathbb{S}$ and that P is an irreducible transition probability on $\mathbb{S}$ with invariant distribution π.*

Let $(X_k)_{k\geq 0}$ be a Markov chain with transition probability P and with any initial distribution, then for any $f : \mathbb{S} \to \mathbb{R}$ such that $\mathbb{E}_\pi[f(X)]$ is finite,

$$\frac{1}{n}\sum_{k=1}^{n} f(X_k) \xrightarrow{n\to\infty} \mathbb{E}_\pi[f(X)] \qquad \textit{almost surely.}$$

For completeness, let us restate also convergence in distribution, which is often of considered instead of Theorem 6.5 in Markov chain theory.

Definition 6.6 (Periodicity, aperiodicity)**.** A Markov chain $(X_k)_{k\geq 0}$ is *periodic* with period $m \in \mathbb{N}$ if there exists a partition $S_0, \ldots, S_{m-1}$ of $\mathbb{S}$, where S_k are non-empty, such that

$$\mathbb{P}(X_n \in S_{(n \bmod m)} \mid X_0 \in S_0) = 1 \qquad \text{for all } n \in \mathbb{N}.$$

The chain is *aperiodic* if it is not periodic with any period $m \geq 2$.

Theorem 6.7. *Suppose P is irreducible and aperiodic, with invariant distribution π. If X_n is a Markov chain with transition probability P with any initial distribution,*

$$\mathbb{P}(X_n = x) \xrightarrow{n\to\infty} \pi(x) \qquad \textit{for any } x \in \mathbb{S}.$$

Remark 6.8. Usually in sampling, we are rather more interested in SLLN in Theorem 6.5, but in some cases Theorem 6.7 may be of interest as well. MCMC chains are rarely periodic, so we usually get Theorem 6.7 automatically. We shall not consider aperiodicity in detail further.

6.2 Reversibility

We shall consider next a Markov chain concept, which may not appear in a general course on Markov chain theory, but proves very useful in checking invariance in the MCMC context.

Definition 6.9 (Reversibility)**.** Suppose P is a transition probability and π is a p.m.f. on $\mathbb{S}$. If

$$\pi(x)P(x, y) = \pi(y)P(y, x) \qquad \text{for all } x, y \in \mathbb{S}, \tag{13}$$

then P is *reversible with respect to* π, or *π-reversible*. (Condition (15) is also known as the *detailed balance*.)

Proposition 6.10. *If P is π-reversible, then π is invariant for P.*

Proof. $(\pi P)(x) = \sum_y \pi(y)P(y,x) = \pi(x)\sum_y P(x,y) = \pi(x)$. □

Remark 6.11. The contrary does not hold true. That is, if π is invariant for P, it *does not imply* π-reversibility.

Remark 6.12. Suppose P is reversible with respect to π and $X_0 \sim \pi$. Then the joint distribution of (X_0, X_1) is symmetric,

$$\mathbb{P}(X_0 = x,\, X_1 = y) = \pi(x)P(x,y) = \pi(y)P(y,x) = \mathbb{P}(X_0 = y,\, X_1 = x).$$

In other words, $(X_0, X_1) \overset{d}{=} (X_1, X_0)$. This generalises to

$$(X_0, X_1, \ldots, X_n) \overset{d}{=} (X_n, X_{n-1}, \ldots, X_0),$$

which can be understood so that the Markov chain *initialised from the stationarity distribution* can be 'time-reversed' without affecting its distribution.

The reversibility can also be understood in terms of the 'backwards' transition probability being equal to the 'forward' transition probability (assuming again $X_0 \sim \pi$),

$$\begin{aligned}\mathbb{P}(X_0 = i \mid X_1 = j) &= \frac{\mathbb{P}(X_0 = i, X_1 = j)}{\mathbb{P}(X_1 = j)} = \frac{\pi(j)P(j,i)}{\pi(j)} \\ &= \mathbb{P}(X_1 = i \mid X_0 = j).\end{aligned}$$

6.3 The Metropolis-Hastings algorithm on discrete $\mathbb{X}$

Assume $\mathbb{X}$ is discrete and p is a p.m.f. on $\mathbb{X}$, and for each $x \in \mathbb{X}$ we have a proposal p.m.f. $q(x, \cdot)$ on $\mathbb{X}$ which we can draw samples from.

Algorithm 6.13 (Metropolis-Hastings)**.** Choose some initial value $X_0 \equiv x_0$ with $p(x_0) > 0$ and iterate for $k = 1, 2, \ldots$

(a) Generate $Y_k \sim q(X_{k-1}, \cdot)$.

(b) Generate $U_k \sim \mathcal{U}(0,1)$, and if $U_k \le \alpha(X_{k-1}, Y_k)$ *accept* and set $X_k = Y_k$, otherwise *reject* and set $X_k = X_{k-1}$, where the *acceptance probability* α is defined as follows:

$$\alpha(x,y) := \begin{cases} \min\left\{1, \dfrac{p(y)}{p(x)}\dfrac{q(y,x)}{q(x,y)}\right\}, & p(x)q(x,y) > 0, \\ 0, & \text{otherwise.} \end{cases}$$

Finally, for some function $f : \mathbb{X} \to \mathbb{R}$, report

$$I^{(n)}_{p,q,\mathrm{MH}}(f) := \frac{1}{n}\sum_{k=1}^{n} f(X_k)$$

as the Metropolis-Hastings approximation of $\mathbb{E}_p[f(X)]$.

Remark 6.14. In Algorithm 6.13,

(i) The distribution p is called the *target distribution*.

(ii) Unnormalised distributions $p_u(x) = Z_p p(x)$ and $q_u(x,y) = Z_q q(x,y)$ can be used, because

$$\frac{p_u(y)}{p_u(x)}\frac{q_u(y,x)}{q_u(x,y)} = \frac{Z_p p(y)}{Z_p p(x)}\frac{Z_q q(y,x)}{Z_q q(x,y)} = \frac{p(y)}{p(x)}\frac{q(y,x)}{q(x,y)}.$$

(iii) The accept-reject step (b) is implemented in practice by drawing $U_k \sim \mathcal{U}(0,1)$ and setting

$$X_k := \begin{cases} Y_k, & \text{if } U_k < \dfrac{p_u(Y_k)}{p_u(X_{k-1})}\dfrac{q_u(Y_k, X_{k-1})}{q_u(X_{k-1}, Y_k)} \\ X_{k-1}, & \text{otherwise.} \end{cases}$$

In many cases, it is easier (and numerically more stable) to compute

$$r_u(x,y) := \log p_u(y) + \log q_u(y,x) - \log p_u(x) - \log q_u(x,y),$$

and then accept if $U_k < \exp\big(r_u(X_{k-1}, Y_k)\big)$ and reject otherwise.

(iv) There is no need to define $\alpha(x,y)$ for $p(x)q(x,y) = 0$ in practice, because $p(X_{k-1})q(X_{k-1}, Y_k) = 0$ never occurs (almost surely).

Proposition 6.15. *The Metropolis-Hastings algorithm:*

(i) Defines a Markov chain on the support of p,

$$\mathbb{S} := \{x \in \mathbb{X} \,:\, p(x) > 0\}.$$

(ii) Has transition probability K given as

$$K(x,y) = q(x,y)\alpha(x,y) + \rho(x)\mathbf{1}\left(y = x\right), \quad x, y \in \mathbb{S},$$

where the probability of rejection $\rho(x)$ *can be given as*

$$\rho(x) = 1 - \sum_{y \in \mathbb{X}} q(x,y)\alpha(x,y).$$

Proof. The transition probability is straightforward to write. Let us then check that $X_n \in \mathbb{S}$. For any $x \in \mathbb{S}$ and $y \in \mathbb{X} \setminus \mathbb{S}$

$$\mathbb{P}(X_{n+1} = y \mid X_n = x) = q(x,y)\alpha(x,y) = 0,$$

because $\alpha(x,y) = 0$. This means $\mathbb{P}(X_{n+1} \in \mathbb{S}) = 1$ if $X_n \in \mathbb{S}$, and by definition, $X_0 = x_0 \in \mathbb{S}$. □

Proposition 6.16. *The Metropolis-Hastings transition probability K is reversible with respect to the target distribution p.*

Proof. Exercise. □

Now, Propositions 6.15 and 6.10 applied with Theorem 6.5 imply the strong consistency.

Corollary 6.17. *If the Metropolis-Hastings transition probability K targetting p is irreducible on $\mathbb{S} = \mathrm{supp}(p)$, then for any function $f : \mathbb{X} \to \mathbb{R}$ with $\mathbb{E}_p[f(X)]$ finite*

$$I^{(n)}_{p,q,\mathrm{MH}}(f) = \frac{1}{n}\sum_{k=1}^{n} f(X_k) \xrightarrow{n\to\infty} \mathbb{E}_p[f(X)] \quad \textit{(almost surely).}$$

Remark 6.18. Irreducibility is ensured by proper choice of proposal distributions $q(x, y)$. The proposal distributions need to be defined so that every point $y \in \mathbb{S}$ is reachable from any $x \in \mathbb{S}$ in $n = n(x, y)$ steps.

Example 6.19. Let $p(x) = x/Z_p$ for $x \in \mathbb{X} := \{1, \ldots, m\}$ with $Z_p = \sum_{x=1}^{m} x$. Let us design a Metropolis-Hastings algorithm targetting p.

Step 1: Choose a proposal distribution $q(x, y)$. It needs to be easy to simulate and to determine an irreducible chain. A simple distribution that 'will do' is drawing $Y_k \sim \mathcal{U}(\mathbb{X})$ independent of X_{k-1}, so

$$q(x, y) = q(y) = 1/m, \quad y \in \mathbb{X}$$

This proposal scheme is irreducible, because for all $x, y \in \mathbb{X}$,

$$\begin{aligned}\mathbb{P}(X_1 = y \mid X_0 = x) &\geq q(x, y) \min\left\{1, \frac{p(y)}{p(x)}\frac{q(y, x)}{q(x, y)}\right\} \\ &= \frac{1}{m}\min\left\{1, \frac{y}{x}\right\} > 0.\end{aligned}$$

That is, we can get from any $x \in \mathbb{S}$ to any $y \in \mathbb{S}$ in one step (we can take $n(x, y) \equiv 1$ in Definition 6.4).

Step 2: write down the algorithm. Start from $X_0 = 1$ (say), and for $k = 1, \ldots, n$ do

(a) Simulate $Y_k \sim U\{1, 2, ..., m\}$.

(b) Simulate $U_k \sim \mathcal{U}(0, 1)$ and if

$$U_k \quad \leq \quad \frac{Y_k}{X_{k-1}}$$

set $X_k = Y_k$, otherwise set $X_k = X_{k-1}$.

```
function imh_example(m=30, n=10_000)
    X = zeros(n); X[1] = 1
    for k = 2:n
        x = X[k-1]
        y = ceil(m*rand())  # y ~ U{1,2,...,m}
        if (rand() < y/x)
            X[k] = y
        else
            X[k] = x
        end
    end
    X
end
```

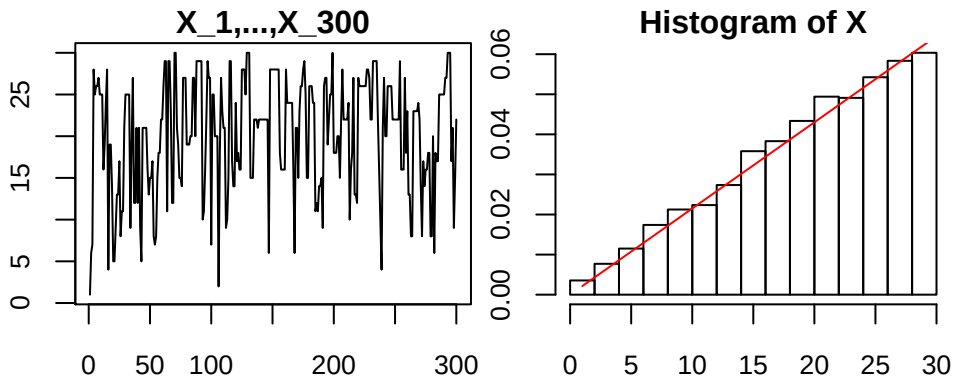


Figure 9: Left: x-axis is Markov chain step counter $k = 1, 2, \ldots, 300$ and y-axis is Markov chain state X_k. Right: histogram of $X_1, X_2, \ldots, X_n$ for $n = 10,000$ along with p.

Example 6.19 is an instance of the following class of Metropolis-Hastings algorithms.

Definition 6.20. Metropolis-Hastings algorithm with $q(x, y) = q(y)$, that is, proposal is independent of current state, is referred to as *independence sampler* or *independent Metropolis-Hastings* (IMH).

The IMH acceptance probability takes the form

$$\alpha(x, y) = \min\left\{1, \frac{p(y)q(x)}{p(x)q(y)}\right\} = \min\left\{1, \frac{w(y)}{w(x)}\right\},$$

where $w(x) = p(x)/q(x)$ for $q(x) > 0$. In order the IMH to be irreducible, we need $q(x) = 0$ implies $p(x) = 0$.

Remark 6.21. (Self-normalised) importance sampling can always be used instead of the IMH.

6.4 The Metropolis-Hastings algorithm on $\mathbb{X} = \mathbb{R}^d$

The Metropolis-Hastings (Algorithm 6.13) generalises directly to continuous setting, that is, $\mathbb{X} = \mathbb{R}^d$:

(i) p is a probability density on $\mathbb{R}^d$.

(ii) $q(x, \cdot)$ is a probability density on $\mathbb{R}^d$ for each $x \in \mathbb{R}^d$.

Everything else in Algorithm 6.13 remains unchanged.

Fact 6.22. The MH algorithm defines a Markov chain on $\mathbb{S} := \{x \in \mathbb{R}^d : p(x) > 0\}$. The transition probability K can be written as

$$\mathbb{P}(X_n \in A \mid X_{n-1} = x) =: K(x, A) = \int_A k(x, y)\mathrm{d}y + \rho(x)\mathbf{1}(x \in A), \qquad (14)$$

where $k(x, y) := q(x, y)\alpha(x, y)$ is a *sub-probability density* for each $x \in \mathbb{X}$ and $\rho(x) = 1 - \int k(x, y)\mathrm{d}y$ is the probability of rejection.

Precise definition of Markov chains on $\mathbb{S} \subset \mathbb{R}^d$ will be out of the scope of the course, but we shall see how the necessary ingredients are defined in this case. The article [20] by Nummelin contains a minimal self-contained proofs about the strong law of large numbers and more.

Definition 6.23. The $\mathbb{R}^d$-valued Markov chain is $(X_k)_{k\geq 1}$ is p-reversible, if $X_0 \sim p$ then $(X_0, X_1) \stackrel{d}{=} (X_1, X_0)$. That is, $\mathbb{P}(X_0 \in A, X_1 \in B) = \mathbb{P}(X_0 \in B, X_1 \in A)$.

Proposition 6.24. *Markov transition probability defined as in* (14) *is reversible with respect to a p.d.f. p on $\mathbb{X}$ if*

$$p(x)k(x,y) = p(y)k(y,x) \qquad \textit{for all } x, y \in \mathbb{X}. \tag{15}$$

The condition (15), sometimes also called as detailed balance, is essentially equivalent[9] with reversibility with transition probabilities of the form (14). This is identical to the definition of reversibility in the discrete case for $x \neq y$, which turns out to be sufficient. The proof of reversibility of Metropolis-Hastings is identical to the discrete case.

The irreducibility condition in the continuous case is likewise slightly different, as there is zero probability of reaching any single state from other states. Rather, any set of positive p-probability have to be reachable from everywhere.

Definition 6.25 (p-irreducibility)**.** Suppose that p is a p.d.f. on $\mathbb{S}$. The Markov chain $X_0, X_1, \ldots$ if p-irreducible if for any $x \in \mathbb{S}$ and any set $A \subset \mathbb{S}$ such that $\int_A p(y)\mathrm{d}y > 0$, there exists $n = n(x, A) < \infty$ such that

$$\mathbb{P}(X_n \in A \mid X_0 = x) > 0.$$

The proposal densities $q(x, y)$ are chosen to satisfy this condition.

We state the following general consistency theorem without proof[10]

Theorem 6.26. *If the Metropolis-Hastings algorithm is p-irreducible, then for any function f with $\mathbb{E}_p|f(X)| < \infty$, the MH-estimate is (strongly) consistent*

$$I^{(n)}_{p,q,\mathrm{MH}}(f) = \frac{1}{n}\sum_{k=1}^{n} f(X_k) \xrightarrow{n\to\infty} \mathbb{E}_p[f(X)] \qquad \textit{(almost surely).}$$

Note that Theorem 6.26 holds both when $\mathbb{X}$ is discrete or when $\mathbb{X} = \mathbb{R}^d$.

