

Lectures on stochastic simulation

Matti Vihola

March 11, 2018

Contents

1	Introduction	3
1.1	Conventions	4
1.2	The Monte Carlo method	5
1.3	Properties of Monte Carlo estimators	7
1.4	About uniformly distributed pseudo-random numbers	9
1.5	Monte Carlo vs. other numerical integration methods	9
2	Variable transformation methods	10
2.1	Inverse distribution function method	10
2.2	Distribution of transformed random variables	12
2.3	(Multivariate) normal random variables	13
2.4	Relations of probability distributions (*)	15
2.5	Spherically/elliptically symmetric distributions (*)	15
3	Rejection sampling	16
3.1	Rejection sampling algorithm	17
3.2	Unnormalised distributions and rejection sampling	18
4	Importance sampling	21
4.1	Unbiased importance sampling	21
4.2	Application: Rare event estimation	25
4.3	Self-normalised importance sampling	26
5	Variance reduction techniques	29
5.1	Rao-Blackwellisation	29
5.2	Stratification	30
5.3	Introducing dependence: antithetic variables	32
5.4	Control variates (*)	33
6	Markov chain Monte Carlo	33
6.1	Recap of some Markov chain theory	34
6.2	Reversibility	35
6.3	The Metropolis-Hastings algorithm on discrete $\mathbb{X}$	36
6.4	The Metropolis-Hastings algorithm on $\mathbb{X} = \mathbb{R}^d$	39
6.5	On tuning of random-walk Metropolis (*)	42
6.6	Componentwise updates	44
6.7	About ‘gradient-informed’ proposals (*)	50
7	MCMC convergence and mixing	51
7.1	Burn-in bias	52
7.2	Asymptotic variance of MCMC	52
7.3	Practical summary	54
7.4	Optimising MCMC	55
8	Sequential Monte Carlo	56

8.1	General state-space models/hidden Markov models	57
8.2	First attempt: Sequential importance sampling	58
8.3	Generic SIS	61
8.4	The particle filter	62
8.5	Unbiasedness of PF estimator	64
9	Particle MCMC	67
9.1	Bayesian state-space model	68
9.2	Particle marginal Metropolis-Hastings algorithm	68
9.3	Conditional particle filter/particle Gibbs (*)	69

Preface

This is a summary of the lectures of the Spring 2018 course “MATS442 Stochastic simulation” at Department of Mathematics and Statistics, University of Jyväskylä. These notes are inspired by the lecture notes of Antti Penttinen [18], Nicholls [16] and Johansen, Evers and Whiteley [11], and by the textbooks [21, 4].

The purpose of these notes is to support the lectures, so they may not be well suited for self study. Some important methods (or examples) are covered in the problems classes. The “Monte Carlo Statistical Methods” book by Christian P. Robert and George Casella [20, 21] is a recommended supporting material. Other references to the literature are given later 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, of which many are touched within this course.

After taking this course, you will be able to:

- *apply* simulation methods in practice,
- *understand* why the methods work, (and why they sometimes do not work),
- *modify* the methods and *combine* them for your needs (a particular application).

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

The R 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 the R programming environment in the course. If you are not familiar with R, that should not be a problem, as no advanced R skills will be needed. There is also plenty of online introductory material available for self-study.

- You can take this course even if you never used R before.
- TILA410 R course is warmly recommended.
 - Web course, ideal to complete during this course.
 - If you need English material, please contact the course teacher (Johanna Ärje).
- There is a lot of other freely available R introductions online.

1.1 Conventions

We use the symbols $\mathbb{P}$ and $\mathbb{E}$ for probability and expectation, denote random variables¹ 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 $\mathbb{X} = \{x_1, \dots, x_m\}$ or a countably infinite space $\mathbb{X} = \{x_1, x_2, \dots\}$.

The notation $X \sim p$ means that X is a random variable has distribution p and $X_1, \dots, X_n \stackrel{\text{i.i.d.}}{\sim} p$ or $(X_k)_{k \geq 1} \stackrel{\text{i.i.d.}}{\sim} p$ means $(X_k)_{k \geq 1}$ are independent and each $X_k \sim p$. For $f : \mathbb{X} \rightarrow \mathbb{R}$, we write $\mathbb{E}_p[f(X)]$ meaning the expectation of $f(X)$ with $X \sim p$. That is,

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

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

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

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

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

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

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

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

(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 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 $dx = \mu(dx)$, in which case $p \equiv 1$. Most of the techniques presented in the course generalise into such a setting directly.

The functions $f : \mathbb{X} \rightarrow \mathbb{R}$ for which expectations are defined must, of course, be measurable. We will assume the required measurability of given functions (and sets) hereafter without explicit notification.

We do not explicitly state the probability 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 with countably generated spaces, such as discrete-time Markov chains, for which existence of the underlying probability space is standard.

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 $H_k = 0$ otherwise², then

$$4\left(\frac{1}{n} \sum_{k=1}^n H_k\right) \xrightarrow{n \rightarrow \infty} \pi, \quad (\text{almost surely})$$

by the (strong) law of large numbers.

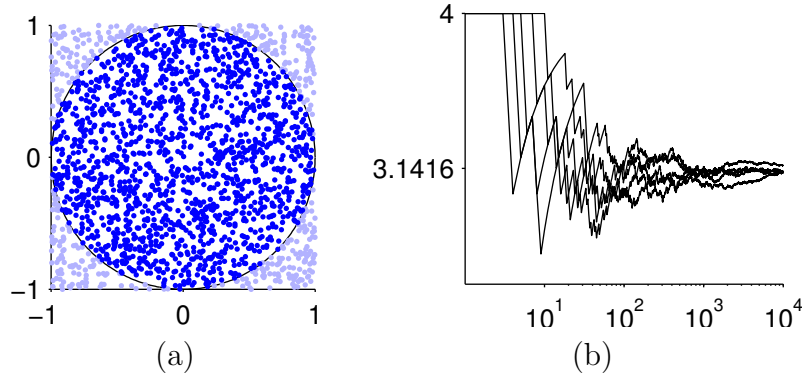


Figure 1: The rain drops example: (a) ‘rain drops’ falling inside the unit disc highlighted (b) five realisations of the experiment: estimates of π converge as $n \rightarrow \infty$ (note the log scale on n).

Definition 1.5 (Monte Carlo). Assume $X_1, \dots, X_n \stackrel{\text{i.i.d.}}{\sim} p$. Then, for $f : \mathbb{X} \rightarrow \mathbb{R}$,

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

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

Example 1.6. Example 1.4 corresponds to

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

or, equivalently, to simulated Bernoulli random variables,

- $\mathbb{X} = \{0, 1\}$
- $(H_k)_{k \geq 1} \stackrel{\text{i.i.d.}}{\sim} \text{Bernoulli}(\pi/4)$ (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 the expectation is a finite sum and can, in principle, be computed exactly. However, it may be that the space $\mathbb{X}$ has huge number of elements, for example if $\mathbb{X}$ is the set of all 100×100 binary images $\mathbb{X} = \{0, 1\}^{100 \times 100}$, the Monte Carlo method can still be relevant...