Example 6.27. Suppose want to simulate the standard normal distribution $X \sim N(0, 1)$. The target density is

$$p(x) \propto p_u(x) = \exp(-x^2/2).$$

Step 1: Choose the proposal distribution. We need something simple that can 'take us everywhere' (for irreducibility). Fix a constant $a > 0$ and choose

9. To be precise, the continuous part $k(x, y)$ in the representation of (14) is unique only up to a set of measure zero. So the statement would be 'there exists a k such that...'.

10. The proof follows, for example, from Corollary 2 of Tierney [29] along with Theorem 17.0.1 of Meyn and Tweedie [17]

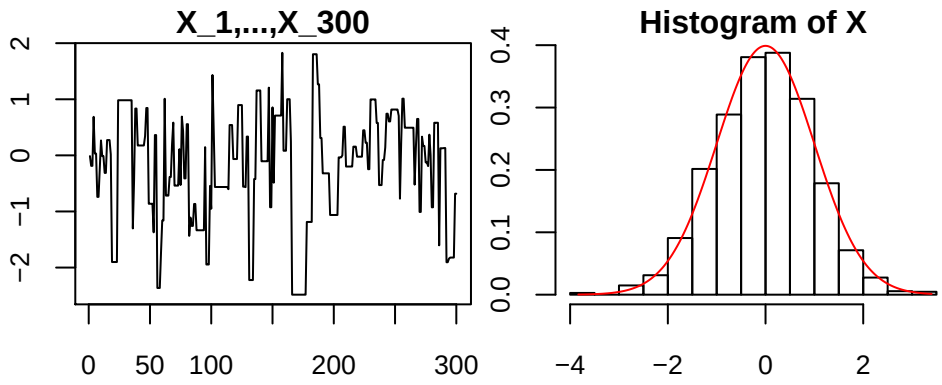


Figure 10: Simulation of Example 6.27: MCMC samples (left) and histogram approximation of the theoretical density (right).

a new point uniformly at random in a window of length $2a$ centred at x. The proposal density is

$$q(x, y) = \frac{1}{2a} \mathbf{1}\left(x - a < y < x + a\right).$$

Notice that $q(x, y) = q(y, x)$; this simplifies the acceptance probability

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

Step 2: Write the MCMC algorithm. Start from $X_0 = 0$ (say), and iterate for $k = 1, \ldots, n$:

(a) Simulate $Z_k \sim \mathcal{U}(-a, a)$ and set $Y_k = X_{k-1} + Z_k$.

(b) Simulate $U_k \sim \mathcal{U}(0, 1)$ and set

$$X_k = \begin{cases} Y_k, & \text{if } U_k \leq \exp\left(r(X_{k-1}, Y_k)\right), \\ X_{k-1}, & \text{otherwise.} \end{cases}$$

where $r(x, y) = \log p_u(y) - \log p_u(x) = -y^2/2 + x^2/2$.

```
function rwm_example(a=3, n=10_000)
    X = zeros(n); x = 0; L_px = -.5*x^2
    for k = 1:n
        y = x + (2rand()-1)*a
        L_py = -.5*y^2      # NB L_px calculated only once!
        if (rand() < exp(L_py-L_px))
            x = y; L_px = L_py
        end
        X[k] = x
    end
    X
end
```

Example 6.27 belongs to the following class of Metropolis-Hastings algorithms.

Definition 6.28. If $q(x,y) = q(y,x)$ for all $x, y \in \mathbb{X}$, then $\alpha(x,y) = \min\{1, p(y)/p(x)\}$. Such an algorithm is often called a *Metropolis* algorithm. More specifically, in a *symmetric random walk Metropolis* algorithm

$$Y_n = X_{n-1} + Z_n, \qquad Z_n \sim \tilde{q},$$

where the increment density $\tilde{q}$ is symmetric: $\tilde{q}(z) = \tilde{q}(-z)$ for all $z \in \mathbb{R}^d$.

The symmetricity of $\tilde{q}$ implies $q(x,y) = \tilde{q}(y-x) = \tilde{q}(x-y) = q(y,x)$. It is common to take $\tilde{q}$ to be density of $N(0,\Sigma)$, which implies that $Y_n \mid (X_{n-1} = x) \sim N(x,\Sigma)$.

Example 6.29 (Bivariate distribution with Gaussian random walk Metropolis).

$$\log p_u(x) = -\frac{1}{2} y(x)^T \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}^{-1} y(x), \qquad \text{where} \qquad y(x) = \begin{pmatrix} a^{-1}x_1 \\ ax_2 + ab(x_1^2 + a^2) \end{pmatrix},$$

and with $a = b = 1$.

For a proposal distribution q we want something simple to sample. Let's try bivariate standard normal, that is,

$$Y_k = X_{k-1} + Z_k, \qquad Z_k \sim N(0, I_2).$$

Note that this is symmetric random walk Metropolis algorithm. We choose to start from $x_0 = (0,0)^T$.

```
using Distributions
function log_p(x; a=1, b=1) # Log-pdf of a 'banana-shaped' distribution
  y = [x[1]/a, x[2]*a + a*b*(x[1]^2 + a^2)]
  logpdf(MvNormal([1 0.9; 0.9 1]), y)
end
function metropolis(n=10_000, d=2, log_p=log_p)
    X = zeros(d,n); x = zeros(d); px = log_p(x)
    for k = 1:n
        y = x + randn(2); py = log_p(y)  # Proposal & its density value
        if rand() < exp(py-px)
            x = y; px = py               # Accept
        end
        X[:,k] = x                       # Save output
    end
    X
end
```

6.5 On tuning of random-walk Metropolis (*)

Suppose that $\hat{q}$ is some symmetric distribution, that is, $\hat{q}(z) = \hat{q}(-z)$, and let $L \in \mathbb{R}^{d\times d}$ be an invertible matrix. If the proposals Y_k are formed as follows

$$Y_k = X_{k-1} + L\hat{Z}_k, \qquad \hat{Z}_k \sim \hat{q}.$$

The question is how the proposal 'shape/size' L should be chosen so that the algorithm would be 'efficient'.

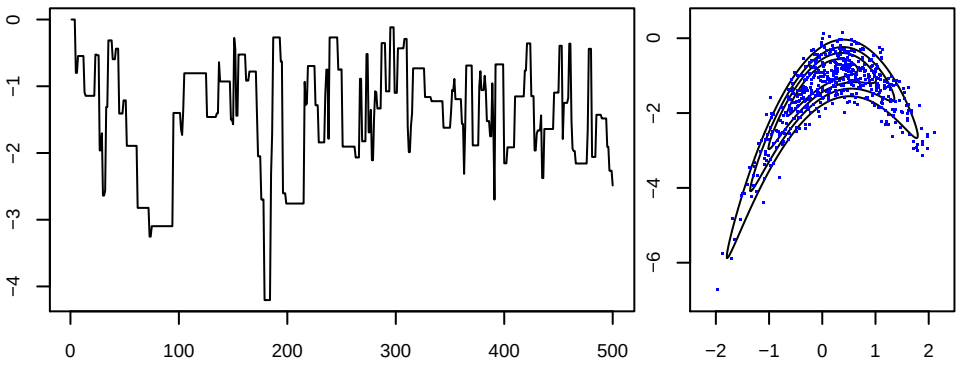


Figure 11: Simulation of Example 6.29: Second coordinate of the MCMC (left); The samples and the density contours (right).

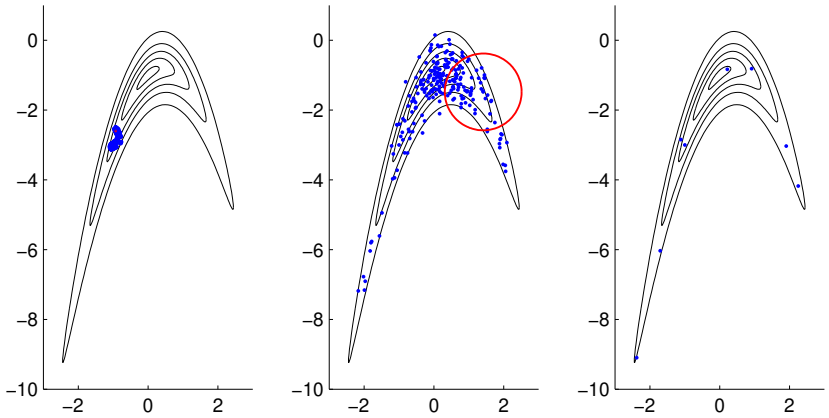


Figure 12: 1000 samples of the random walk Metropolis algorithm in $\mathbb{R}^2$ with $\tilde{q} = N(0, \Sigma)$. Contours of 'banana-shaped' p are shown in black.

Remark 6.30. There are some theoretical optimality results determining which L is 'the best' when $\hat{q}$ is standard normal [e.g. 27].

(a) First rule of thumb: Set $LL^T \approx \theta \mathrm{Cov}(p)$ where $\theta \in (0, \infty)$ is a scaling parameter.

(b) Second rule of thumb: Set θ such that around 25% of the samples should be accepted on average.[11]

These heuristics are often useful when p is (close to) unimodal.

Because $\mathrm{Cov}(p)$ is usually not available, $\mathrm{Cov}(p)$ is often estimated by a 'trial' MCMC targetting p, and θ is found also by trial and error.

Remark 6.31. There are various *adaptive MCMC* algorithms which can be used to automatise this process, and learn L 'progressively' [e.g. 11, 2]. Such methods have been observed to work well in practice, but the theoretical results ensuring the validity of the methods require subtle technical conditions.

Example 6.32. Implementation of an adaptive MCMC which finds 'good' L automatically [31].

```
using LinearAlgebra
function ram_adapt!(C, z, k, acc; gam=0.66, acc_opt=0.234)
  nz = norm(z); u = nz>0 ? z/nz : 0*z; step = (k+1)^(-gam); fact = acc-acc_opt
  dx = sqrt(step*abs(fact))*(C.L * (z/nz))
  if fact >= 0 lowrankupdate!(C, dx) else lowrankdowndate!(C, dx) end
end
function adapt_mcmc(log_p, x0, n)
  d = length(x0);  x = x0;  p_x = log_p(x); C = cholesky(diagm(ones(d)))
  X = zeros(d, n); acc = 0; z = zeros(d)
  for k = 1:n
      z = randn(d); y = x + C.L * z                # Proposal
      p_y = log_p(y); alpha = min(1, exp(p_y-p_x)) # Acc.prob.
      if (rand() <= alpha)
          x = y; p_x = p_y; acc += 1
      end
      X[:,k] = x
      ram_adapt!(C, z, k, alpha)     # Adapt the proposal covariance
  end
  (X=X, L=C.L, acc_rate=acc/n)
end
```

6.6 Componentwise updates

In higher dimensions, it is often difficult to design efficient proposal distributions $q(x, y)$. Instead, it is easier to design rules to update a *single coordinate* or a *block of coordinates* in each iteration.

In order to consider such updates, consider $\mathbb{X}$ to be d-dimensional, $\mathbb{X} = \mathbb{X}_1^d$; for instance, $\mathbb{X} = \mathbb{Z}^d$ or $\mathbb{X} = \mathbb{R}^d$. Let us introduce the following shorthand notation

$$x^{(-i)} := (x^{(1)}, \ldots, x^{(i-1)}, x^{(i+1)}, \ldots, x^{(d)})$$

for the vector $x \in \mathbb{X}$ with i:th coordinate omitted.

11. The theoretical value, 0.234, is optimal in high dimensions under very strong assumptions.

Suppose p is a p.m.f. (p.d.f.) on $\mathbb{X}$, and denote its i:th conditional with respect to other variables as

$$p_{i|-i}(x^{(i)} \mid x^{(-i)}) = \frac{p(x)}{p_{-i}(x^{(-i)})},$$

whenever the marginal $p_{-i}(x^{(-i)}) > 0$, where

$$p_{-i}(x^{(-i)}) := \sum_{z \in \mathbb{Z}} p(x^{(1)}, \ldots, x^{(i-1)}, z, x^{(i+1)}, \ldots, x^{(d)})$$
$$\left(p_{-i}(x^{(-i)}) := \int p(x^{(1)}, \ldots, x^{(i-1)}, z, x^{(i+1)}, \ldots, x^{(d)}) \mathrm{d}z \right).$$

Algorithm 6.33 ((Random scan) Metropolis-within-Gibbs)**.** Suppose that $q_i(x^{(i)}, \cdot \mid x^{(-i)})$ determines a p.m.f. (p.d.f.) on $\mathbb{X}_1$ for each $x \in \mathbb{X}$ and for all $i = 1, \ldots, d$. Choose some $X_0 \equiv x_0$ with $p(x_0) > 0$ and iterate for $k = 1, \ldots, n$

(a) Draw random coordinate index $I_k \sim \mathcal{U}\{1, \ldots, d\}$.

(b) Set $X_k^{(-I_k)} = X_{k-1}^{(-I_k)}$.

(c) Simulate $Y_k^{(I_k)} \sim q_{I_k}(X_{k-1}^{(I_k)}, \cdot \mid X_{k-1}^{(-I_k)})$

(d) With probability $\alpha_{I_k}(X_{k-1}^{(I_k)}, Y_k^{(I_k)} \mid X_{k-1}^{(-I_k)})$ accept and set $X_k^{(I_k)} = Y_k^{(I_k)}$, otherwise set $X_k^{(I_k)} = X_{k-1}^{(I_k)}$, where

$$\alpha_i(x, y \mid z^{(-i)}) := \min \left\{ 1, \frac{p_{i|-i}(y \mid z^{(-i)})}{p_{i|-i}(x \mid z^{(-i)})} \frac{q_i(y, x \mid z^{(-i)})}{q_i(x, y \mid z^{(-i)})} \right\}.$$

NB: In practice, we calculate the ratio of conditionals as

$$\frac{p_{i|i-1}(y \mid z^{(-i)})}{p_{i|i-1}(x \mid z^{(-i)})} = \frac{p_u(z^{(1)}, \ldots, z^{(i-1)}, y, z^{(i+1)}, \ldots, z^{(d)})}{p_u(z^{(1)}, \ldots, z^{(i-1)}, x, z^{(i+1)}, \ldots, z^{(d)})},$$

and in case $p(x)$ is defined as a product of terms, of which only few depend on the i:th coordinate, the ratio simplifies...

Proposition 6.34. *Algorithm 6.33 is reversible with respect to p.*

Proof. (Discrete case) We may write the Markov transition in Algorithm 6.33 as follows

$$\begin{aligned} K(x, y) &= \sum_{i=1}^{d} \mathbb{P}(X_k = y \mid X_{k-1} = x, I_k = i) \mathbb{P}(I_k = i \mid X_{k-1} = x) \\ &= \frac{1}{d} \sum_{i=1}^{d} K_i(x, y), \end{aligned}$$

where $K_i(x, y) = \mathbb{P}(X_k = y \mid X_{k-1} = x, I_k = i)$ are Markov transition probabilities, which correspond to the steps (b), (c) and (d) of Algorithm 6.33.

In fact, given $I_k = i$ and $X_{k-1}^{(-i)} = z^{(-i)}$, (c) and (d) correspond a Metropolis-Hastings algorithm targetting $p_{i|-i}(\,\cdot \mid z^{(-i)})$ with proposals $q_i(x, y \mid z^{(-i)})$. If we denote its transition probability $\hat{K}_i(x, y \mid z^{(-i)})$, we have

$$K_i(x, y) = \hat{K}_i(x^{(i)}, y^{(i)} \mid x^{(-i)}) \mathbf{1}\left(y^{(-i)} = x^{(-i)}\right)$$

and then

$$\begin{aligned} p(x)K_i(x, y) &= p_{-i}(x^{(-i)}) p_{i|-i}(x^{(i)} \mid x^{(-i)}) \hat{K}_i(x^{(i)}, y^{(i)} \mid x^{(-i)}) \mathbf{1}\left(y^{(-i)} = x^{(-i)}\right) \\ &= p_{-i}(x^{(-i)}) p_{i|-i}(y^{(i)} \mid x^{(-i)}) \hat{K}_i(y^{(i)}, x^{(i)} \mid x^{(-i)}) \mathbf{1}\left(y^{(-i)} = x^{(-i)}\right) \\ &= p_{-i}(y^{(-i)}) p_{i|-i}(y^{(i)} \mid y^{(-i)}) \hat{K}_i(y^{(i)}, x^{(i)} \mid y^{(-i)}) \mathbf{1}\left(x^{(-i)} = y^{(-i)}\right) \\ &= p(y)K_i(y, x), \end{aligned}$$

where we first use reversibility of $\hat{K}_i(\,\cdot\,, \cdot \mid x^{(-i)})$ with respect to $p_{i|-i}(\,\cdot \mid x^{(-i)})$ and then the fact that the expression is non-zero with $x^{(-i)} = y^{(-i)}$.

The p-reversibility of K follows now easily:

$$p(x)K(x, y) = \frac{1}{d}\sum_{i=1}^{d} p(x)K_i(x, y) = \frac{1}{d}\sum_{i=1}^{d} p(y)K_i(y, x) = p(y)K(y, x). \qquad \square$$

Remark 6.35. In fact, the proof of Proposition 6.34 suggests that we may use multiple possible MCMC transitions, which we use at random. The mixture transition probability is reversible as long as the component transition probabilities are. And the mixing weights need not be uniform.

For instance, we could have K_1 being an independence sampler transition and K_2 a random-walk Metropolis transition, and choose randomly which update we follow.

Definition 6.36. *Gibbs sampling* is a specific instance of Metropolis-within-Gibbs, where the proposal distributions are the conditional distributions,

$$q_i(x, y \mid z^{(-i)}) = p_{i|-i}(y \mid z^{(-i)}).$$

Note that in Gibbs sampling, the acceptance probability $\alpha_i(x, y \mid z^{(-i)}) \equiv 1$.

Remark 6.37 (*). Algorithm 6.33 is valid also in the continuous case $\mathbb{X} = \mathbb{R}^d$. We cannot use Proposition 6.24 directly to verify reversibiity, but we need to check that if $X_0 \sim p$, then $(X_0, X_1) \stackrel{d}{=} (X_1, X_0)$. The proof follows similarly as in the discrete case

$$\begin{aligned} &\mathbb{P}(X_0 \in A, X_1 \in B) \\ &= \int_A \left[\int_B p_{-i}(x^{(-i)}) p_{i|-i}(x^{(i)} \mid x^{(-i)}) \hat{K}_i(x^{(i)}, y^{(i)} \mid x^{(-i)}) \mathbf{1}\left(y^{(-i)} = x^{(-i)}\right) \mathrm{d}x \right] \mathrm{d}y \\ &= \mathbb{P}(X_0 \in B, X_1 \in A). \end{aligned}$$

Example 6.38 (Ising model). Let $\mathbb{X} = \{0, 1\}^{\ell \times m}$ the set of all $\ell \times m$ binary matrices. We can think them as 'images' $x \in \mathbb{X}$ where $x^{(i,j)} = 0$ or 1 corresponds to (i, j):th pixel being black or white, respectively.

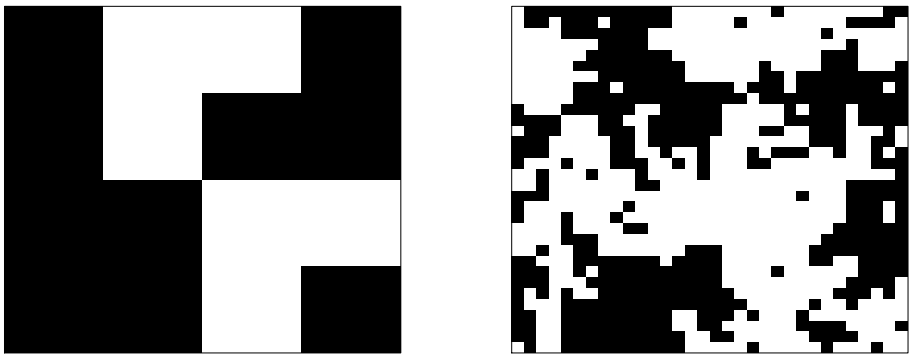


Figure 13: Left: Example 4-by-4 configuration with $\#x = 12$; right: realisation of the Ising model in $m = 32$, $\theta = 0.8$.

For $x \in \mathbb{X}$, denote $\#x$ for the number of disagreeing neighbours in x, which we may calculate by

$$\#x = \sum_{i=1}^{\ell} \sum_{j=1}^{m-1} \mathbf{1}\left(x^{(i,j)} \neq x^{(i,j+1)}\right) + \sum_{j=1}^{m} \sum_{i=1}^{\ell-1} \mathbf{1}\left(x^{(i,j)} \neq x^{(i+1,j)}\right).$$

The *Ising model* is defined as the following distribution on $\mathbb{X}$:

$$p(x) \propto \exp(-\theta \#x),$$

where $\theta > 0$ is a 'smoothing' parameter.

Example 6.39 (MCMC for the Ising model). Let $X_0^{(i,j)} \overset{\text{i.i.d.}}{\sim} \mathcal{U}\{0,1\}$, and do

(a) Draw random indices $I_k \sim \mathcal{U}\{1,\dots,\ell\}$, $J_k \sim U\{1,\dots,m\}$.
(b) Set $Y_k^{(I_k,J_k)} = 1 - X_{k-1}^{(I_k,J_k)}$.
(c) Set $X_k^{(i,j)} = X_{k-1}^{(i,j)}$ for all $(i,j) \neq (I_k, J_k)$.
(d) With probability $\alpha_{I_k,J_k}(X_{k-1}^{(I_k,J_k)}, Y_k^{(I_k,J_k)} \mid X_{k-1}^{(-(I_k,J_k))})$ set $X_k^{(I_k,J_k)} = Y_k^{(I_k,J_k)}$; otherwise set $X_k^{(I_k,J_k)} = X_{k-1}^{(I_k,J_k)}$, where

$$\alpha_{i,j}(x, y \mid z^{(-(i,j))}) = \min\left\{1, \exp\left[-\theta\big(\#(y, z^{(-(i,j))}) - \#(x, z^{(-(i,j))})\big)\right]\right\},$$

where $(x, z^{(-(i,j))})$ stands for the image where the (i,j):th pixel equals x and the rest are defined by $z^{(-(i,j))}$.

Remark 6.40. Note that $q_{i,j}$ here corresponds to a deterministic 'flip' of the (i,j):th pixel value. In fact, we shall see later that this choice of $q_{i,j}$ is the most efficient in terms of the *asymptotic variance*.

Remark 6.41. Note that in practice one should not re-calculate $\#(y, z^{(-(i,j))})$ and $\#(x, z^{(-(i,j))})$, but only their difference.

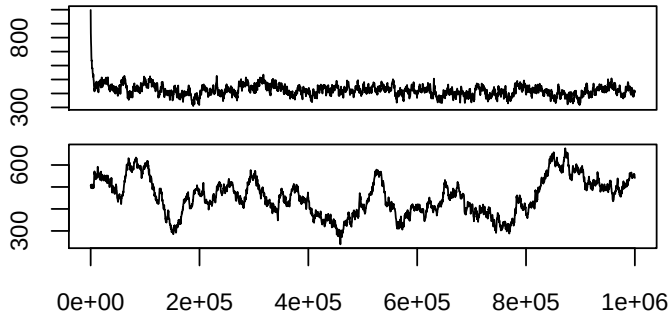


Figure 14: Trajectories of $f(X_k) = \#X_k$ (top) and $w(X_k)$ (bottom).

```
m = 32; n = 32; theta = 0.8; n = 100_000
function hashdiff(y, X, i, j, m, n)
    hash_y = 0; hash_x = 0; x = X[i,j]
    function check_ind!(i_, j_)
        hash_y += (y != X[i_,j_]);  hash_x += (x != X[i_,j_])
    end
    if i>1 check_ind!(i-1,j) end
    if i<m check_ind!(i+1,j) end
    if j>1 check_ind!(i,j-1) end
    if j<n check_ind!(i,j+1) end
    hash_y - hash_x
end
X = [rand(0:1) for i=1:m, j=1:m] # Independent random initialisation
for k = 1:n
  i = rand(1:m); j = rand(1:m) # Pick random index
  y = 1-X[i,j]                 # Propose swap 0<->1
  if rand() < exp(-theta*hashdiff(y, X, i, j, m, m))
      X[i,j] = y
  end
end
```

What would be good indicators to monitor the convergence of the Ising model simulation? We could look at:

- the function $f(x) = \#x$,
- the function $w(x) = \sum_{i,j} \mathbf{1}\left(x^{(i,j)} = 1\right)$, that is, the total number of white pixels.

Example 6.42 (Bayesian image recovery). Let X be an unknown true image,

$$X \sim \text{Ising}(\theta),$$

with θ known. Denote $p_0(x) = \mathbb{P}(X = x)$.

Suppose we do not observe X directly, but through a 'noisy channel'. At pixel $i, j = 1, 2, \ldots, m$ we observe

$$O^{(i,j)} = X^{(i,j)} + \epsilon^{(i,j)}, \qquad \text{with} \qquad \epsilon^{(i,j)} \overset{\text{i.i.d.}}{\sim} N(0, \sigma^2)$$

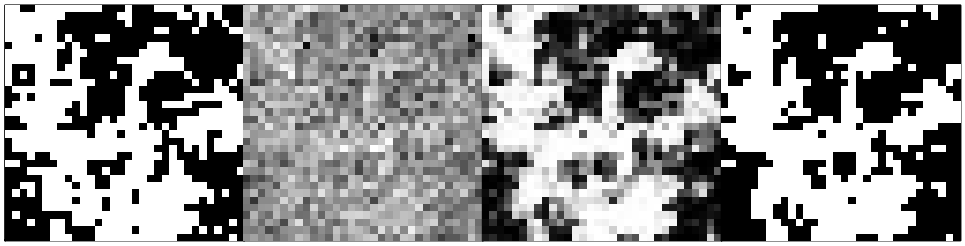


Figure 15: From the left: Simulation of the Ising model X with $\theta = 0.8$; the noisy observations O of X with $\sigma^2 = 1$; the posterior mean approximation by MCMC; the MAP approximation by MCMC (estiamtes with ten million samples).

and with σ known. The likelihood for $x^{(i,j)}$ is $L(x^{(i,j)}; o^{(i,j)}) = N(o^{(i,j)}; x^{(i,j)}, \sigma^2)$ so

$$L(x; o) \propto \prod_{i,j=1}^{m} \exp\left(- \frac{(x^{(i,j)} - o^{(i,j)})^2}{2\sigma^2} \right).$$

If we observe $O = o$ we are interested in the *posterior distribution* of X given $O = o$,

$$p(x) = \mathbb{P}(X = x \mid O = o) \propto L(x; o) p_0(x),$$

so we have

$$\log p_u(x) = -\frac{|x - o|^2}{2\sigma^2} - \theta \# x \qquad \text{where} \qquad |x - o|^2 = \sum_{i,j=1}^{m} (x^{(i,j)} - o^{(i,j)})^2.$$

We will simulate $X_1, \ldots, X_n \sim p$ with MCMC and use the samples to approximate the posterior mean and pixel-wise maximum a Posteriori (MAP) estimates $i = 1, \ldots, m^2$

$$\bar{X}^{(i)} := \frac{1}{n} \sum_{k=1}^{n} X_k^{(i)} \approx \mathbb{E}[X^{(i)} \mid O = o]$$
$$\mathbf{1}\left(\bar{X}^{(i)} > 1/2\right) \approx \arg\max_{x \in \{0,1\}} \mathbb{P}(X^{(i)} = x \mid O = o].$$

In order to implement the MCMC, we can recycle the implementation in Example 6.39 only modifying the acceptance probability $\alpha(y \mid x)$ to incorporate $-|x - o|^2/(2\sigma^2)$ factor.

Variants of Metropolis-within-Gibbs

Algorithm 6.33 introduced earlier is only variant of (Metropolis-within-)Gibbs sampling, in terms how $(I_k)_{k \geq 1}$ are chosen.

Random scan means we choose I_k at random, as in Algorithm 6.33. It is customary to take $I_k \sim \mathcal{U}\{1, \ldots, d\}$, but I_k can be chosen also from a non-uniform distribution over $\{1, \ldots, d\}$.

Deterministic scan version of the algorithm means I_k are not random, but deterministic. The common choice is to sweep $I_k = (k-1 \mod d) + 1$. Unlike the random scan version, the deterministic scan algorithm is *time-inhomogeneous*, but the 'skeleton' chain $(X_{dk})_{k\geq 0}$, is homogeneous, with composition of transition probabilities

$$\mathbb{P}(X_{dk} = y \mid X_{d(k-1)} = x) = (K_1 K_2 \cdots K_d)(x, y)$$

This transition probability is *not reversible* wrt. p in general, but is still p-invariant.

Random sweep is a hybrid of the two above: Simulate a random permutation of $\{1, \ldots, d\}$, and sweep through once in the corresponding order; simulate a new random permutation etc.

Remark 6.43. Metropolis-within-Gibbs moves can update a 'block' of coordinates instead of a single coordinate. The blocks need not be fixed size, and there can be moves with overlapping blocks (sharing same variables).

Convergence of Metropolis-within-Gibbs

Theorem 6.44. *Suppose that the Metropolis-within-Gibbs chain is p-irreducible and that starting from any $x \in \operatorname{supp}(p)$, there is a positive probability of accepting at least one move in each coordinate direction. Then, the strong law of large numbers holds (see Theorem 6.26).*

Proof. (*) Theorem 12 of [26] shows that the chain is Harris recurrent[12], and the SLLN is implied by [17, Theorem 17.0.1 (i)]. □

Remark 6.45 (*). Theorem 6.44 adds one natural (and practically non-restrictive) condition over the irreducibility condition of Theorem 6.26, which only avoids some pathological scenarios (like if $x \in \operatorname{supp}(p)$ but the conditionals are well-defined...).

Because all moves in the Gibbs sampler are accepted, we have:

Corollary 6.46. *Any p-irreducible Gibbs sampler satisfies the SLLN.*

We conclude with a simple sufficient condition which ensures p-irreducibility of Gibbs sampling.

Definition 6.47 (Positivity of p)**.** The distribution p satisfies the *positivity condition* if the marginal distributions $p_i(x)$ satisfy for all $x \in \mathbb{R}$

$$\operatorname{supp}(p) = \operatorname{supp}(p_1) \times \cdots \times \operatorname{supp}(p_d).$$

In other words, $p_i(x^{(i)}) > 0$ for all $i = 1, \ldots, d$ if and only if $p(x^{(1)}, \ldots, x^{(d)}) > 0$.

Proposition 6.48. *If p satisfies the positivity condition, then the conditional densities $p_{i|-i}$ are well-defined everywhere on the support of p and the Gibbs sampling Markov chain is p-irreducible.*

---

12. *From any initial point $x \in \operatorname{supp}(p)$, the chain will visit each set $A \subset \mathbb{X}$ such that $\int_A p(x)\mathrm{d}x > 0$ with probability one.

About BUGS (*)

The BUGS (Bayesian inference Using Gibbs Sampling) software [28] is an implementation of Gibbs sampling (and sometimes also other Metropolis-within-Gibbs updates). The user supplies only the model (using a specialised 'programming language') and the data, and the BUGS software outputs MCMC simulation of a given length.

The model is given in BUGS by specifying the joint distribution $\hat{p}$

$$\hat{p}(x^{(1:d)}) = p_1(x^{(1)}) \prod_{i=2}^{d} p_i(x^{(i)} \mid x^{(1:i-1)}),$$

where '$x^{(1:i)}$' is a shorthand for '$x^{(1)}, \dots, x^{(i)}$. This specifies $\hat{p}$ fully, and on the other hand, any d-dimensional distribution $\hat{p}$ can be factored like this.

Usually, the model is sparse, that is, $p_i(x^{(i)} \mid x^{(1:i-1)})$ do not depend on all $x^{(1:i-1)}$, but on a subset of 'parent' variables. This reflects conditional independencies, which define a directed acyclic graph. For instance, a Markov chain with initial distribution λ and transition probability P could be given as above, where $p_1 = \lambda$ and $p_i(x^{(i)} \mid x^{(1:i-1)}) = P(x^{(i-1)}, x^{(i)})$ for $i \geq 2$.

The distribution of interest p is a conditional distribution of $\hat{p}$, given some 'data'. For instance, if the first two variables $X^{(1)} = x_*^{(1)}$ and $X^{(2)} = x_*^{(2)}$ were observed, and the others not, then the MCMC targets the posterior distribution of $X^{(3:d)} \mid X_{(1:2)} = x_*^{(1:2)}$ which satisfies

$$p(x^{(3:d)}) \propto p_u(x^{(3:d)}) = \hat{p}(x_*^{(1)}, x_*^{(2)}, x^{(3:d)}).$$

This can be simulated with (Metropolis-within-)Gibbs that updates only the unobserved $x^{(3:d)}$, one at a time.