Example 1.8. We may be interested in calculating

- Probabilities: $\mathbb{P}(X \in A)$ for $X \sim p$, then we choose $f(x) = \mathbf{1}(x \in A)$ (cf. Example 1.4).
- Multiple expectations (or probabilities) simultaneously: $\mathbb{E}_p[f_k(X)]$ for a set of functions $f_1, \dots, f_m$. For example, the mean of random vector $X = (X^{(1)}, \dots, X^{(d)}) \sim p$ is a vector of means of individual coordinates,

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

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

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

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

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 $Y_1, Y_2, \dots$ are i.i.d. random numbers such that $\mu = \mathbb{E}[Y_1]$ is finite. Then,*

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

Remark 1.10. Recall that $Z_n \rightarrow Z$ a.s. implies $Z_n \rightarrow Z$ in probability, and that

$$\begin{aligned} Z_n \rightarrow Z \text{ a.s.} & \iff \mathbb{P}(Z_n \rightarrow Z) = 1 \\ Z_n \rightarrow Z \text{ in probability} & \iff \text{For any } \epsilon > 0, \mathbb{P}(|Z_n - Z| \leq \epsilon) \rightarrow 1. \end{aligned}$$

That is, you may always replace “almost surely” in (2) by “in probability.” The latter statement is known as the weak law of large numbers.

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

$$\frac{1}{\sqrt{n}} \sum_{k=1}^n (Y_k - \mu) \xrightarrow{n \rightarrow \infty} N(0, \sigma^2) \quad \text{in distribution,}$$

in other words,

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

where Φ is the standard normal c.d.f., that is, $\Phi(x) := \mathbb{P}(Z \leq x)$ 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 $n \geq 1$.*

Strong consistency *If $\mathbb{E}_p[f(X)]$ is finite, then $I_p^{(n)}(f) \xrightarrow{n \rightarrow \infty} \mathbb{E}_p[f(X)]$ almost surely.*

Variance *If $\text{Var}_p[f(X)] < \infty$, then $\text{Var}[I_p^{(n)}(f)] = \frac{1}{n} \text{Var}_p[f(X)]$.*

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,

$$\text{Var}[I_p^{(n)}(f)] = \frac{1}{n^2} \text{Var}\left(\sum_{k=1}^n Y_k\right) = \frac{1}{n} \text{Var}(Y_1). \quad \square$$

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

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

Furthermore, if $\hat{\sigma}_n^2$ stands for the sample variance

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

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

$$\mathbb{P}(\sqrt{n}[I_p^{(n)}(f) - \mathbb{E}_p[f(X)]] \leq \beta \hat{\sigma}_n) \xrightarrow{n \rightarrow \infty} \Phi(\beta).$$

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

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

Recall the following lemma for the proof:

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

(i) $X_n Y_n \rightarrow X y$ in distribution.

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

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

Consider then (iii), and observe that

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

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 $\epsilon > 0$. $\square$

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 can be used directly to build non-asymptotic upper bounds by Chebychev's inequality,

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

Note that we need to know $\text{Var}_p[f(X)]$ (or we need to be able to upper bound for $\text{Var}_p[f(X)]$) in order to have an exact upper bound.

Remark 1.17 (*). The Chebychev tail bound is rather pessimistic: 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

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

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 assume that we can access $(U_k)_{k \geq 1} \stackrel{\text{i.i.d.}}{\sim} \mathcal{U}(0, 1)$, 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* [13], which is the default PRNG for many environments, including 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).

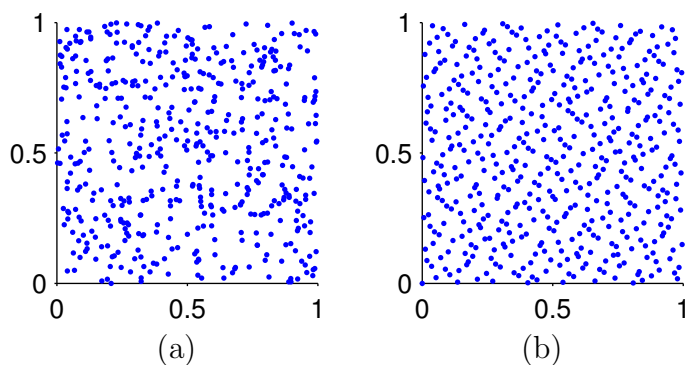


Figure 2: 500 points on $[0, 1]^2$ which are (a) i.i.d. pseudo-random (b) from a low-discrepancy sequence (Halton).

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

- Transform single $U \sim U(0, 1)$ into a single $X \sim p$.
- Transform multiple $U_1, \dots, U_n \stackrel{\text{i.i.d.}}{\sim} U(0, 1)$ into single or multiple $X_1, \dots, X_m \stackrel{\text{i.i.d.}}{\sim} p$, 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

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

Theorem 2.1. Assume $U \sim \mathcal{U}(0, 1)$ and let $F : A \rightarrow (0, 1)$ be a c.d.f. on an open interval³ $A \subset \mathbb{R}$, which is continuous and strictly increasing, with inverse $F^{-1} : (0, 1) \rightarrow A$. 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 $x \in A$:

$$\begin{aligned}\mathbb{P}(X \leq x) &= \mathbb{P}(F^{-1}(U) \leq x) \\ &= \int_0^1 \mathbf{1}(F^{-1}(u) \leq x) \, du \\ &= \int_0^1 \mathbf{1}(u \leq F(x)) \, du = F(x). \quad \square\end{aligned}$$

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

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

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

$$F(x) = \int_0^x r \exp(-rt) \, dt = 1 - \exp(-rx),$$

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 <- 1e3      # number of samples to simulate
u <- runif(n)  # n independent U(0,1)
x <- -log(u)   # n independent Exp(1)
```

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

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

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

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

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 $X \sim \text{Geometric}(\tilde{p})$ then

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

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 <- 1e3; q <- 3/4
u <- runif(n)
x <- ceiling(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 $F : \mathbb{R} \rightarrow [0, 1]$ be a c.d.f.⁴. Define

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

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 min above is well-defined). The proof follows as in the proof of Theorem 2.1, by noticing that

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

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. $\square$

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

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

Its generalised inverse is

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

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

```
u <- runif(1e3)
x <- -log(2*u) * (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.

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

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

4. Recall that F is a c.d.f. if it is increasing⁵, right-continuous, $\lim_{x \rightarrow -\infty} F(x) = 0$ and $\lim_{x \rightarrow \infty} F(x) = 1$.