6.7 Langevin-type proposals (*)

One way to construct proposal distributions $q(x, y)$ in the Metropolis-Hastings algorithm is to use random-walk like proposals, but also use $\nabla \log p(x)$ to 'inform' the direction of proposals, based on the shape of p around x. The simplest such proposal is of the 'Langevin' type, where

$$Y_k = X_{k-1} + \frac{\tau}{2} \nabla \log p(X_{k-1}) + \sqrt{\tau} Z, \qquad Z \sim N(0, \Sigma), \tag{16}$$

for some parameters $\tau \in (0, \infty)$ and covariance $\Sigma \in \mathbb{R}^{d \times d}$.[13] This algorithm is known as the *Metropolis adjusted Langevin* algorithm (MALA).

MALA is just Metropolis-Hastings algorithm with proposal $q(x, y) = N(y; x + \frac{\tau}{2} \nabla \log p(x), \tau\Sigma)$ corresponding to (16). Note that in this case, $q(x, y) \neq q(y, x)$ and so the ratio $q(y, x)/q(x, y)$ does not vanish from the acceptance probability!

---

13. The proposal (16) stems from an Euler discretisation of the (overdamped) Langevin diffusion of the form $\mathrm{d}X_t = \frac{1}{2} \nabla \log p(X_t) \mathrm{d}t + \mathrm{d}B_t$, which is a continuous-time Markov process that admits p as its stationary distribution...

6.8 Hamiltonian Monte Carlo (*)

In recent years, a so-called *Hamiltonian Monte Carlo* (HMC) MCMC algorithm has gained attention [cf. 18]. Its proposal is based on a physics-motivated continuous-time process (*Hamiltonian dynamics*) involving an auxlliary *momentum* random vector.

The HMC is based on the target distribution $\tilde{p}(x, m) = p(x)q(m)$, where the auxiliary 'momentum' variable m has distribution q, a density of $N(0, \Sigma)$. The related 'Hamiltonian' can be written as

$$H(x, m) := -\log \tilde{p}(x, m) = U(x) + K(m),$$

where $U(x) := -\log p(x)$ and $K(m) := -\log q(m) = \frac{1}{2} m^T \Sigma^{-1} m$ (up to a constant). The proposal is *inspired by* the following system of differential equations:

$$\mathrm{d}m_t/\mathrm{d}t = -\nabla U(x_t) \qquad\qquad \mathrm{d}x_t/\mathrm{d}t = \Sigma^{-1} m_t. \tag{17}$$

These differential equations leave $\tilde{p}$ invariant (that is, if $(m_0, x_0) \sim \tilde{p}$, then also $(m_t, x_t) \sim \tilde{p}$ for any $t > 0$!), but of course we cannot solve them exactly. HMC uses a specific kind of numerical approximation of (17), (with $L \geq 1$ steps and with step size $\tau > 0$) in order to construct the proposals, and an acceptance ratio which ensures reversibility.

Algorithm 6.49 (Hamiltonian Monte Carlo)**.** Let $X_0 \equiv x_0$ s.t.$p(x_0) > 0$. For $k = 1, \ldots, n$:

(i) Draw $M_{k-1} \sim q$.
(ii) Calculate $(\hat{X}_k, \hat{M}_k) \leftarrow \mathrm{LF}(X_{k-1}, M_{k-1})$
(iii) Generate $U_k \sim \mathcal{U}(0, 1)$, and if $U_k \leq \alpha(X_{k-1}, M_{k-1}; \hat{X}_k, \hat{M}_k)$ *accept* and set $X_k = \hat{X}_k$, otherwise *reject* and set $X_k = X_{k-1}$, where the *acceptance probability* α is defined as follows:

$$\alpha(x, m; \hat{x}, \hat{m}) := \min\left\{1, \frac{\tilde{p}(\hat{x}, \hat{m})}{\tilde{p}(x, m)}\right\} = \min\left\{1, \exp\big(H(x, m) - H(\hat{x}, \hat{m})\big)\right\}.$$

where

LF(x_0, m_0):
For $t = 1, \ldots, L$:
(i) $\hat{m}_t \leftarrow m_{t-1} + \frac{\tau}{2} \nabla \log p(x_{t-1})$
(ii) $x_t \leftarrow x_{t-1} + \tau \Sigma^{-1} \hat{m}_t$
(iii) $m_t \leftarrow \hat{m}_t + \frac{\tau}{2} \nabla \log p(x_t)$
Return $(x_L, -m_L)$

(NB: The *momentum flip* in the end of LF($\cdot$) is unnecessary in practice, but included here for mathematical convenience...)

The HMC algorithm looks similar to Metropolis-Hastings (and indeed may be seen as an instance of a generalisation of Metropolis-Hastings).

The key observations required to check p-reversibility of the HMC are:

1. If $X_{k-1} \sim p$, then $(X_{k-1}, M_{k-1}) \sim \tilde{p}$.

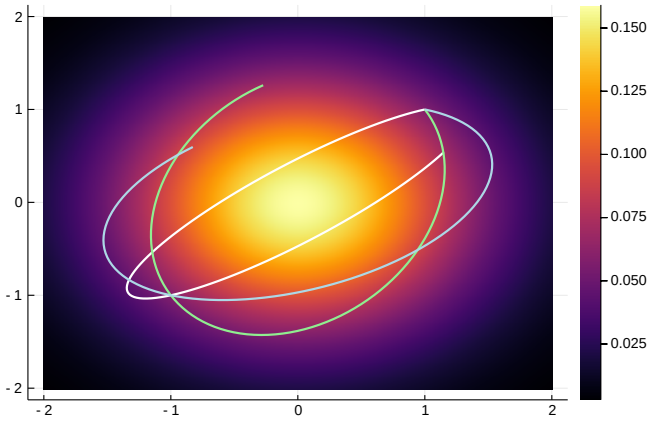


Figure 16: Three trajectories $(x_0, \ldots, x_L)$ of the leapfrog integrator starting from $x_0 = [1, 1]^T$ with three independent realisations of m_0 from $N(0, I_2)$. Here, $p = N(0, I_2)$ with density values shown as background color, $L = 100$ and $\tau = 0.05$.

2. The leapfrog integrator $\mathrm{LF}(\,\cdot\,)$ is *reversible*, in the sense that if $(\hat{x}, \hat{m}) = \mathrm{LF}(x, m)$, then $(x, m) = \mathrm{LF}(\tilde{x}, \tilde{m})$. (Or, equivalently, it is an involution: $\mathrm{LF}(\mathrm{LF}(x, m)) = (x, m)$.)
3. The leapfrog integrator $LF(\,\cdot\,)$ is isometric, or volume-preserving.

See [8] for details, as well as result showing the p-irreducibility of the HMC (which turns out to be a non-trivial exercise!).

There are a number of user-friendly implementations of (variants of) HMC. Stan [5] is the most popular, and has an interface similar to BUGS, allowing to build model from blocks. Stan can provide good performance in some scenarios where BUGS struggles, but it does not always outperform BUGS. If you intend to use Stan, there are certain inherent restrictions that come with it, which are good to know:

- Discrete variables cannot be unknowns.
- Unknowns need to be (easily transformable) to $\mathbb{R}$ (or $\mathbb{R}^d$).[14]
- Tail behaviour and geometry of p may have a dramatic influence in performance.
- The variables need to be (roughly) unit-scaled.

Even though the HMC (and its implementation in Stan) have showed great promise in many practical situations, they may not always provide a reliable outcome, and this may not be easy to predict.

This is in contrast with Gibbs sampling and random-walk proposals, which are rather well understood by now (including their weaknesses!).

14. Stan transforms $x > 0$ and $x \in (a, b)$ automatically with exponential and logistic transformations.

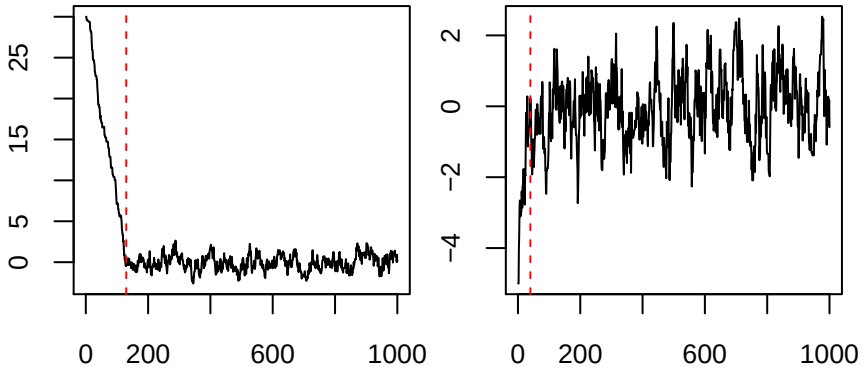


Figure 17: The first 1000 samples simulated from Example 6.27 with $a = 1$ and with with $x_0 = 30$ (left) and $x_0 = -5$ (right). The red vertical line indicates the 'burn-in time'.

7 MCMC convergence and mixing

With MCMC, there are two issues considering the reliability of the calculated averages: $I^{(n)}_{p,q,\mathrm{MH}}(f) = n^{-1}\sum_{k=1}^{n} f(X_k)$:

- The MCMC chain does not start from the invariant/stationary distribution, so $\mathbb{E}[f(X_k)] \neq \mathbb{E}_p[f(X)]$, and the difference may well be substantial for small k. This can induce significant *bias* to the estimator.
- It is not direct to assess the reliability of MCMC averages, because of the dependence of the random variables (X_k). The dependence usually adds *variance* to the estimator, when compared against simple Monte Carlo averages.

7.1 Burn-in bias

MCMC Markov chain X_n converges in distribution to p as $n \to \infty$ (under an aperiodicity condition, cf. Theorem 6.7). The common practice with MCMC is to discard b first values of the Markov chain $X_0, \ldots, X_b$, to minimise bias. It is assumed that X_{b+1} will have approximately the distribution p, and then use the estimator

$$\frac{1}{n-b}\sum_{k=b+1}^{n} f(X_k).$$

The initial period $X_0, \ldots, X_b$ is called *burn-in* of the MCMC.

Remark 7.1. Several statistics may be calculated in order to 'detect' a bias in MCMC. However, they usually rely on certain rather strong assumptions, such as the asymptotic normality, or at least unimodality of the target.

7.2 Asymptotic variance of MCMC

With classical Monte Carlo and importance sampling, the confidence intervals can be constructed with help of the CLT, and the associated variance is relatively straightforward to calculate.

Also Markov chains satisfy CLT in many cases. For example, we may record the following statement without proof.

Theorem 7.2. *If the Metropolis-Hastings Markov chain (X_k) on finite $\mathbb{X}$ is irreducible and aperiodic, then*

$$\sqrt{n}\big[I^{(n)}_{p,q,\mathrm{MH}}(f) - \mathbb{E}_p[f(X)]\big] \xrightarrow{n\to\infty} N(0, \sigma^2_{\mathrm{MH}}), \tag{18}$$

with $\sigma^2_{\mathrm{MH}} = \lim_{n\to\infty} n\mathrm{Var}\big(I^{(n)}_{p,\mathrm{MH}}(f)\big) < \infty$.

Remark 7.3. The CLT (18) holds quite generally, *under certain technical regularity conditions*. Because there are no general and easily verifiable conditions available, we shall not detail a more general form of the CLT, but assume it to hold.

We shall look next at an expression of the CLT variance (when finite), which gives a method to estimate the CLT variance.

Theorem 7.4. *Let $X_0, X_1, \ldots$ be a stationary Markov chain, that is, $X_0 \sim p$, where p is the invariant distribution. Suppose $f : \mathbb{S} \to \mathbb{R}$ such that $\mathbb{E}_p[f^2(X)] < \infty$ and denote $Y_k = f(X_k)$.*

Assuming $\sum_{k=1}^{\infty} \rho_k < \infty$ where $\rho_k := \mathrm{Corr}\big(Y_0, Y_k\big)$, we have

$$\lim_{n\to\infty} n\mathrm{Var}\bigg(\frac{1}{n}\sum_{k=1}^{n} f(X_k)\bigg) = \mathrm{Var}_p(f(X))\bigg(1 + 2\sum_{k=1}^{\infty}\rho_k\bigg).$$

Remark 7.5. With MCMC, X_0 is of course never exactly a sample of p, but as discussed earlier, X_b can be regarded to have approximately the distribution p whenever b is large. Therefore, if we apply Theorem 7.4 to $\tilde{X}_n := X_{b+n}$ for $n \geq 0$, the result is still relevant. (Rigorous extension to arbitrary initial measure is possible, but we shall not consider it here.)

Remark 7.6. Theorem 7.4 holds more generally, for any (weak-sense) stationary process $(Y_k)_{k\geq 1}$.

Proof of Theorem 7.4. Let us define $Y_k = f(X_k)$, and $\bar{Y}_k = Y_k - \mathbb{E}[Y_k]$, then

$$\begin{aligned}
\mathrm{Var}\bigg(\frac{1}{n}\sum_{k=1}^{n} f(X_k)\bigg) &= \frac{1}{n^2}\mathbb{E}\bigg[\bigg(\sum_{k=1}^{n}\bar{Y}_k\bigg)^2\bigg] \\
&= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{E}[\bar{Y}_i\bar{Y}_j] \\
&= \frac{\mathrm{Var}_p\big(f(X)\big)}{n} + \frac{2}{n^2}\sum_{i=1}^{n}\sum_{j=i+1}^{n}\mathrm{Cov}(Y_i, Y_j) \\
&= \frac{\mathrm{Var}_p\big(f(X)\big)}{n}\bigg(1 + \frac{2}{n}\sum_{h=1}^{n-1}(n-h)\rho_h\bigg).
\end{aligned}$$

Multiply with n and take limits, and apply Lemma 7.7 to show that $n^{-1}\sum_{h=1}^{n-1} h\rho_h \xrightarrow{n\to\infty} 0$. □

Lemma 7.7 (Kronecker)**.** *Suppose* $(x_k)_{k\geq 1}$ *is a sequence of real numbers with* $\sum_{k=1}^{\infty} x_k = s \in \mathbb{R}$*. Then,* $n^{-1}\sum_{k=1}^{n} kx_k \xrightarrow{n\to\infty} 0$*.*

Definition 7.8. The *integrated autocorrelation time* of (Y_i) and the *effective sample size* of $(Y_1, \ldots, Y_n)$ are defined, respectively, as

$$\mathrm{IACT} := 1 + 2\sum_{i=1}^{\infty} \rho_i \qquad \text{and} \qquad n_{\mathrm{eff}} := \frac{n}{\mathrm{IACT}}.$$

The definitions of 'effective sample size' makes sense when we use Theorem 7.4 to deduce that for n large enough

$$\mathrm{Var}(I^{(n)}_{p,q,\mathrm{MH}}) \approx \frac{\mathrm{IACT}}{n}\mathrm{Var}_p\big(f(X)\big) = \frac{1}{n_{\mathrm{eff}}}\mathrm{Var}_p\big(f(X)\big).$$

Suppose then $(Z_1, \ldots, Z_{\lfloor n_{\mathrm{eff}}\rfloor})$ are independent from p, the classical Monte Carlo satisfies

$$\mathrm{Var}\Big(\frac{1}{\lfloor n_{\mathrm{eff}}\rfloor}\sum_{k=1}^{\lfloor n_{\mathrm{eff}}\rfloor} f(Z_k)\Big) = \frac{1}{\lfloor n_{\mathrm{eff}}\rfloor}\mathrm{Var}_p\big(f(X)\big).$$

So, the mean estimator based on the sample $X_1, \ldots, X_n$ from MCMC is (asymptotically) as efficient as the one based on $Z_1, \ldots, Z_{\lfloor n_{\mathrm{eff}}\rfloor} \overset{\text{i.i.d.}}{\sim} p$.

Remark 7.9 (*). Simple (and traditional) way to estimate IACT (and equivalently the asymptotic variance or n_{eff}) is to sum sample autocorrelations up to a truncation point, which is chosen based on an inspection of the sample autocorrelations. However, there are also reasonably straightforward and provably consistent estimators of the asymptotic variance [9].

Remark 7.10. Note that a MCMC sample $(X_k)_{k=1,\ldots,n}$ does *not* have a single effective sample size n_{eff}, but n_{eff} depends on the function. So if you are interested in different functions $f_1, \ldots, f_m : \mathbb{X} \to \mathbb{R}$, you need to calculate $n^{(1)}_{\mathrm{eff}}, \ldots, n^{(m)}_{\mathrm{eff}}$! This is particularly important if $\mathbb{X} = \mathbb{R}^d$, and $f_i(x) = x_i$, in which case the effective sample size of different coordinates may differ significantly.

7.3 Practical summary

When using MCMC, always do the following checks:

(i) Plot MCMC traces of the variables and key functions of the variables. They should look stationary after burn-in.