6. The function F^{-1} is called the *generalised inverse c.d.f.*

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

$$p_Y(y) = F'_Y(y) = F'_X(g^{-1}(y))(g^{-1})'(y) = \frac{p_X(g^{-1}(y))}{g'(g^{-1}(y))},$$

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

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

Theorem 2.7. *Suppose $X \sim p_X$ and $S := \text{supp}(p) := \{x \in \mathbb{R}^d : p_X(x) > 0\}$ is an open set. If $g : S \rightarrow \mathbb{R}^d$ 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 $\text{supp}(p)$ can be partitioned (up to set of volume (measure) zero) into disjoint open sets $S_1, S_2, \dots$ such that g satisfies the conditions required in Theorem 2.7, then Theorem 2.7 can be applied piecewise, leading into

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

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

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

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

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

The random vector $X := (X_1, \dots, X_d)^T$, where $(X_k) \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$ 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 $U_1, U_2 \stackrel{i.i.d.}{\sim} \mathcal{U}(0, 1)$ and define

$$\begin{aligned} X_1 &:= R \cos(T) \\ X_2 &:= R \sin(T), \end{aligned} \quad \text{where} \quad \begin{aligned} R &:= \sqrt{-2 \ln U_1} \\ T &:= 2\pi U_2. \end{aligned}$$

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

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 $t(x, y) := \text{atan2}(y, x)$. □

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 $Z \sim N(0, I_d)$,

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

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

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

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-triangular structure allows some savings when computing the transform (3).

Example 2.13. Generating bivariate Gaussians.

<code>m <- c(-1,1)</code>	<code># Mean vector</code>
<code>S <- matrix(c(5,-3,-3,4), nrow=2)</code>	<code># Covariance matrix</code>
<code>L <- t(chol(S))</code>	<code># chol() gives U=L^T</code>
<code>Z <- matrix(rnorm(2*1000), nrow=2)</code>	<code># Z[,i] i.i.d. N(0,I_2)</code>
<code>X <- m + L %*% Z</code>	<code># X[,i] i.i.d. N(m,S)</code>

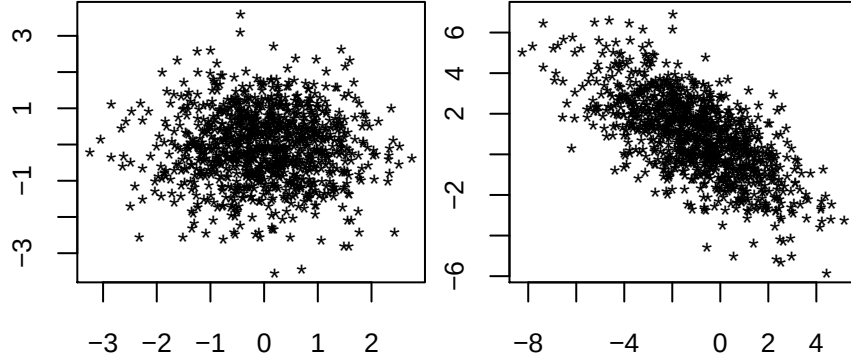


Figure 3: Standard bivariate samples $(Z_k)_{k \geq 1}$ (left) and $N(m, S)$ samples (X_k) generated in Example 2.13.

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.

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

Inverse CDF method is not easily applicable. Instead,

- (a) Simulate $Y_1, \dots, Y_\alpha \stackrel{\text{i.i.d.}}{\sim} \text{Exp}(1)$.
- (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

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

so the m.g.f. of X is

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

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,*

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

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

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

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

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

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,

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

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 <- 1e3; d <- 2; Z <- matrix(rnorm(d*n), nrow=d)
X <- apply(Z, 2, function(z) {(runif(1))^(1/d)*(z/sqrt(sum(z^2)))})
```

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 $(V_k)_{k \geq 1} \stackrel{\text{i.i.d.}}{\sim} \mathcal{U}([0, 1]^2)$. 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, $(\hat{V}_k) \stackrel{\text{i.i.d.}}{\sim} \mathcal{U}(D)$, 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

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

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

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

- (A) If $U_T \leq \frac{p(Y_T)}{Mq(Y_T)}$, 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’ $B_k := \mathbf{1}(U_k \leq h(Y_k))$, 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 $t \in \mathbb{N}$,

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

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. 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

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

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

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

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

```
M <- sqrt(2*pi)*exp(-.5)
n <- 10e3; x <- numeric(n)
while (n>0) {
  y <- rcauchy(1)
  if (M*runif(1) < dnorm(y)/dcauchy(y)) {
    x[n] <- y; n <- n-1
  }
}
```

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

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

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

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

with the *normalising constant*

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

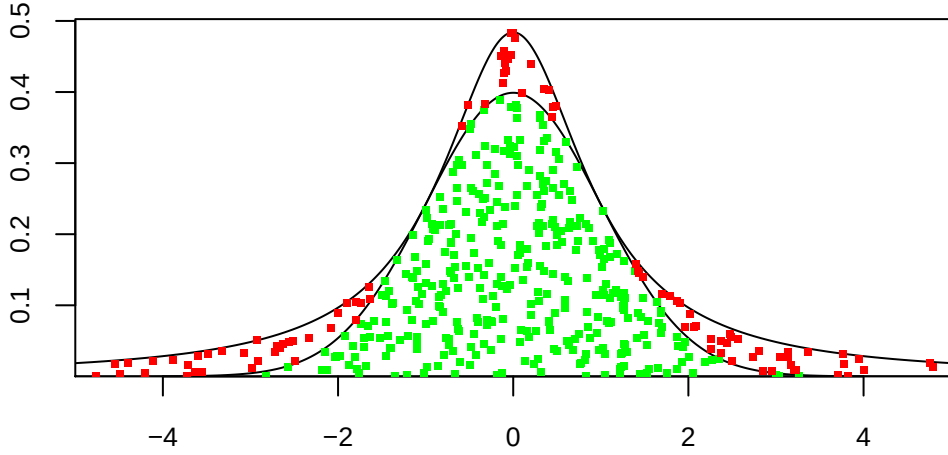


Figure 4: Illustration of Example 3.8: The points $(Y_k, U_k Mq(Y_k))$ are *distributed uniformly* in between the x -axis and the function $Mq(x)$ (the upper curve). The points that fall below the curve p are accepted (green), others are rejected (red).

The distribution $p(x)$ is fully determined by $p_u(x)$, even though we could not calculate values of $p(x)$.

Example 3.9. Suppose we know $p(x)$, and we are interested in the tail distribution

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

Example 3.10. In Bayesian inference, we are interested in a conditional distribution (posterior) which may be of the form

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

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

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

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

(5)

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

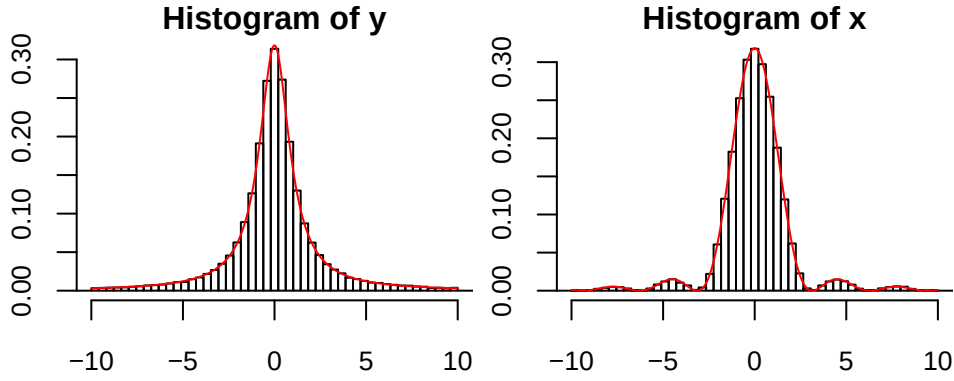


Figure 5: Simulated samples from the standard Cauchy distribution (left) and samples from $p(x) \propto \sin^2(x)/x^2$ (right) with corresponding densities.

- (A) If $U_T \leq \frac{p_u(Y_T)}{M q_u(Y_T)}$, 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, $\frac{p_u(y)}{M q_u(y)} = \frac{p(y)}{M q(y)}$, so Algorithm 3.11 coincides with Algorithm 3.2 with $\hat{M} = M$.)

Example 3.12. Consider the probability density

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

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.)

```
y <- rcauchy(100e3)
u <- runif(100e3)
ratio_pu_qu_M <- sin(y)^2*(1+y^2) / (y^2*2)
x <- y[u <= ratio_pu_qu_M]
```

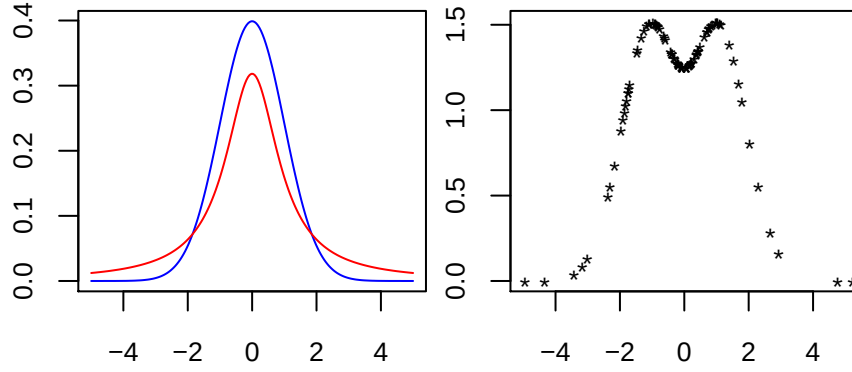


Figure 6: Some importance weights (right) when p is standard normal (blue) and q Cauchy (red).

4 Importance sampling

All methods up to this point have aimed at simulating i.i.d. random variables $X \stackrel{\text{i.i.d.}}{\sim} p$. It is possible to use an auxiliary distribution q for Monte Carlo integration like in 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 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 $f : \mathbb{X} \rightarrow \mathbb{R}$.

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

Define

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

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

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

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

Remark 4.2. The distribution q is called the *importance distribution* (sometimes also *instrumental* or *proposal*). 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*: $I_{p,q}^{(n)}(f) \xrightarrow{n \rightarrow \infty} \mathbb{E}_p[f(X)]$ (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 continuous case,

$$\mathbb{E}[Z_1] = \int_{q(y)>0} f(y) \frac{p(y)}{q(y)} q(y) dy = \int f(y) p(y) dy = \mathbb{E}_p[f(X)],$$

and similarly in the discrete case, changing the integral to a sum. The almost sure convergence (b) follows from the strong law of large numbers. $\square$

Remark 4.4 (*). Mathematically, importance sampling is a *change of measure*, and 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 $Y_1, \dots, Y_n \stackrel{\text{i.i.d.}}{\sim} \Gamma(a, b)$, 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

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

so the importance weights are given as

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

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

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

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 $Y_1, \dots, Y_n \stackrel{\text{i.i.d.}}{\sim} q$ 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*

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

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

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

$$\mathbb{E}[Z_1^2] = \int_{q>0} f^2(y) \frac{p^2(y)}{q^2(y)} q(y) dy = \int f^2(y) w(y) p(y) dy = \mathbb{E}_p[f^2(X)w(X)].$$

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

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*

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

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

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

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

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

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

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

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

Figure 7 shows an example simulation where $\text{Var}(I_{p,q}^{(n)}(f)) = \infty$. 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*

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

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

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

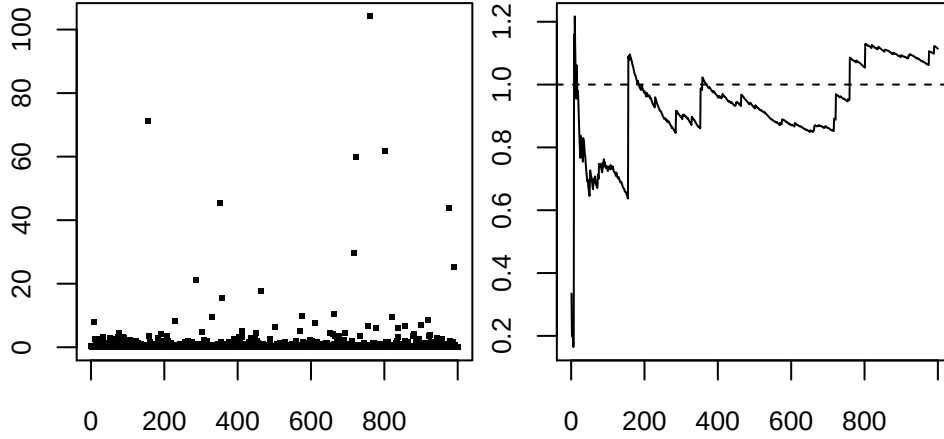


Figure 7: Example 4.10 with $a = 2$, $b = 2$ and $\beta = 2.5$, $\alpha = 0.5$ (NB $\alpha < a$) and $f(x) \equiv 1$. Values of the weights $w(Y_n)$ (left) and the sequence of estimates $I_{p,q}^{(n)}(f)$ (right) for $n = 1, 2, \dots, 1000$.

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 at most M times worse than the classical Monte Carlo estimate. Why this result is analogous to the case of rejection sampling?

If we are calculating simultaneously expectations of several functions f_i , it is a good rule of thumb to try to make sure that the weights are bounded (whenever $f_i(x) \neq 0$).

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

Proposition 4.13. *Suppose that $f : \mathbb{X} \rightarrow \mathbb{R}$ satisfies $\mathbb{E}_p[|f(X)|] \in (0, \infty)$. Then, the importance 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 continuous case, we have

$$\mathbb{E}_p[f^2(X)w_*(X)] = \int_{q_*(x)>0} f^2(x) \frac{p^2(x)}{q_*(x)} dx = (\mathbb{E}_p[|f(X)|])^2$$

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

$$(\mathbb{E}_p[|f(X)|])^2 = \left(\mathbb{E}_q[|f(X)|w(X)] \right)^2 \leq \mathbb{E}_q[f^2(X)w^2(X)] = \mathbb{E}_p[f^2(x)w(X)],$$

by Jensen's inequality. This implies $\sigma_{p,q_*}^2 \leq \sigma_{p,q}^2$ by Proposition 4.7. $\square$

Remark 4.14. The result of Proposition 4.13 is, of course, mostly theoretical. However, it serves as a 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 $w(u) \leq M$ 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 $X_1, \dots, X_n \stackrel{\text{i.i.d.}}{\sim} p$, 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 importance 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,

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

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,

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

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.

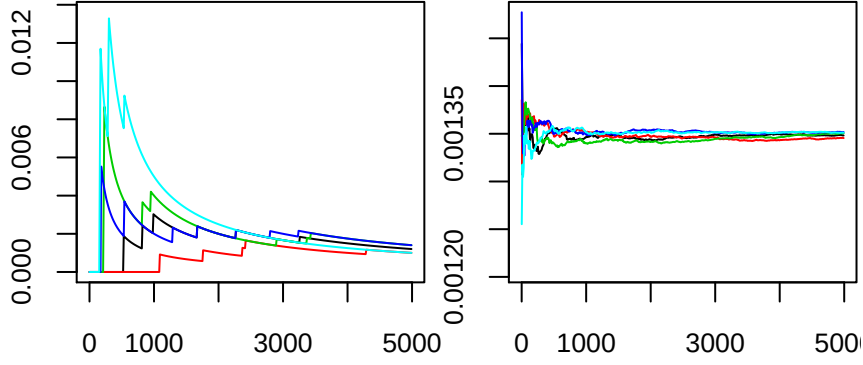


Figure 8: Five trajectories of classical Monte Carlo (left) and IS of Example 4.16 (right) with $x_0 = 3$. Number of samples in x -axis and value of estimate in y -axis.

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

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

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).

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

$$\boxed{\text{Assumption: } q(x) = 0 \implies p(x) = 0.} \quad (9)$$

Then, if $Y_1, Y_2, \dots \stackrel{\text{i.i.d.}}{\sim} q$,

$$\hat{I}_{p,q}^{(n)}(f) := \sum_{k=1}^n f(Y_k) W_k^{(n)}, \quad (10)$$

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

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

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

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

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

Theorem 4.19. *Suppose (9) holds. Then, $\hat{I}_{p,q}^{(n)}(f) \xrightarrow{n \rightarrow \infty} \mathbb{E}_p[f(X)]$ (almost surely).*

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

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

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

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 $I_{p,q}^{(n)}(1) \rightarrow 1$. This is more stringent than with unbiased IS, where we only need $q(x) = 0 \implies p(x)f(x) = 0$ which ensures $I_{p,q}^{(n)}(f) \rightarrow \mathbb{E}_p[f(X)]$.

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,

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

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 $\mathbb{E}_p[f(X)]$.