(ii) Make multiple MCMC runs from different initial state x_0 and check that the marginal distributions (or the estimators) look similar.
This test reveals if your chain is 'almost reducible'.

(iii) Plot sample autocorrelations of the variables and functions (e.g. `autocor` of `StatsBase`).

(iv) Calculate n_{eff} and check that it is reasonably large. Use it to construct a CI:

$$\Big[I_{p,q,\mathrm{MH}}(f) \pm \beta\frac{\hat{\sigma}_n}{\sqrt{n_{\mathrm{eff}}}}\Big],$$

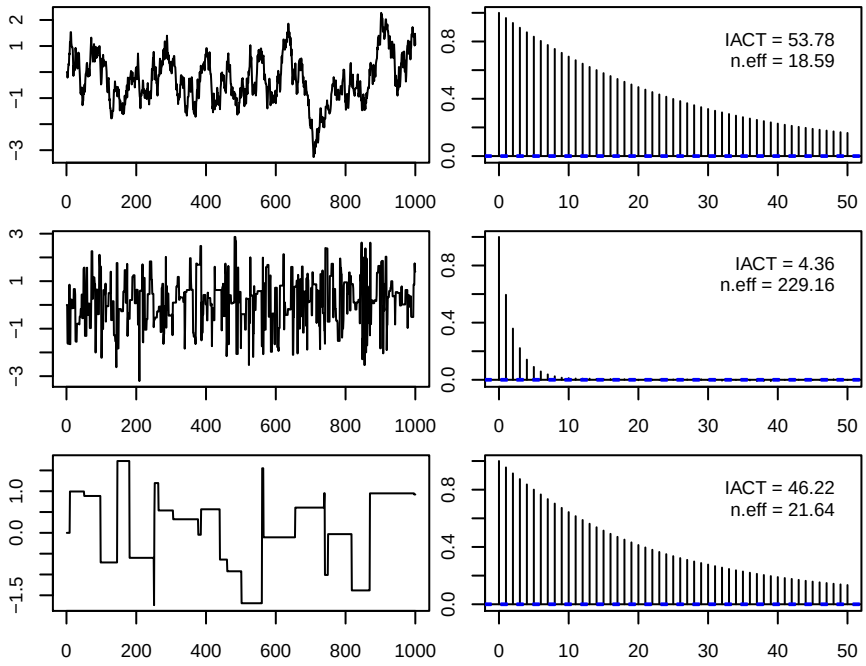


Figure 18: Sample paths and correlations of MH in Example 6.27 with $a = 0.5$ (top), $a = 5$ (middle) and $a = 50$ (bottom); here $f(x) := x$.

where $\hat{\sigma}_n^2 := (n-1)^{-1} \sum_{k=1}^n \big[f(X_k) - I_{p,q,\mathrm{MH}}(f)\big]^2 \xrightarrow{n\to\infty} \mathrm{Var}_p\big(f(X)\big)$ and β is the desired Normal quantile; cf. Proposition 1.13.

Remember to discard the burn-in samples before proceeding to (iii) and (iv). Remember also that both ACF and n_{eff} depend on the function!

7.4 Optimising MCMC (*)

Usually asymptotic variance cannot be calculated in a closed form, but comparison of asymptotic variances may be possible.

Theorem 7.11 (Peskun [23], Tierney [30])**.** *Suppose that P and Q are transition probabilities both reversible wrt. common distribution π. Suppose that*

$$\sum_{x,y\in\mathbb{X}} \pi(x)P(x,y)[f(x)-f(y)]^2 \geq \sum_{x,y\in\mathbb{X}} \pi(x)Q(x,y)[f(x)-f(y)]^2, \qquad (19)$$

for all $f : \mathbb{S} \to \mathbb{R}$ with $\mathbb{E}_\pi[f^2(X)] < \infty$. Then, P is always better than Q in the following sense: for any function $f : \mathbb{S} \to \mathbb{R}$ with $\mathbb{E}_\pi[f^2(X)] < \infty$,

$$\lim_{n\to\infty} n\mathrm{Var}\left(\frac{1}{n}\sum_{k=1}^n f(X_k^{(P)})\right) \leq \lim_{n\to\infty} n\mathrm{Var}\left(\frac{1}{n}\sum_{k=1}^n f(X_k^{(Q)})\right),$$

where $(X_k^{(P)})_{k\geq 0}$ and $(X_k^{(Q)})_{k\geq 0}$ are stationary Markov chains with transition probabilities P and Q, respectively.

Remark 7.12. It is easy to see that

$$P(x, y) \geq Q(x, y) \qquad \text{for all } x \neq y, \tag{20}$$

implies (19). The condition (20) is referred to as the *off-diagonal order* or the *Peskun order* and (19) is known as the *covariance order*.

Remark 7.13. In the continuous case, if P and Q are in the form (14) with $k_P(x, y)$ and $k_Q(x, y)$, respectively, then the covariance order (19) corresponds to

$$\iint \pi(x)k_P(x, y)[f(x) - f(y)]^2 \mathrm{d}x\mathrm{d}y \geq \iint \pi(x)k_Q(x, y)[f(x) - f(y)]^2 \mathrm{d}x\mathrm{d}y,$$

which holds if the analogous off-diagonal order holds:

$$k_P(x, y) \geq k_Q(x, y) \qquad \text{for all } x \neq y.$$

The covariance order is equivalent with order $\mathcal{E}_P(f) \geq \mathcal{E}_Q(f)$ of Dirichet forms

$$\mathcal{E}_P(f) := \langle f, (I - P)f \rangle_\pi, \qquad \langle f, g \rangle_\pi := \int \pi(x)f(x)g(x)\mathrm{d}x,$$

where I is identity operator so $(If)(x) = f(x)$ and $(Pf)(x) = \int P(x, \mathrm{d}y)f(y)\mathrm{d}y$.

Example 7.14. In the Ising model Example 6.39, we have a choice of the proposal distribution $q_i(x, y \mid x^{(-i)})$. Note that here $x, y \in \{0, 1\}$. The best choice in terms of asymptotic variance is to take $q_i(x, y \mid x^{(-i)}) = \mathbf{1}(y = 1 - x)$, because any other choice would be worse in terms of the off-diagonal order (20).

Example 7.15 (Barker's algorithm). In the Metropolis-Hastings algorithm, we could use an alternative acceptance probability

$$\alpha_B(x, y) := \frac{r(x, y)}{r(x, y) + 1}, \qquad r(x, y) := \frac{p(y)q(y, x)}{p(x)q(x, y)}.$$

Similarly as with Metropolis-Hastings, it is direct to check that

$$p(x)q(x, y)\alpha_B(x, y) = p(y)q(y, x)\alpha_B(y, x),$$

so the resulting algorithm is still reversible wrt. p.

Direct calculation shows that $\alpha_B(x, y) \leq \alpha(x, y) = \min\{1, r(x, y)\}$, which implies an off-diagonal order, so the Barker's algorithm using α_B acceptance rate is never better than Metropolis-Hastings. (There are certain situations where α_B is easier to calculate, though.)

8 Sequential Monte Carlo

We shall focus next on algorithms which operate on a *sequence* of distributions $\pi_1, \pi_2, \ldots, \pi_T$, which gradually evolve towards the distribution of interest $p = \pi_T$. The samples are often called *particles* in this context, and the key algorithm in this context is known as the *particle filter*.

We will motivate the algorithms in a time-series context, which was their original motivation, and where they have been applied extensively. We present the methods with densities on an Euclidean space; discrete case follows similarly.

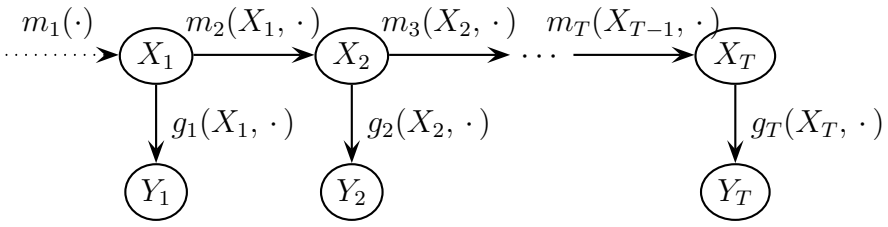


Figure 19: General state-space model.

In this section, we denote for $a \leq b$ the vector $x_{a:b} = (x_a, \ldots, x_b)$. We also exceptionally denote *'time' indices* in subscript (not Monte Carlo samples as before), and superscript contain *sample indices* (not coordinates as before).

8.1 Motivation: General state-space models/hidden Markov models

Figure 19 illustrates a general state-space model. It consists of two parts:

- 'Latent' Markov chain $(X_t)_{t\geq 1}$ evolving in $\mathbb{S} = \mathbb{R}^d$ with initial density $X_1 \sim m_1$, and with conditional densities $m_t(x_{t-1}, x_t)$ of $X_t \mid (X_t = x_t)$. (Note that the transition densities may depend on time t.)
- Conditionally independent observed process $(Y_t)_{t\geq 1}$ following the observation densities $Y_t \mid X_t \sim g_t(X_t, \cdot)$.

More precisely, the model defines the joint density of the form $\hat{p}(x_{1:T}, y_{1:T}) := m_1(x_1)g_1(x_1, y_1)\prod_{t=2}^{T} m_t(x_{t-1}, x_t)g_t(x_t, y_t)$.

We are interested in Bayesian inference of $X_{1:T}$ having observed $Y_{1:T} = y_{1:T}$, that is, we focus on the conditional density p of $\hat{p}$:

$$p(x_{1:T}) \propto p_u(x_{1:T}) := m_1(x_1)g_1(x_1, y_1)\prod_{t=2}^{T} m_t(x_{t-1}, x_t)g_t(x_t, y_t), \tag{21}$$

where $y_{1:T}$ are the observed values, which are constant in our case, and omitted from the notation.

Remark 8.1. What we call state-space models (SSM), some other authors call *hidden Markov models* (HMM) [e.g. 4, 12]. Some authors reserve HMM to mean the case where X_k are discrete, taking values on a finite set. Some others reserve SSM to mean only linear(-Gaussian) models.

Example 8.2 (Noisy AR(1) process). Let $\sigma_1^2, \sigma_x^2, \sigma_y^2 \in (0, \infty)$ and $\rho \in \mathbb{R}$ be known parameters. Then, let $m_1 = N(0, \sigma_1^2)$ and for $k \geq 2$, assume $(Z_k)_{k\geq 1}, (W_k)_{k\geq 1} \overset{\text{i.i.d.}}{\sim} N(0, 1)$, and define

$$X_k := \rho X_{k-1} + \sigma_x Z_k$$
$$Y_k := X_k + \sigma_y W_k.$$

This corresponds to setting

$$m_k(x_{k-1}, x_k) := N(x_k; \rho x_{k-1}, \sigma_x^2)$$
$$g_k(x_k, y_k) := N(y_k; x_k, \sigma_y^2).$$

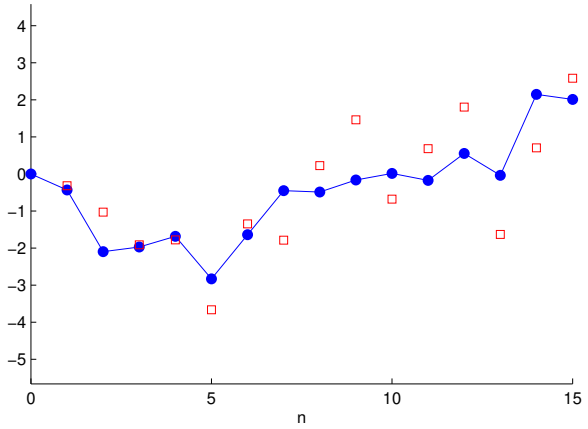


Figure 20: Sample path of the noisy AR(1) process in Example 8.2 with $\rho = 1$ and $\sigma_1^2 = \sigma_x^2 = 1 = \sigma_y^2$: The Markov chain $X_{1:15}$ in blue and the noisy observations $Y_{1:15}$ in red.

In other words, $(X_k)_{k\geq 1}$ is an AR(1) process.[15] Given a realisation of the process $(X_1, \ldots, X_T)$, the observations are conditionally independent and perturbed by Gaussian increments with variance σ_y^2. Figure 20 shows an example realisation of the process.

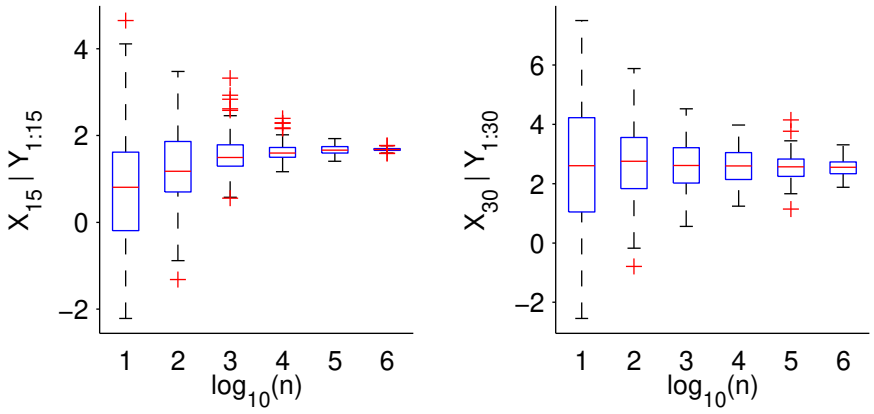
Remark 8.3. The generic methods such as importance sampling and MCMC (Random-walk Metropolis, Metrpolis-within-Gibbs, Hamiltonian Monte Carlo...) are, in theory, directly applicable in the SSM context. However, when T is large, the space $\mathbb{S}^T$ is high-dimensional, and there are substantial correlations in the model, which often lead to poor performance...

Remark 8.4 (*). Exact SSM inference (i.e. when the conditional distribution is available in a closed form) is possible only in some specific cases, most notably [e.g. 4]:

- When $\mathbb{S}$ is finite, exeact inference is possible through the *forward-backward* algorithm.
- If $\mathbb{S} = \mathbb{R}^d$ and and the conditional distributions m_t and g_t are linear Gaussian, that is, $g_t(x_t, \cdot)$ is a Gaussian density with mean $L_t x_t$ and some covariance matrix R_t, and similarly for m_t, then, the smoothing density (and consequently all the marginals) are Gaussian. Then, the mean & covariance parameters can be computed by simple matrix formulae (the Kalman filter and smoother).

In most other cases, inference need to be based on an approximation, such as SMC.

15. Stationary iff $|\rho| < 1$ and $\sigma_1^2 = \frac{\sigma_x^2}{1-\rho^2}$.



(a) (b)

Figure 21: Box plot of estimates from Example 8.6 with up to one million samples, and 100 repetitions. (a) $T = 15$ (true value: 1.685) (b) $T = 30$ (true value: 2.508).

8.2 First attempt: Sequential importance sampling

Let us see what happens if we apply self-normalised importance sampling in the context of SSMs.

Generic self-normalised importance sampling is straightforward to apply here, because assuming $q(x_{1:T})$ is a proposal density on $\mathbb{S}^T$, with support covering that of $p(x_{1:T})$, we could just draw $X_{1:T}^{(k)} \overset{\text{i.i.d.}}{\sim} q$ and approximate

$$\mathbb{E}_p[f(X_{1:T})] \approx \frac{\sum_{k=1}^n w_u(X_{1:T}^{(k)}) f(X_{1:T}^{(k)})}{\sum_{j=1}^n w_u(X_{1:T}^{(j)})}, \qquad \text{where} \qquad w_u(x_{1:T}) := \frac{p_u(x_{1:T})}{q(x_{1:T})}.$$

Remark 8.5. Note that also the proposal q may depend on the observations $y_{1:T}$, in an arbitrary manner. Recall also that the notation differs here from the notation in Section 4.3: we write the sample index in superscript.

Example 8.6 (Noisy AR(1) with prior as q). Consider Example 8.2 and let q be the prior of $X_{1:T}$, that is,

$$q(x_{1:T}) = m_1(x_1) \prod_{t=2}^{T} m_t(x_{t-1}, x_t).$$

This means that we simulate $X_{1:T}$ to be the trajectories of T steps of a random walk with independent Gaussian $N(0, 1)$ increments.

Figures 21 and 22 show simulation results of Example 8.6.

The problem with Example 8.6 is that, even if the weights are bounded, the discrepancy of p and q increases very rapidly as T increases. In intuitive terms, most samples from q fall into low density area of p, and consequently the variance of the weights is large.