```

snis_gamma <- function(n, alpha, beta, f) {
  y <- rexp(n)                                # y ~ 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
}
# Use the function to estimate mean
snis_gamma(1000, alpha=2, beta=4, f=function(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 (9) holds and $\bar{\sigma}_{p,q}^2 := \mathbb{E}_p[w(X)\bar{f}^2(X)] < \infty$, where $\bar{f}(x) = f(x) - \mathbb{E}_p[f(X)]$.*

- (i) $\sqrt{n}(\hat{I}_{p,q}^{(n)}(f) - \mathbb{E}_p[f(X)]) \xrightarrow{n \rightarrow \infty} N(0, \bar{\sigma}_{p,q}^2)$ in distribution.
- (ii) *If also $\mathbb{E}_p[w(X)] < \infty$, then the following hold:*
 - $nv_{p,q}^{(n)} \xrightarrow{n \rightarrow \infty} \bar{\sigma}_{p,q}^2$ (a.s.), where $v_{p,q}^{(n)} := \sum_{k=1}^n (W_k^{(n)})^2 [f(Y_k) - \hat{I}_{p,q}^{(n)}(f)]^2$, and
 - $\mathbb{P}\left(\mathbb{E}_p[f(X)] \in \left[\hat{I}_{p,q}^{(n)} \pm \alpha \sqrt{v_{p,q}^{(n)}}\right]\right) \rightarrow 1 - 2\Phi(-\alpha)$ for any $\alpha \in (0, \infty)$.

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

Remark 4.24 (*). The quantity $n_{\text{eff}} = \left(\sum_{k=1}^n (W_k^{(n)})^2\right)^{-1} \in [1, n]$ is widely known as the *effective sample size* of (self-normalised) IS. This may be (loosely) justified by thinking of any function 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 $I_p^{(n)}(f)$ would have variance $\text{Var}_p(f(X))/n = (c/2)^2/n$, 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

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

so $\mathbb{E}_p[w(X)]/n \approx \text{Var}_p(f(X))/n_{\text{eff}}$ 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,

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

where $X \sim p$. The approximation is consistent assuming (9), in the following sense:

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

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

With unbiased IS, we have

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

Given (9) this is consistent and also unbiased $\mathbb{E}[\mu_{p,q}^{(n)}(A)] = \mathbb{P}(X \in A)$, but unlike self-normalised IS and plain MC, $\mu_{p,q}^{(n)}$ is not a probability distribution, because $\mu_{p,q}^{(n)}(\mathbb{X}) \neq 1$ 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 $\text{Var}(X) < \infty$, then*

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

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

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

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

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

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

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

However, if the conditional expectation $h(y) := \mathbb{E}_p[f(X, y) \mid Y = y]$ is available for all $y \in \mathbb{R}$ (that is, $\mathbb{E}_p[f(X, Y) \mid Y] = h(Y)$), we may use instead

$$I_{p,\text{RB}}^{(n)}(f) := \frac{1}{n} \sum_{k=1}^n h(Y_k), \tag{11}$$

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 distribution $p_Y(y) := \int p(x, y) dx$. We have a choice to simulate either $(X_k, Y_k)_{k \geq 1} \stackrel{\text{i.i.d.}}{\sim} p$ and throwing away X_k , or simulating $(Y_k) \stackrel{\text{i.i.d.}}{\sim} p_Y$, whichever is cheaper.

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* could be more appropriate, but Rao-Blackwellisation is widely used for historical reasons.

Remark 5.7 (*). Sometimes, it may be useful to employ some 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 $Z_k^{(i)} \stackrel{\text{i.i.d.}}{\sim} p_i$ and $U_k \stackrel{\text{i.i.d.}}{\sim} U(0, 1)$:

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

where $N_1 \sim \text{Binom}(n, 1/2)$ 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 $\hat{I}_p^{(n/2, n/2)}(f)$, because it turns out that $\text{Var}(\hat{I}_p^{(n/2, n/2)}(f)) \leq \text{Var}(I_p^{(n)}(f))$. 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_i are distributions.

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

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

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

- (i) Unbiasedness: $\mathbb{E}[I_p^{(\ell_1, \dots, \ell_m)}(f)] = \mathbb{E}_p[f(X)]$.
- (ii) If $\ell_i = w_i n$ (proportional allocation), then

$$\text{Var}(I_p^{(\ell_1, \dots, \ell_m)}(f)) \leq \text{Var}(I_p^{(n)}(f)).$$

Proof. Unbiasedness (i) is direct, and

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

because $\ell_i = w_i n$. If we write $X = \sum_{i=1}^m \mathbf{1}(s_{i-1} \leq U_i < s_i) X_i$, where $s_0 = 0$ and $s_i = \sum_{k=1}^i w_k$ and $X_i \sim p_i$, we notice that

$$\sum_{i=1}^m w_i \text{Var}_{p_i}(f(X)) = \mathbb{E}[\text{Var}(f(X) \mid U)] \leq \text{Var}_p(f(X)). \quad \square$$

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

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

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

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

$$u(x) := \mathbf{1}(0 < x \leq 1) = \sum_{k=1}^n w_k \tilde{u}_k(x),$$

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

Remark 5.11 (*). Proportional allocation is guaranteed to provide better (not worse) estimates, but optimal allocation strategy would be $\ell_i \propto w_i \sqrt{\text{Var}_{p_i}(f(X))}$. Because this depends on f and we may be interested in several f , and because $\text{Var}_{p_i}(f(X))$ 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 $(X_n, Y_n)_{n \geq 1} \stackrel{\text{i.i.d.}}{\sim} \hat{p}$, where the joint density $\hat{p}(x, y)$ admits p as marginal densities⁷, that is, for all $x, y \in \mathbb{X}$,

$$p(x) = \int \hat{p}(x, z) dz \quad \text{and} \quad p(y) = \int \hat{p}(z, y) dz.$$

That is, $(X_n)_{n \geq 1} \stackrel{\text{i.i.d.}}{\sim} p$ and $(Y_n)_{n \geq 1} \stackrel{\text{i.i.d.}}{\sim} p$, but X_k and Y_k may be dependent. The antithetic variable estimator

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

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

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

$$\text{Var}(I_{\hat{p}, \text{anti}}^{(n)}(f)) = \frac{1}{2n} [\text{Var}_p f(X) + \text{Cov}(f(X_1), f(Y_1))].$$

Therefore, if $\text{Cov}(f(X_1), f(Y_1)) < 0$ then $\text{Var}(I_{\hat{p}, \text{anti}}^{(n)}(f)) \leq \text{Var}(I_p^{(2n)}(f))$.

Note that $I_p^{(2n)}(f)$ has the same total number of samples as $I_{\hat{p}, \text{anti}}^{(n)}(f)$, so they are often of equal 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 $f : \mathbb{R} \rightarrow \mathbb{R}$ is monotonic. Define $X_k = F^{-1}(U_k)$ and $Y_k = F^{-1}(1 - U_k)$ where $(U_k)_{k \geq 0} \stackrel{\text{i.i.d.}}{\sim} \mathcal{U}(0, 1)$. Then, $\text{Cov}(f(X_1), f(Y_1)) \leq 0$.*

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 $\text{Var}(g(U)) = 0$, the claim is trivial, so assume $\text{Var}(g(U)) > 0$.

Because of symmetry

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

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

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

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

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

5.4 Control variates (*)

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

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

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

Proposition 5.16. *We have the expression of the variance*

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

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

Remark 5.17. Theoretically, the best value is

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

which leads into

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

Remark 5.18. The value β_* is often unavailable, and it is possible to choose a bad value for β which in fact increases variance! Introducing such controlled dependence allowing reduction of variance can be done in practice only in some specific scenarios.

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 $X_1, X_2, \dots$ 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. 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 $x_0, \dots, x_k \in \mathbb{S}$,

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

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

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

The initial distribution λ of $(X_k)_{k \geq 0}$ is distribution of X_0 , that is, $\mathbb{P}(X_0 = x) = \lambda(x)$ 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 $\pi = (\pi(x))_{x \in \mathbb{S}}$ is a p.m.f. on $\mathbb{S}$ taken as a row vector, and if

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

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

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 $f : \mathbb{S} \rightarrow \mathbb{R}$ such that $\mathbb{E}_\pi[f(X)]$ is finite,

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

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 $S_0, \dots, S_{m-1}$ of $\mathbb{S}$, where S_k are non-empty, such that

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

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,

$$\mathbb{P}(X_n = x) \xrightarrow{n \rightarrow \infty} \pi(x) \quad \text{for any } x \in \mathbb{S}.$$

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 this usually holds true as well. However, 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

$$\pi(x)P(x, y) = \pi(y)P(y, x) \quad \text{for all } x, y \in \mathbb{S}, \quad (12)$$

then P is *reversible with respect to π* , or π -reversible. (Condition (12) is also called 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 $\boxed{X_0 \sim \pi}$. 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, $(X_0, X_1) \stackrel{d}{=} (X_1, X_0)$. This generalises to

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

which can be understood so that the (stationary) Markov chain can be ‘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 $k = 1, 2, \dots$

- (a) Generate $Y_k \sim q(X_{k-1}, \cdot)$.
- (b) Generate $U_k \sim U(0, 1)$, and if $U_k \leq \alpha(X_{k-1}, Y_k)$ *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:

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

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

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

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

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

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

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

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 $U_k < \exp(r_u(X_{k-1}, Y_k))$ 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*

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

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, $X_0 = x_0 \in \mathbb{S}$. $\square$

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

Proof. Exercise. $\square$

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 $\mathbb{S} = \text{supp}(p)$, then for any function $f : \mathbb{X} \rightarrow \mathbb{R}$ with $\mathbb{E}_p[f(X)]$ finite*

$$I_{p,q,\text{MH}}^{(n)}(f) = \frac{1}{n} \sum_{k=1}^n f(X_k) \xrightarrow{n \rightarrow \infty} \mathbb{E}_p[f(X)] \quad (\text{almost surely}).$$

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 $x \in \mathbb{X} := \{1, \dots, m\}$ 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 $Y_k \sim U(\mathbb{X})$ 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 $x, y \in \mathbb{X}$,

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

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 $k = 1, \dots, n$ do

- (a) Simulate $Y_k \sim U\{1, 2, \dots, m\}$.
- (b) Simulate $U_k \sim U(0, 1)$ and if

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

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

```
m <- 30; n <- 10e3; X <- numeric(n); X[1] <- 1
for (k in 2:n) {
  x <- X[k-1]
  y <- ceiling(m*runif(1)) # y~U{1,2,...,m}
  if (runif(1) < y/x) {
    X[k] <- y
  } else {
    X[k] <- x
  }
}
```

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 $p(x) = 0$. IMH is always aperiodic.