Let us have another attempt with more carefully chosen q:

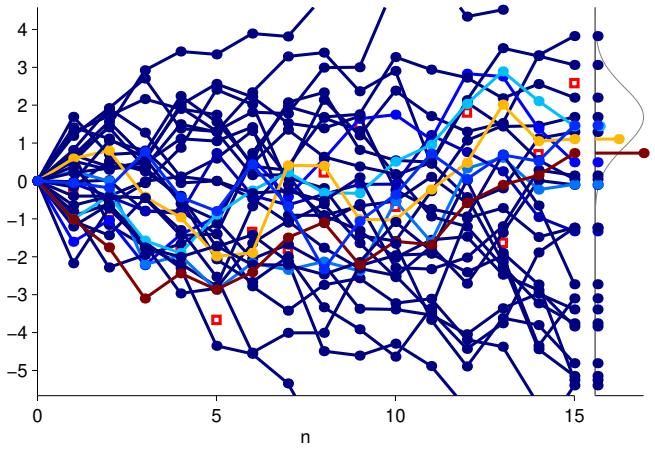
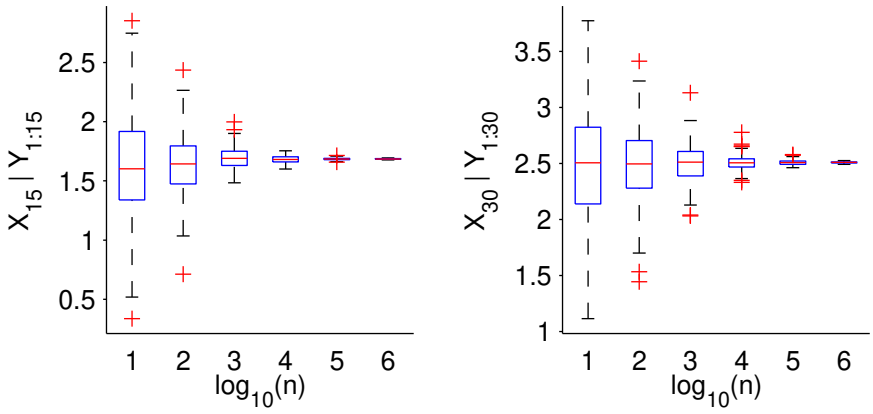


Figure 22: Some samples corresponding Example 8.6. Note that the weight distribution is very unequal. The true posterior density is shown on the right.



(a) (b)

Figure 23: Box plot of estimates from Example 8.7; compare with 21.

Example 8.7 (Noisy AR(1) with a 'one-step optimal' q). Consider Example 8.6, but choose now

$$q(x_{1:T}) = q_1(x_1)\prod_{t=2}^{T} q_t(x_t \mid x_{t-1}), \qquad q_t(x_t \mid x_{t-1}) = N\left(x_t; \frac{x_{t-1}+y_t}{2}, \frac{1}{2}\right) \qquad (\text{with } x_0 \equiv 0).$$

In fact, this choice of q_t corresponds to the conditional distribution of X_t given $X_{t-1} = x_{t-1}$ and $Y_t = y_t$. The conditional distribution is, in a certain sense, the best choice we can have (if we restrict on q_t that can only depend on $y_{1:t}\ldots$). It is direct to check that the unnormalised weights $w_u(z_{1:T})$ resulting from this choice are also bounded (exercise).

Figures 23 and 24 show simulation results corresponding Example 8.7.

Using a better proposal distribution in Example 8.7 improved significantly. It made reliable inference possible for up to $T = 30$ with around one million samples. This is achieved by better approximation of p by q, which shows in

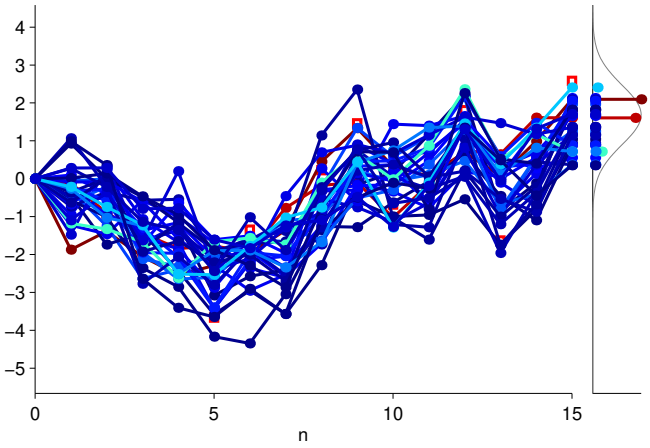
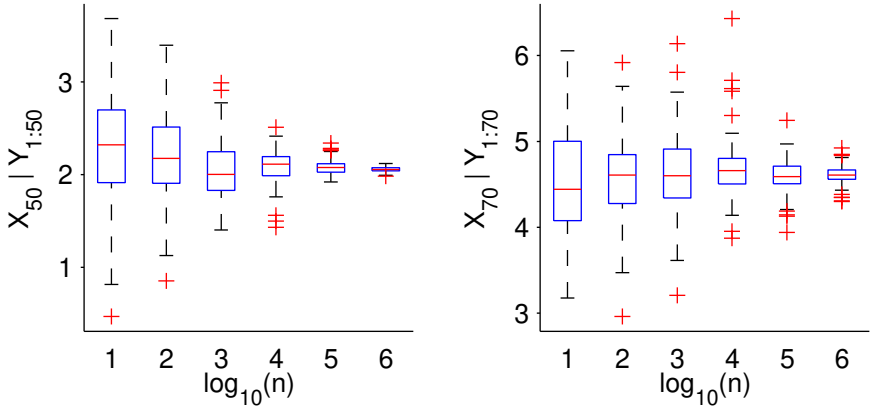


Figure 24: Some samples corresponding Example 8.7; compare with Figure 22.



(a) (b)

Figure 25: Box plot of estimates from Example 8.7 with $T = 50$ (true: 2.058) and $T = 70$ (true: 4.606).

Figure 24 by concentration of the samples around the measured values.

However, if we increase T a bit more, we see that even the very good proposal distribution in 8.7 is insufficient for efficient inference; see Figure 25. In fact, the variance typically increases exponentially in T (cf. [4, Example 7.3.1]).

The particle filter algorithm, which we discuss next, is a simple algorithmic modification of the SIS, which resolves the 'mismatch' by further randomisation...

8.3 Generic form of sequential importance sampling

Suppose now that $M_t(x_t \mid x_{1:t-1})$ for $t = 2, \dots, T$ determines a distribution on $\mathbb{S}$ for x_t for any $x_{1:t-1} \in \mathbb{S}^{t-1}$, and that $G_t(x_{1:t}) \geq 0$ are some 'potential' functions, for which:

$$\boxed{M_1(x_1)G_1(x_1)\prod_{t=2}^{T} M_t(x_t \mid x_{1:t-1})G_t(x_{1:t}) \equiv p_u(x_{1:T}).} \tag{22}$$

Remark 8.8. Note that in the SSM context, (22) is equivalent with $q(x_{1:T}) = M_1(x_1)\prod_{t=2}^T M_t(x_t \mid x_{1:t-1})$ satisfying the SNIS support condition (10) and G_t forming a *factorisation* of the unnormalised importance weight:

$$\prod_{t=1}^T G_t(x_{1:t}) = w_u(x_{1:T}) = \frac{m_1(x_1)g_1(x_1,y_1)\prod_{t=2}^T m_t(x_{t-1},x_t)g_t(x_t,y_t)}{M_1(x_1)\prod_{t=2}^T M_t(x_t \mid x_{1:t-1})}, \qquad \text{when } q(x_{1:T}) > 0.$$

We may choose $G_t(x_{1:t}) = \frac{m_t(x_{t-1},x_t)g_t(x_t,y_t)}{M_t(x_t|x_{1:t-1})}$, which satisfies (22), but other choices are possible.

Remark 8.9 (*). The model with ingredients of the form $M_{1:T}$ and $G_{1:T}$ is known as the *Feynman-Kac* model [7].

Algorithm 8.10 (Sequential importance sampling)**.** In each line of the algorithm, $i = 1, \ldots, n$:

(i) Sample $X_1^{(i)} \sim M_1(\cdot)$ and set $\mathbf{X}_1^{(i)} = X_1^{(i)}$.

(ii) Calculate $\omega_1^{(i)} := G_1(\mathbf{X}_1^{(i)})$.

For $t = 2, \ldots, T$, do:

(iii) Sample $X_t^{(i)} \sim M_t(\cdot \mid \mathbf{X}_{t-1}^{(i)})$ and set $\mathbf{X}_t^{(i)} = (\mathbf{X}_{t-1}^{(i)}, X_t^{(i)})$.

(iv) Calculate $\omega_t^{(i)} := G_t(\mathbf{X}_t^{(i)})$.

Report unnormalised sample $(V^{(1:n)}, \mathbf{X}^{(1:n)})$ where $V^{(j)} := \prod_{t=1}^T \omega_t^{(j)}$ and $\mathbf{X}^{(j)} := \mathbf{X}_T^{(j)}$.

Proposition 8.11. *Let $t \in \{1{:}T\}$ such that $\int M_1(x_1)G_1(x_1)\prod_{k=2}^t M_k(x_k \mid x_{1:k-1})G_k(x_{1:k})\mathrm{d}x_{1:t} < \infty$. Consider Algorithm 8.10, and and let $\pi_t(x_{1:t}) \propto M_1(x_1)G_1(x_1)\prod_{k=2}^t M_k(x_k \mid x_{1:k-1})G_k(x'_{1:k})$ be a probability density. Then, denoting $V_t^{(i)} := \prod_{k=1}^t \omega_k^{(i)}$,*

$$\frac{\sum_{i=1}^n V_t^{(i)} f(\mathbf{X}_t^{(i)})}{\sum_{j=1}^n V_t^{(j)}} \xrightarrow{n\to\infty} \mathbb{E}_{\pi_t}[f(X_{1:t})] \qquad \textit{(in distribution),}$$

whenever the expectation is well-defined and finite.

Proof. This is self-normalised IS, because $\mathbf{X}_t \sim q_t(x_{1:t}) = M_1(x_1)\prod_{k=2}^t M_k(x_k \mid x_{1:k-1})$ and $V_t^{(i)} \propto \pi_t(\mathbf{X}_t)/q_t(\mathbf{X}_t)$. The result follows from Theorem 4.19. □

Corollary 8.12. *If assumption* (22) *holds, then the output of Algorithm 8.10 satisfies:*

$$\mathrm{SIS}^{(n)}_{M_{1:T},G_{1:T}}(f) := \frac{\sum_{k=1}^n V^{(k)} f(\mathbf{X}^{(k)})}{\sum_{j=1}^n V^{(j)}} \xrightarrow{n\to\infty} \mathbb{E}_p[f(X_{1:T})] \qquad \textit{(in distribution)}$$

Proof. Direct application of Proposition 8.11, because $p = \pi_T$ and $V^{(i)} = V_T^{(i)}$. □

Remark 8.13. When $\pi_1, \ldots, \pi_T = p$ are all well-defined, Proposition 8.11 indicates that Algorithm 8.10 may be regarded as approximating these distributions sequentially, by re-using the approximation for π_{t-1} when building the approximation for π_t.

Remark 8.14 (*). In self-normalised IS, we have *almost sure* convergence instead of *in distribution*. We state the results here using the latter, because we regard now Algorithm 8.15 to be run with a fixed n — and therefore 'adding samples' does not make immediate sense, but the algorithm may just be repeated with a higher n...

8.4 The particle filter

Algorithm 8.15 (Particle filter)**.** In each line of the algorithm, $i = 1, \dots, n$:

(i) Sample $X_1^{(i)} \sim M_1$ and set $\mathbf{X}_1^{(i)} = X_1^{(i)}$.

(ii) Calculate $\omega_1^{(i)} := G_1(\mathbf{X}_1^{(i)})$ and set $\bar{\omega}_1^{(i)} := \omega_1^{(i)}/\omega_1^*$ where $\omega_1^* = \sum_{j=1}^n \omega_1^{(j)}$.

For $t = 2, \dots, T$, do:

(iii) Sample $A_{t-1}^{(i)} \sim \text{Categorical}(\bar{\omega}_{t-1}^{(1:N)})$, that is, $\mathbb{P}(A_{t-1}^{(i)} = j) = \bar{\omega}_{t-1}^{(j)}$.

(iv) Sample $X_t^{(i)} \sim M_t(\cdot \mid \mathbf{X}_{t-1}^{(A_{t-1}^{(i)})})$ and set $\mathbf{X}_t^{(i)} = (\mathbf{X}_{t-1}^{(A_{t-1}^{(i)})}, X_t^{(i)})$.

(v) Calculate $\omega_t^{(i)} := G_t(\mathbf{X}_t^{(i)})$ and set $\bar{\omega}_t^{(i)} := \omega_t^{(i)}/\omega_t^*$ where $\omega_t^* = \sum_{j=1}^n \omega_t^{(j)}$.

Report $(V^{(1:n)}, \mathbf{X}^{(1:n)})$ where $V^{(j)} := \left(\prod_{t=1}^T \frac{1}{n}\omega_t^*\right)\bar{\omega}_T^{(j)}$ and $\mathbf{X}^{(j)} := \mathbf{X}_T^{(j)}$.
(In case $\omega_t^* = 0$, the algorithm is terminated with $V^{(i)} = 0$ and with arbitrary $\mathbf{X}^{(i)} \in \mathsf{S}^T$.)

Remark 8.16. The proposal $M_t(x_t \mid x_{1:t-1})$ and the potential $G_t(x_{1:t})$ typically depend on x_t and perhaps x_{t-1}, but not $x_{1:t-2}$. In such a case, it is not necessary to explicitly store $\mathbf{X}_t^{(i)}$, because $\omega_t^{(i)} = G_t(X_{t-1}^{(A_{t-1}^{(i)})}, X_t)$ and $\mathbf{X}^{(i)} = \mathbf{X}_T^{(i)}$ may be recovered from $X_{1:T}^{(j)}$ and $A_{1:T-1}^{(j)}$.

Example 8.17. Implementation with $M_t(x_t \mid x_{1:t-1}) = M_t(x_t \mid x_{t-1})$ and $G_t(x_{1:t}) = G_t(x_t)$:

```
function norm_logw(logw) # Normalised probabilities from log weights ('log-sum-trick')
  m = maximum(logw); u = exp.(logw.-m); return m+log(mean(u)), u/sum(u)
end
function pf(M, logG, n, T)  # Univariate particle filter
    X = zeros(n, T); A = zeros(Int, n, T); wu = zeros(n)
    for i = 1:n
        X[i,1] = x = M(1); wu[i] = logG(1, x)
    end
    V, omega = norm_logw(wu);
    for t = 2:T
        a = rand(Categorical(omega), n); A[:,t-1] = a
        for i = 1:n
            X[i,t] = x = M(t, X[a[i],t-1]); wu[i] = logG(t, x)
        end
        V_, omega = norm_logw(wu); V += V_
  end
  XT = zeros(n,T); XT[:,T]=X[:,T]; a = collect(1:n) # Trace back X^{(i)}
  for t = T-1:-1:1  a = A[a,t]; XT[:,t] = X[a,t]  end
  (logV=V.+log.(omega), XT=XT, X=X)
end
```

Application in Example 8.6, with an estimate for $\mathbb{E}_p[X]$:

```
using Distributions, Random, Plots
Random.seed!(42); T=50; x0=0; rho=sigma_x=sigma_y=1
function M(t, x=0.0) # Generate observations from M_t(.|x)
    rand(Normal(x, sigma_x))
end
x_true = zeros(T); x_true[1] = M(1)           # Generate synthetic data:
for t = 2:T x_true[t] = M(t, x_true[t-1]) end # trajectory of x_{1:T}
y = x_true + rand(Normal(0, sigma_y), T)      # and corresponding observations
function logG(t, x) # Calculate log G_t(x)
    logpdf(Normal(y[t], sigma_y), x)
end
o = pf(M, logG, 100, T)
scatter(o.X', color=:black); plot!(o.XT', width=2, legend=false)
sum(norm_logw(o.logV)[2] .* o.XT[:,T])
```

Under certain technical assumptions [cf. 7]:

$$\mathrm{PF}^{(n)}_{M_{1:T},G_{1:T}}(f) := \frac{\sum_{k=1}^n V^{(k)} f(\mathbf{X}^{(k)})}{\sum_{j=1}^n V^{(j)}} = \sum_{k=1}^n \bar{\omega}_T^{(i)} f(\mathbf{X}_T^{(k)}) \xrightarrow{n\to\infty} \mathbb{E}_p[f(X_{1:T})], \quad (23)$$

in distribution.