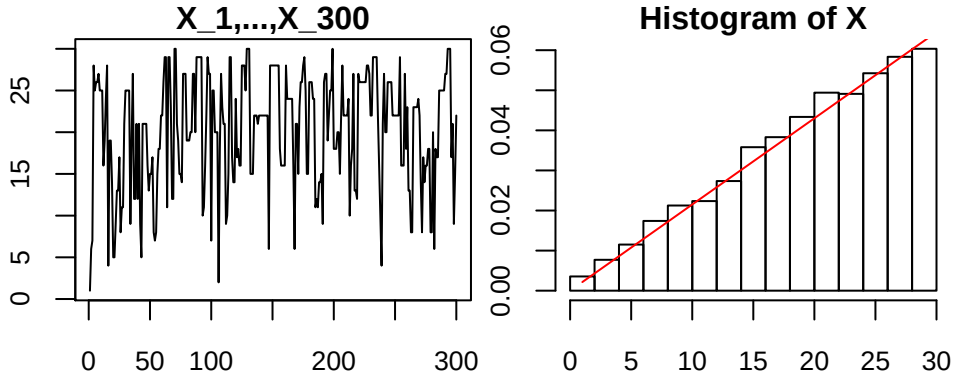


Figure 9: Left: x -axis is Markov chain step counter $k = 1, 2, \dots, 300$ and y -axis is Markov chain state X_k . Right: histogram of $X_1, X_2, \dots, X_n$ for $n = 10,000$ along with p .

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 6.21. 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

$$\mathbb{P}(X_n \in A \mid X_{n-1} = x) =: K(x, A) = \int_A k(x, y) dy + \rho(x) \mathbf{1}(x \in A), \quad (13)$$

where $k(x, y) := q(x, y)\alpha(x, y)$ is a *sub-probability density* for each $x \in \mathbb{X}$ and $\rho(x) = 1 - \int k(x, y) dy$ 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 [17] by Nummelin contains a minimal self-contained proofs about the strong law of large numbers and more.

Definition 6.22. 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, $\mathbb{P}(X_0 \in A, X_1 \in B) = \mathbb{P}(X_0 \in B, X_1 \in A)$.

Definition 6.23. Markov transition probability defined as in (13) satisfies detailed balance with respect to a p.d.f. p on $\mathbb{X}$ if

$$p(x)k(x, y) = p(y)k(y, x) \quad \text{for all } x, y \in \mathbb{X}.$$

Detailed balance is equivalent⁸ to reversibility with transition probabilities of the form (13). This is identical to the definition of reversibility in the discrete case for $x \neq y$, which turns out to be sufficient. The proof is identical as well.

8. To be precise, the continuous part $k(x, y)$ in the representation of (13) is unique only up to a set of measure zero. So the statement would be ‘there exists a k such that...’.

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 6.24 (π -irreducibility). Suppose that π is a p.d.f. on $\mathbb{S}$. The Markov chain $X_0, X_1, \dots$ is π -irreducible if for any $x \in \mathbb{S}$ and any set $A \subset \mathbb{S}$ such that $\int_A \pi(y)dy > 0$, 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 proof⁹

Theorem 6.25. *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*

$$I_{p,q,\text{MH}}^{(n)}(f) = \frac{1}{n} \sum_{k=1}^n f(X_k) \xrightarrow{n \rightarrow \infty} \mathbb{E}_p[f(X)] \quad (\text{almost surely}).$$

Note that Theorem 6.25 holds both when $\mathbb{X}$ is discrete or when $\mathbb{X} = \mathbb{R}^d$.

Example 6.26. 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 a new point uniformly at random in a window of length $2a$ centred at x . The proposal density is

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

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 $k = 1, \dots, n$:

- (a) Simulate $Z_k \sim U(-a, a)$ and set $Y_k = X_{k-1} + Z_k$.
- (b) Simulate $U_k \sim U(0, 1)$ and set

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

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

⁹ The proof follows, for example, from Corollary 2 of Tierney [25] along with Theorem 17.0.1 of Meyn and Tweedie [14]

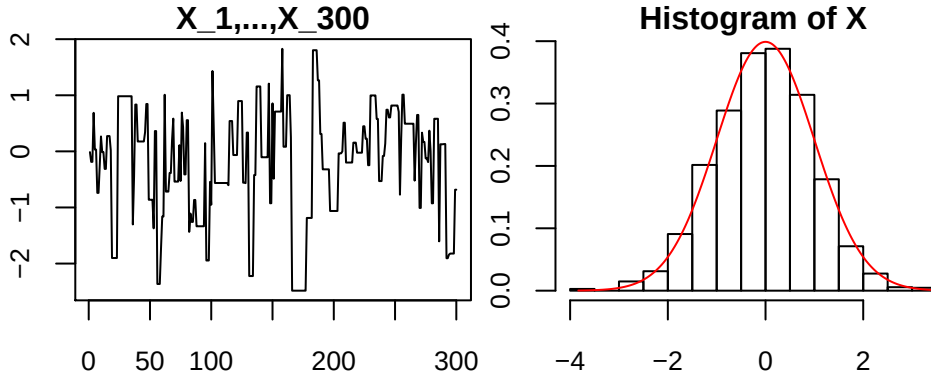


Figure 10: Simulation of Example 6.26: MCMC samples (left) and histogram approximation of the theoretical density (right).

```

a <- 3; n <- 10000; X <- numeric(n)
x <- 0; L_px <- -.5*x^2
for (k in 1:n) {
  y <- x + (2*runif(1)-1)*a
  L_py <- -.5*y^2      # NB L_px calculated only once!
  if (runif(1) < exp(L_py-L_px)) { x <- y; L_px <- L_py }
  X[k] <- x
}

```

Example 6.26 belongs to the following class of Metropolis-Hastings algorithms.

Definition 6.27. 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*

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

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

The symmetry 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 6.28 (Bivariate distribution with Gaussian random walk Metropolis).

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

and with $a = b = 1$.

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

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

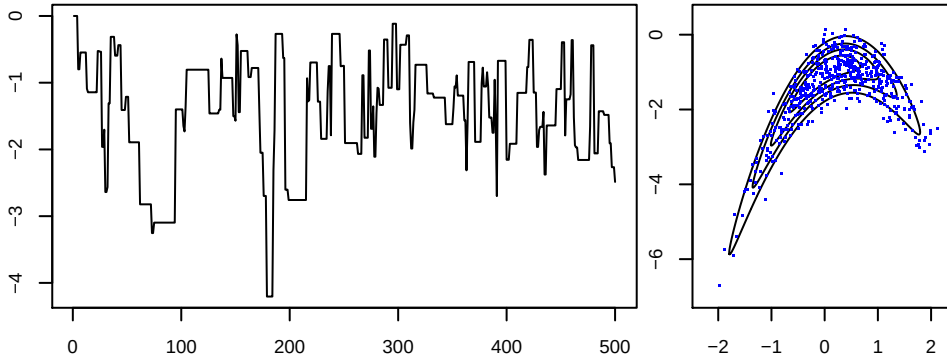


Figure 11: Simulation of Example 6.28: Second coordinate of the MCMC (left); The samples and the density contours (right).