Remark 8.18. While (23) holds quite generally, the estimator $\mathrm{PF}^{(n)}_{M_{1:T},G_{1:T}}(f)$ typically converges

- quickly for functions that depend only on the last variable (or few last variables), that is, $f(x_{1:T}) = f(x_T)$ (or $f(x_{1:T}) = f(x_{(T-l):T})$ for $l \ll T$). [In the PF, the 'intermediate' distributions π_t are called the *filtering* distributions, from which the name *particle filter* arises.]
- much slower for $f(x_{1:T}) = f(x_1)$ when T is large.

In the latter case, instead of increasing n in a single run of PF, the algorithm may be run several times with fixed n...

Remark 8.19 (*). The step (iii) in Algorithm 8.15 is called *resampling* or *selection*. Algorithm 8.15 was introduced for SSMs in [10], using the specific choice $M_t = m_t$; this algorithm is known as the *bootstrap filter*. The rationale of resampling is, in intuitive terms, to discard 'unlikely paths', and concentrate on 'good candidates.' Similar procedure is used also in genetic algorithms, which aim for (global) optimisation.

Remark 8.20 (*). In fact, the *multinomial resampling* (iii) may be replaced with another procedure drawing non-independent set of indices $A_{t-1}^{(1:n)}$, which still satisfy *unbiasedness*, in the following sense:

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \mathbf{1}\left(A_{t-1}^{(i)} = j\right) \,\middle|\, X_1^{(1:n)}, X_{t-1}^{(1:n)}, A_1^{(1:n)}, \ldots, A_{t-2}^{(1:n)}\right] = \bar{\omega}_{t-1}^{(j)}.$$

For instance, stratified sampling is commonly used, and other choices are possible [cf. 4]. (NB: Even though stratification makes one-step conditional variance smaller, this does not necessarily mean more efficient overall estimator $\mathrm{PF}^{(n)}_{M_{1:T},G_{1:T}}(f)$, even though this is commonly observed empirically...)

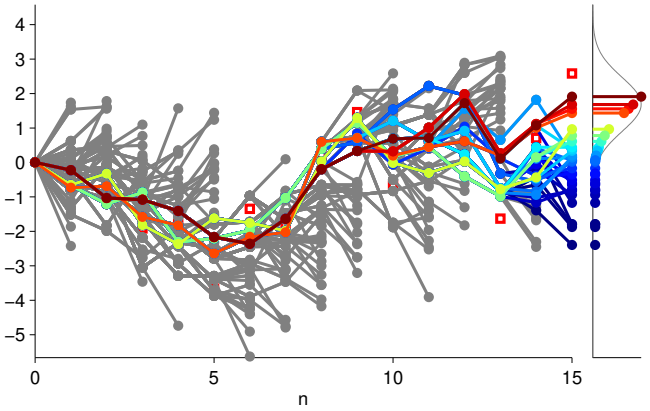
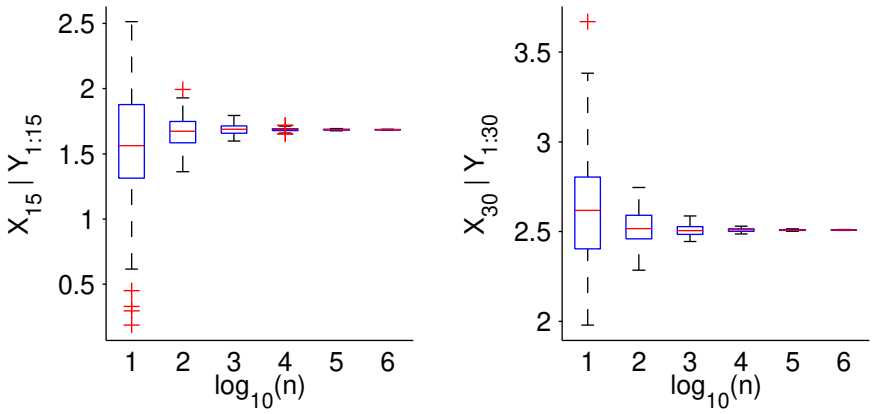


Figure 26: Some samples corresponding to the PF in Example 8.21. The grey paths show the 'dead branches': the ones that were not selected in resampling.



(a) (b)

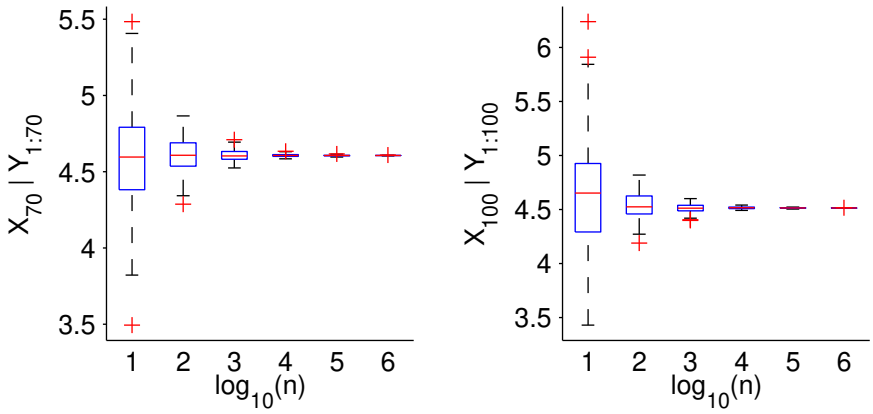
Figure 27: Box plot of the PF estimates with M_t corresponding to the prior, Example 8.6. Compare with 21. The estimates outperform also SIS with the 'optimal' proposal density in Example 8.7; see Figure 23.

Example 8.21. Let us revisit Example 8.6 with the particle filter; Figure 26 shows the samples produced. It is clear that the resampling helps to concentrate paths (compare with Figure 22). Figure 27 shows a summary of estimates, analogous to Figure 21, and Figure 28 demonstrates that the PF is reliable even for long series of observations, even with this simple proposal distribution.

(Choosing M_t to be q_t as in Example 8.7 would make the results even better, but it is noteworthy that even with $M_t = m_t$, the PF appears to perform reasonably well for bigger T...)

8.5 Unbiasedness of the particle filter

We shall not pursue a detailed proof of (23), but instead focus on the following non-asymptotic unbiasedness property of the PF [cf. 7, Theorem 7.4.2], which turns out to be key property for particle MCMC, which we discuss later.



(a) (b)

Figure 28: Box plot of the particle filter estimates with M_t corresponding to the prior, Compare with Figure 25. True value for $T = 100$ is approximately 4.514.

Theorem 8.22. *Under assumption* (22)*, for any* $f : \mathbb{S}^T \to \mathbb{R}$ *with* $\mathbb{E}_p[f(X)] < \infty$*, and any* $n \in \mathbb{N}$*, the following holds for the output of Algorithm 8.15:*

$$\mathbb{E}\left[\sum_{k=1}^{n} V^{(k)} f(\mathbf{X}^{(k)})\right] = \int p_u(x_{1:T}) f(x_{1:T}) \mathrm{d}x_{1:T}.$$

Proof. (*) Define the functions $f_T(x_{1:T}) := f(x_{1:T})$, and for $t = T, \ldots, 2$

$$f_{t-1}(x_{1:t-1}) := \int f_t(x_{1:t}) M_t(x_t \mid x_{1:t-1}) G_t(x_{1:t}) \mathrm{d}x_t.$$

Assumption (22) implies that $f_0 := \int M_1(x_1) G_1(x_1) f_1(x_1) \mathrm{d}x_1$ coincides with the desired integral, and all f_t are necessarily (almost everywhere) well-defined if the latter integral is well-defined.

Let us denote $X_{1:t}^{(*)} := \{X_{1:t}^{(i)}, i \in \{1{:}n\}\}$ and similarly $A_{1:t}^{(*)}$, and observe first that for $t = 2, \ldots, T$ and $i \in \{1{:}n\}$,

$$\begin{aligned}
&\mathbb{E}\big[\omega_t^{(i)} f_t(\mathbf{X}_t^{(i)}) \mid X_{1:t-1}^{(*)}, A_{1:t-2}^{(*)}\big] \\
&= \mathbb{E}\Big[\mathbb{E}\big[\omega_t^{(i)} f_t(\mathbf{X}_{t-1}^{(A_{t-1}^{(i)})}, X_t^{(i)}) \mid X_{1:t-1}^{(*)}, A_{1:t-1}^{(*)}\big] \;\Big|\; X_{1:t-1}^{(*)}, A_{1:t-2}^{(*)}\Big] \\
&= \mathbb{E}\left[\int M_t(x_t \mid \mathbf{X}_{t-1}^{(A_{t-1}^{(i)})}) G_t(\mathbf{X}_{t-1}^{(A_{t-1}^{(i)})}, x_t) f_t(\mathbf{X}_{t-1}^{(A_{t-1}^{(i)})}, x_t) \mathrm{d}x_t \;\middle|\; X_{1:t-1}^{(*)}, A_{1:t-2}^{(*)}\right] \\
&= \sum_{j=1}^{n} \mathbb{P}\big(A_{t-1}^{(i)} = j \mid X_{1:t-1}^{(*)}, A_{1:t-2}^{(*)}\big) f_{t-1}(\mathbf{X}_{t-1}^{(j)}),
\end{aligned}$$

so we may conclude that

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \omega_t^{(i)} f_t(\mathbf{X}_t^{(i)}) \;\middle|\; X_{1:t-1}^{(*)}, A_{1:t-2}^{(*)}\right] = \sum_{j=1}^{n} \bar{\omega}_{t-1}^{(j)} f_{t-1}(\mathbf{X}_{t-1}^{(j)}). \tag{24}$$

We may apply (24) recursively with $t = T, \dots, 2$, yielding

$$\begin{aligned}\mathbb{E}\Big[\sum_{k=1}^{n} V^{(k)} f(\mathbf{X}_T^{(k)})\Big] &= \mathbb{E}\Big[\Big(\prod_{t=1}^{T-1} \frac{1}{n}\omega_t^*\Big)\mathbb{E}\Big[\Big(\frac{1}{n}\omega_T^*\Big)\sum_{i=1}^{n} \bar{\omega}_T^{(i)} f_T(\mathbf{X}_T^{(i)}) \;\Big|\; X_{1:T-1}^{(*)}, A_{1:T-2}^{(*)}\Big]\Big] \\ &= \mathbb{E}\Big[\Big(\prod_{t=1}^{T-1} \frac{1}{n}\omega_t^*\Big)\sum_{i=1}^{n} \bar{\omega}_{T-1}^{(i)} f_{T-1}(\mathbf{X}_{T-1}^{(i)})\Big] = \dots \\ &= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\Big[\omega_1^* \bar{\omega}_1^{(i)} f_1(X_1^{(i)})\Big],\end{aligned}$$

which equals to f_0 by a similar calculation as above. □

One immediate consequence of the unbiasedness is that we may *combine* easily the output of independent particle filters, and deduce a consistent estimator as in self-normalised IS:

Corollary 8.23 (*)**.** *Fix $n \in \mathbb{N}$ and suppose $(V^{(1:n)}, \mathbf{X}^{(1:n)})$ is the output of Algorithm 8.15. Let $\zeta(f) := \sum_{k=1}^{n} V^{(k)} f(\mathbf{X}^{(k)})$ and $\zeta(1) := \sum_{k=1}^{n} V^{(k)}$. Suppose $(\zeta_i(f), \zeta_i(1))_{i\geq 1}$ are independent realisations of $(\zeta(f), \zeta(1))$, then*

(i) $E_{M_{1:T},G_{1:T}}^{(N,n)}(f) := \dfrac{\sum_{i=1}^{N} \zeta_i(f)}{\sum_{j=1}^{N} \zeta_j(1)} \xrightarrow{N\to\infty} \mathbb{E}_p[f(X)]$ *a.s.*

(ii) If $\mathbb{E}\big[|\zeta(f) - \zeta(1)\mathbb{E}_p[f(X)]|^2 + |\zeta(1)|^2\big] < \infty$, *then for any* $\alpha \in (0, \infty)$,

$$\mathbb{P}\Big(\mathbb{E}_p[f(X)] \in \Big[E_{M_{1:T},G_{1:T}}^{(N,n)}(f) \pm \alpha\sqrt{\hat{v}^{(N,n)}}\Big]\Big) \xrightarrow{N\to\infty} 1 - 2\Phi(-\alpha), \quad \textit{where}$$

$$\hat{v}^{(N,n)} := \frac{\sum_{i=1}^{N} \big(\zeta_i(f) - \zeta_i(1) E_{M_{1:T},G_{1:T}}^{(N,n)}(f)\big)^2}{\big(\sum_{j=1}^{N} \zeta_k(1)\big)^2}.$$

Proof. (i) follows from Theorem 8.22, because $\mathbb{E}[\zeta(f)]/\mathbb{E}[\zeta(1)] = \mathbb{E}_p[f(X)]$, and (ii) follows similarly as Theorem 4.23, once we observe that as $N \to \infty$,

$$N\hat{v}^{(N,n)} = \frac{\frac{1}{N}\sum_{i=1}^{N} \big(\zeta_i(f) - \zeta_i(1) E_{M_{1:T},G_{1:T}}^{(N,n)}(f)\big)^2}{\big(\frac{1}{N}\sum_{j=1}^{N} \zeta_k(1)\big)^2} \to \frac{\mathbb{E}\big[(\zeta(f) - \zeta(1)\mathbb{E}_p[f(X)])^2\big]}{\mathbb{E}_p[\zeta(1)]^2}.$$

□

Remark 8.24 (*). Suppose $\mathrm{PF}_{M_{1:T},G_{1:T}}^{(n,i)}(f)$ are independent realisations of $\mathrm{PF}_{M_{1:T},G_{1:T}}^{(n)}(f)$ in (23), then, unlike $E_{M_{1:T},G_{1:T}}^{(N,n)}(f)$, the naive combination $\frac{1}{N}\sum_{i=1}^{N} \mathrm{PF}_{M_{1:T},G_{1:T}}^{(n,i)}(f)$ is *not* consistent, because $\mathbb{E}[\mathrm{PF}_{M_{1:T},G_{1:T}}^{(n)}(f)] \neq E_p[f(X)]$ in general (even though, under general conditions, $\mathbb{E}[\mathrm{PF}_{M_{1:T},G_{1:T}}^{(n)}(f)] \to E_p[f(X)]$ as $n \to \infty$). On the contrary, the estimator $E_{M_{1:T},G_{1:T}}^{(N,n)}(f)$ is consistent with any n, and only requires an asymptotic in N.

9 Particle MCMC

As a final topic of the course, we discuss particle MCMC algorithms introduced in the seminal paper [3]. They are relatively straightforward combinations of MCMC and particle filter, in a way that allows for consistent estimation.

9.1 Parameterised model

Consider now a family of models, determined by a parameter $\theta \in \mathsf{T} = \mathbb{R}^{d_\theta}$:

$$p_u^{(\theta)}(x_{1:T}) = M_1^{(\theta)}(x_1)G_1^{(\theta)}(x_1)\prod_{t=2}^{T} M_t^{(\theta)}(x_t \mid x_{1:t-1})G_t^{(\theta)}(x_{1:t}).$$

and suppose that $\mathrm{pr}(\theta) \geq 0$ is a function such that

$$p_u(\theta, x_{1:T}) = \mathrm{pr}(\theta)p_u^{(\theta)}(x_{1:T})$$

determines an unnormalised probability distribution $p(\theta, x_{1:T}) \propto p_u(\theta, x_{1:T})$ on $\mathsf{T} \times \mathbb{S}^T$.

Remark 9.1. In particular, if $p_u^{(\theta)}(x_{1:T})$ correspond to a parameterised SSM as in (21), that is,

$$p_u^{(\theta)}(x_{1:T}) = m_1^{(\theta)}(x_1)g_1^{(\theta)}(x_1)\prod_{t=2}^{T} m_t^{(\theta)}(x_{t-1}, x_t)g_t^{(\theta)}(x_t, y_t),$$

(cf. Remark 8.8), and pr is the prior of the parameters θ, then $p_u(\theta, x_{1:T})$ corresponds to the conditional distribution $(\theta, X_{1:T}) \mid Y_{1:T} = y_{1:T}$. This is what we care about if we are interested in (full) Bayesian time-series analysis using SSMs...

9.2 Particle marginal Metropolis-Hastings algorithm

Suppose that $n \in \mathbb{N}$ is fixed, and that $q(\theta, \theta')$ is a Metropolis-Hastings proposal on T.

Algorithm 9.2 (Particle marginal Metropolis-Hastings)**.** Let $\Theta_0 \in \mathsf{T}$, and let $(V_0^{(1:n)}, \mathbf{X}_0^{(1:n)})$ be the output of PF Algorithm 8.15 with $(n,\ M_{1:T}^{(\Theta_0)},\ G_{1:T}^{(\Theta_0)})$. For $k = 1, 2, \ldots, N$, iterate:

(i) Sample $\hat{\Theta}_k \sim q(\Theta_{k-1}, \cdot)$.

(ii) Run PF Algorithm 8.15 with $(n,\ M_{1:T}^{(\hat{\Theta}_k)},\ G_{1:T}^{(\hat{\Theta}_k)})$, and call its output $(\hat{V}_k^{(1:n)}, \hat{\mathbf{X}}_k^{(1:n)})$.

(iii) With probability

$$\min\left\{1, \frac{\mathrm{pr}(\hat{\Theta}_k)q(\hat{\Theta}_k, \Theta_{k-1})\sum_{i=1}^n \hat{V}_k^{(i)}}{\mathrm{pr}(\Theta_{k-1})q(\Theta_{k-1}, \hat{\Theta}_k)\sum_{j=1}^n V_{k-1}^{(j)}}\right\},$$

accept and set $(\Theta_k, V_k^{(1:n)}, \mathbf{X}_k^{(1:n)}) \leftarrow (\hat{\Theta}_k, \hat{V}_k^{(1:n)}, \hat{\mathbf{X}}_k^{(1:n)})$; otherwise reject and set $(\Theta_k, V_k^{(1:n)}, \mathbf{X}_k^{(1:n)}) \leftarrow (\Theta_{k-1}, V_{k-1}^{(1:n)}, \mathbf{X}_{k-1}^{(1:n)})$.

Report the following approximation of $\mathbb{E}_p[f(\Theta, X_{1:T})]$:

$$I_{\mathrm{PMMH}}^{(N,n)}(f) := \frac{1}{N}\sum_{k=1}^{N} \frac{\sum_{i=1}^n V_k^{(i)} f(\Theta_k, \mathbf{X}_k^{(i)})}{\sum_{j=1}^n V_k^{(j)}}.$$

Theorem 9.3. *Suppose that $q(\theta, \theta') > 0$ for all $\theta, \theta \in \mathsf{T}$. Then, for any fixed $n \in \mathbb{N}$,*

$$I^{(N,n)}_{\mathrm{PMMH}}(f) \xrightarrow{N\to\infty} \mathbb{E}_p[f(\Theta, X_{1:T})] \qquad a.s.,$$

whenever the expectation is finite.

Proof. (**) Let $Q_\theta(x^{(1:n)}_{1:T}, a^{(1:n)}_{1:T-1})$ stand for the distribution of *all* random variables $X^{(i)}_t$ and $A^{(i)}_t$ generated in Algorithm 8.15 with $(n, M^{(\theta)}_{1:T}, G^{(\theta)}_{1:T})$, and let $v^{(k)}(x^{(1:n)}_{1:T}, a^{(1:n)}_{1:T-1})$ and $\mathbf{x}^{(k)}(x^{(1:n)}_{1:T}, a^{(1:n)}_{1:T-1})$ stand for how the outputs $V^{(k)}$ and $\mathbf{X}^{(k)}$ are determined from $X^{(1:n)}_{1:T}$ and $A^{(1:n)}_{1:T-1}$. Define the following unnormalised distribution (sic!)

$$\pi_u(\theta, x^{(1:n)}_{1:T}, a^{(1:n)}_{1:T-1}) = \mathrm{pr}(\theta) Q_\theta(x^{(1:n)}_{1:T}, a^{(1:n)}_{1:T-1}) \sum_{k=1}^{n} v^{(k)}(x^{(1:n)}_{1:T}, a^{(1:n)}_{1:T-1}),$$

then Algorithm 9.9 may be seen as a Metropolis-Hastings with target $\pi \propto \pi_u$ and proposal $\tilde{q}(\theta, x^{(1:n)}_{1:T}, a^{(1:n)}_{1:T-1}; \hat{\theta}, \hat{x}^{(1:n)}_{1:T}, \hat{a}^{(1:n)}_{1:T-1}) = q(\theta, \hat{\theta}) Q_{\hat{\theta}}(\hat{x}^{(1:n)}_{1:T}, \hat{a}^{(1:n)}_{1:T-1})$.

Theorem 8.22 implies that for any $\varphi : \mathbb{S}^T \to \mathbb{R}$ such that the integral below is finite,

$$\sum_{a^{(1:n)}_{1:T-1}} \int Q_\theta(x^{(1:n)}_{1:T}, a^{(1:n)}_{1:T-1}) \Big(\sum_{i=1}^{n} v^{(i)}(x^{(1:n)}_{1:T}, a^{(1:n)}_{1:T-1}) \varphi\big(\mathbf{x}^{(i)}(x^{(1:n)}_{1:T}, a^{(1:n)}_{1:T-1})\big) \Big) \mathrm{d}x^{(1:n)}_{1:T}$$
$$= \int p^{(\theta)}_u(x_{1:T}) \varphi(x_{1:T}) \mathrm{d}x_{1:T}.$$

This implies that for any function $f : \mathsf{T} \times \mathbb{S}^T \to \mathbb{R}$,

$$\mathbb{E}_\pi \left[\frac{\sum_{i=1}^n v^{(i)}(\mathbf{X}^{(1:n)}_{1:T}, A^{(1:n)}_{1:T-1}) f(\Theta, \mathbf{x}^{(i)}(\mathbf{X}^{(1:n)}_{1:T}, A^{(1:n)}_{1:T-1}))}{\sum_{j=1}^n v^{(j)}(\mathbf{X}^{(1:n)}_{1:T}, A^{(1:n)}_{1:T-1})} \right] = \mathbb{E}_p[f(\Theta, X_{1:T})].$$

The proof is complete once we are convinced that the Markov chain $(\Theta_i, X^{(1:n)}_{1:T}, A^{(1:n)}_{1:T-1})$, and consequently $V^{(1:n)}_i, \mathbf{X}^{(1:n)}_i)_{i\geq 1}$, is Harris, which follows because it is π-irreducible Metropolis-Hastings. □

Remark 9.4 (*). If we are only interested in the variable Θ in p, the PMMH Algorithm 8.15 may be seen as an instance of a so-called *pseudo-marginal* Metropolis-Hastings algorithm [1, 14].

9.3 Conditional particle filter (*)

The PMMH is a relatively simple combination of the PF and Metropolis-Hastings. It can, however, get 'stuck' (many rejects), when $\sum_j V^{(j)}_{k-1}$ gets over-estimated (unusually high value). The paper [3] contained also another scheme, which has better scalability properties wrt. T. It is based on a modified *conditional* particle filter (CPF) algorithm.

Algorithm 9.5. CPF$(n, M^{(\theta)}_{1:T}, G^{(\theta)}_{1:T}, X^*_{1:T})$

(i) Set $X_1^{(1)} = X_1^*$ and sample $X_1^{(2:n)} \overset{\text{i.i.d.}}{\sim} M_1$. Set $\mathbf{X}_1^{(1:n)} = X_1^{(1:n)}$.

(ii) Calculate $\omega_1^{(i)} := G_1(\mathbf{X}_1^{(i)})$ and set $\bar{\omega}_1^{(i)} := \omega_1^{(i)}/\omega_1^*$ where $\omega_1^* = \sum_{j=1}^n \omega_1^{(j)}$.

For $t = 2, \ldots, T$, do:

(iii) Sample $A_{t-1}^{(2:n)}$ independently with $\mathbb{P}(A_{t-1}^{(i)} = j) = \bar{\omega}_{t-1}^{(j)}$, $j \in 1{:}n$.

(iv) Set $X_t^{(1)} = X_t^*$ and sample $X_t^{(i)} \sim M_t(\,\cdot \mid \mathbf{X}_{t-1}^{(A_{t-1}^{(i)})})$ for $i = 2{:}n$.

(v) Set $\mathbf{X}_t^{(1)} = (\mathbf{X}_{t-1}^{(1)}, X_t^*)$ and $\mathbf{X}_t^{(i)} = (\mathbf{X}_{t-1}^{(A_{t-1}^{(i)})}, X_t^{(i)})$ for $i = 2{:}n$.

(vi) Calculate $\omega_t^{(i)} := G_t(\mathbf{X}_t^{(i)})$ and set $\bar{\omega}_t^{(i)} := \omega_t^{(i)}/\omega_t^*$ where $\omega_t^* = \sum_{j=1}^n \omega_t^{(j)}$.

Draw $B \sim \text{Categorical}(\bar{\omega}_T^{(1:n)})$ and output $\mathbf{X}_T^{(B)}$.

Remark 9.6. The CPF defines a *Markov transition* in the trajectory space $\mathbb{S}^T$. It turns out that the transition is reversible with respect to $p^{(\theta)} \propto p_u^{(\theta)}$, again thanks to Theorem 8.22.

Remark 9.7. When $M_t(x_t \mid x_{1:t-1}) = M_t(x_t \mid x_{t-1})$ and $G_t(x_{1.t}) = G_t(x_{t-1:t})$, the CPF may be substantially enhanced by applying it together with so-called *backward sampling* [32] (or the equivalent ancestor sampling [15]). That is, instead of selecting one of $\mathbf{X}_T^{(1:n)}$, the output is 'reselected' among all particles $X_{1:T}^{(1:n)}$ as follows: $(X_1^{(B_1)}, \ldots, X_{T-1}^{(B_{T-1})}, X_T^{(B_T)})$, where $B_T = B$ and for $t = T-1, \ldots, 1$:

$$\mathbb{P}(B_t = i \mid B_{t+1} = j) \propto \omega_t^{(i)} M_{t+1}(X_{t+1}^{(j)} \mid X_t^{(i)}) G_{t+1}(X_t^{(i)}, X_{t+1}^{(j)}). \tag{25}$$

The backward sampling version of the CPF is also $p^{(\theta)}$-reversible [6]. (Note also that if $G_t(x_{t-1:t}) = G_t(x_t)$, then the term $G_{t+1}(\,\cdot\,)$ vanishes from (25).)

Remark 9.8. When using the CPF n has to increase in T linearly $n = O(T)$, but with the backward sampling modification, n need not be increased wrt. T [cf. 13].

9.4 Particle Gibbs (*)

Algorithm 9.9 (Particle Gibbs)**.** Let $\Theta_0 \in \mathsf{T}$ and $\mathbf{X}_0 \in \mathsf{Sp}^T$ such that $p_u(\Theta_0, \mathbf{X}_0) > 0$.

For $k = 1, 2, \ldots, N$, iterate:

(i) $\mathbf{X}_k \leftarrow \text{CPF}(n, M_{1:T}^{(\Theta_{k-1})}, G_{1:T}^{(\Theta_{k-1})}, \mathbf{X}_{k-1})$.

(ii) Sample $\hat{\Theta}_k \sim q(\Theta_{k-1}, \,\cdot\,)$, and with probability

$$\min\left\{1, \frac{p_u(\hat{\Theta}_k, \mathbf{X}_k) q(\hat{\Theta}_k, \Theta_{k-1})}{p_u(\Theta_{k-1}, \mathbf{X}_k) q(\Theta_{k-1}, \hat{\Theta}_k)}\right\},$$

accept and set $\Theta_k \leftarrow \hat{\Theta}_k$; otherwise reject and set $\Theta_k \leftarrow \Theta_{k-1}$.

Report the following approximation of $\mathbb{E}_p[f(\Theta, X_{1:T})]$:

$$I_{\text{PG}}^{(N,n)}(f) := \frac{1}{N} \sum_{k=1}^{N} f(\Theta_k, \mathbf{X}_k).$$

Theorem 9.10. *The particle Gibbs defines a Markov transition which leaves p invariant.*

Proof. Step (i) is a component-wise update of $\mathbf{X}_{k-1} \to \mathbf{X}_k$ by the CPF that leaves the conditional $\propto p_u^{(\Theta_{k-1})}$ invariant, and the step (ii) is a Metropolis-within-Gibbs step. □

Remark 9.11. Consider the PMMH output, and sample $I_k \sim \text{Categorical}(W_k^{(1:n)})$, where $W_k^{(i)} = V_k^{(i)}/(\sum_{j=1}^n V_k^{(j)})$, then we may also use

$$\hat{I}_{\text{PMMH}}^{(N,n)}(f) := \frac{1}{N}\sum_{k=1}^{N} f(\Theta_k, \mathbf{X}_k^{(I_k)}),$$

which remains consistent, but it worse in terms of variance.

Analogously, it is direct to use a more 'refined' estimator in the PG, where the selection of output (sampling of B in Algorihm 9.5) is 'Rao-Blackwellised'...

Remark 9.12. Some authors refer also the CPF as 'particle Gibbs', but the terminology here follows the terminology in the original paper [3].

References

[1] C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *Ann. Statist.*, 37(2):697–725, 2009.

[2] C. Andrieu and J. Thoms. A tutorial on adaptive MCMC. *Statist. Comput.*, 18 (4):343–373, Dec. 2008.

[3] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 72(3):269–342, 2010. (with discussion).

[4] O. Cappé, E. Moulines, and T. Rydén. *Inference in Hidden Markov Models*. Springer, 2005.

[5] B. Carpenter, A. Gelman, M. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. A. Brubaker, J. Guo, P. Li, and A. Riddell. Stan: A probabilistic programming language. *J. Stat. Softw.*, 20:1–37, 2016.

[6] N. Chopin and S. S. Singh. On particle Gibbs sampling. *Bernoulli*, 21(3):1855–1883, 2015.

[7] P. Del Moral. *Feynman-Kac Formulae:: Genealogical and Intercating Particle Systems with Applications*. Springer, New York, 2004.

[8] A. Durmus, E. Moulines, and E. Saksman. On the convergence of Hamiltonian Monte Carlo. *Ann. Statist.*, to appear.

[9] J. M. Flegal and G. L. Jones. Batch means and spectral variance estimators in Markov chain Monte Carlo. *Ann. Statist.*, 38(2):1034–1070, 2010.

[10] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, 140(2):107–113, 1993.

[11] H. Haario, E. Saksman, and J. Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242, 2001.

[12] A. M. Johansen, L. Evers, and N. Whiteley. Monte Carlo methods. Lecture notes, University of Bristol, 2010.

[13] A. Lee, S. S. Singh, and M. Vihola. Coupled conditional backward sampling particle filter. *Ann. Statist.*, to appear.

[14] L. Lin, K. Liu, and J. Sloan. A noisy Monte Carlo algorithm. *Phys. Rev. D*, 61(7): 074505, 2000.

[15] F. Lindsten, M. I. Jordan, and T. B. Schön. Particle Gibbs with ancestor sampling. *J. Mach. Learn. Res.*, 15(1):2145–2184, 2014.

[16] M. Matsumoto and T. Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 8(1):3–30, 1998.

[17] S. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Cambridge University Press, second edition, 2009. ISBN 978-0-521-73182-9.

[18] R. M. Neal. MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, editors, *Handbook of Markov Chain Monte Carlo*, volume 2. CRC Press New York, NY, 2011.

[19] G. Nicholls. Part A Simulation and statistical programming. Lecture notes, University of Oxford, 2015.

[20] E. Nummelin. MC's for MCMC'ists. *Int. Statist. Rev.*, 70(2):215–240, 2002.

[21] A. B. Owen. *Monte Carlo theory, methods and examples*. 2013.

[22] A. Penttinen. MATS442 Stokastinen simulointi. Lecture notes, University of Jyväskylä, 2010. (In Finnish).

[23] P. H. Peskun. Optimum Monte-Carlo sampling using Markov chains. *Biometrika*, 60(3):607–612, 1973.

[24] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, New York, 1999.

[25] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, New York, second edition, 2004.

[26] G. O. Roberts and J. S. Rosenthal. Harris recurrence of Metropolis-within-Gibbs and trans-dimensional Markov chains. *Ann. Appl. Probab.*, 16(4):2123–2139, 2006.

[27] G. O. Roberts, A. Gelman, and W. R. Gilks. Weak convergence and optimal scaling of random walk Metropolis algorithms. *Ann. Appl. Probab.*, 7(1):110–120, 1997.

[28] D. J. Spiegelhalter, A. Thomas, N. G. Best, W. R. Gilks, and D. Lunn. BUGS: Bayesian inference using Gibbs sampling, 1996–2008. URL `http://www.mrc-bsu.cam.ac.uk/bugs/`.

[29] L. Tierney. Markov chains for exploring posterior distributions. *Ann. Statist.*, 22 (4):1701–1728, 1994.

[30] L. Tierney. A note on Metropolis-Hastings kernels for general state spaces. *Ann. Appl. Probab.*, 8(1):1–9, 1998.

[31] M. Vihola. Robust adaptive Metropolis algorithm with coerced acceptance rate. *Statist. Comput.*, 22(5):997–1008, 2012.

[32] N. Whiteley. Discussion on Particle Markov chain Monte Carlo methods. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 72(3):306–307, 2010.