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

```
log.n <- function(y) {      # N(0,[1 0.9; 0.9 1]) density
  z <- matrix(c(2.294,0,-2.065,1), nrow=2) %*% y; -0.5*sum(z^2)
}
log.p <- function(x) {
  a <- 1; b <- 1; y <- rbind(x[1]/a, x[2]*a + a*b*(x[1]^2 + a^2))
  log.n(y)
}
n <- 2000; X <- matrix(NA,2,n); x <- c(0,0); px <- log.p(x)
for (k in 1:n) {
  y <- x + rnorm(2); py <- log.p(y)                # Proposal
  if (runif(1) < exp(py-px)) { x <- y; px <- py } # Acc/rej
  X[,k] <- x                                       # Store
}
```

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

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

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

Remark 6.29. There are some theoretical optimality results determining which L is ‘the best’ when $\hat{q}$ is standard normal [e.g. 23].

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

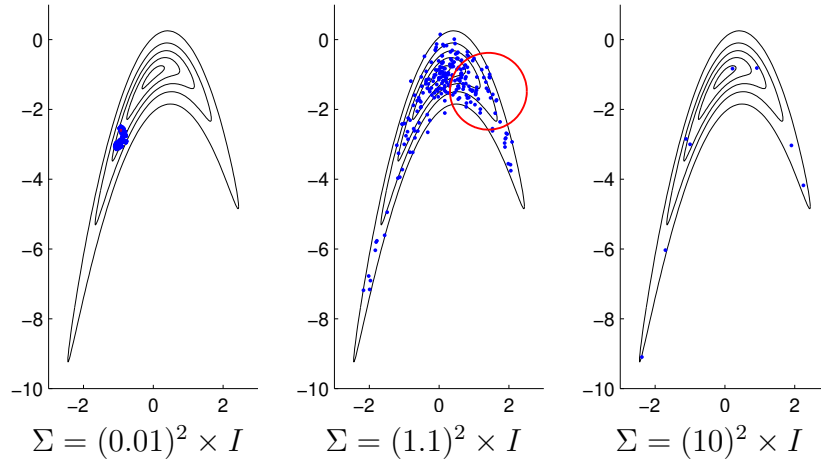


Figure 12: 1000 samples of the random walk Metropolis algorithm in $\mathbb{R}^2$ with $\tilde{q} = N(0, \Sigma)$. Contours of ‘banana-shaped’ p are shown in black.

- (b) Second rule of thumb: Set θ such that around 25% of the samples should be accepted on average.¹⁰

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

Because $\text{Cov}(p)$ is usually not available, $\text{Cov}(p)$ is often estimated by a ‘trial’ MCMC targeting p , and θ is found also by trial and error.

Remark 6.30. There are various *adaptive MCMC* algorithms which can be used to automatise this process, and learn L ‘progressively’ [e.g. 10, 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 6.31. Implementation of an adaptive MCMC which finds ‘good’ L automatically [27].

```
adapt.chol <- function(L, z, k, a, gam=2/3, a.opt=0.234) {
  sz2 <- sqrt(sum(z^2)); if (sz2==0) {sz2<-1}; D <- L %*% (z/sz2)
  t(chol(L %*% t(L) + (k+1)^(-gam)*(a-a.opt)*(D %*% t(D))))
}
adapt.mcmc <- function(log.p, x0, n) {
  d <- length(x0); x <- x0; p.x <- log.p(x); L <- diag(d)
  X <- matrix(nrow=d, ncol=n); acc <- 0 # Samples to X
  for (k in 1:n) {
    z <- rnorm(d); y <- x + L %*% z # Proposal
    p.y <- log.p(y); alpha <- min(1, exp(p.y-p.x)) # Acc.prob.
    if (runif(1) <= alpha) { x <- y; p.x <- p.y; acc <- acc+1 }
    X[,k] <- x
    L <- adapt.chol(L, z, k, alpha) # Update L
  }
  list(X=X, L=L, acc.rate=acc/n)
```

¹⁰. The theoretical value, 0.234, is optimal in high dimensions under very strong assumptions.

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

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

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

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

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

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

$$\begin{aligned} p_{-i}(x^{(-i)}) &:= \sum_{z \in \mathbb{Z}} p(x^{(1)}, \dots, x^{(i-1)}, z, x^{(i+1)}, \dots, x^{(d)}) \\ (p_{-i}(x^{(-i)}) &:= \int p(x^{(1)}, \dots, x^{(i-1)}, z, x^{(i+1)}, \dots, x^{(d)}) dz). \end{aligned}$$

Algorithm 6.32 ((Random scan) Metropolis-within-Gibbs). Suppose that $q_i(x^{(i)}, \cdot | x^{(-i)})$ determines a p.m.f. (p.d.f.) on $\mathbb{X}_1$ for each $x \in \mathbb{X}$ and for all $i = 1, \dots, d$. Choose some $X_0 \equiv x_0$ with $p(x_0) > 0$ and iterate for $k = 1, \dots, n$

- (a) Draw random coordinate index $I_k \sim \mathcal{U}\{1, \dots, d\}$.
- (b) Set $X_k^{(-I_k)} = X_{k-1}^{(-I_k)}$.
- (c) Simulate $Y_k^{(I_k)} \sim q_{I_k}(X_{k-1}^{(I_k)}, \cdot | X_{k-1}^{(-I_k)})$.
- (d) With probability $\alpha_{I_k}(X_{k-1}^{(I_k)}, Y_k^{(I_k)} | X_{k-1}^{(-I_k)})$ accept and set $X_k^{(I_k)} = Y_k^{(I_k)}$, otherwise set $X_k^{(I_k)} = X_{k-1}^{(I_k)}$, where

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

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

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

and in case $p(x)$ is defined as a product of terms, which do not all depend on the i :th coordinate, this may be further simplified...

Proposition 6.33. *Algorithm 6.32 is reversible with respect to p .*

Proof. (Discrete case) We may write the Markov transition in Algorithm 6.32 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 steps (b), (c) and (d) of Algorithm 6.32).

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

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

and then

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

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 $x^{(-i)} = y^{(-i)}$. $\square$

Remark 6.34. In fact, the proof of Proposition 6.33 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 may do full-dimensional Metropolis-Hastings and Metropolis-within-Gibbs simultaneously, by allowing the ‘choice indicator’ I_k to take value $d + 1$, and define K_{d+1} as the full-dimensional Metropolis-Hastings.

Definition 6.35. *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 6.36 (*). Algorithm 6.32 is valid in $\mathbb{X} = \mathbb{R}^d$. We cannot use Definition 6.23 of reversibility, but rather mean 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

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

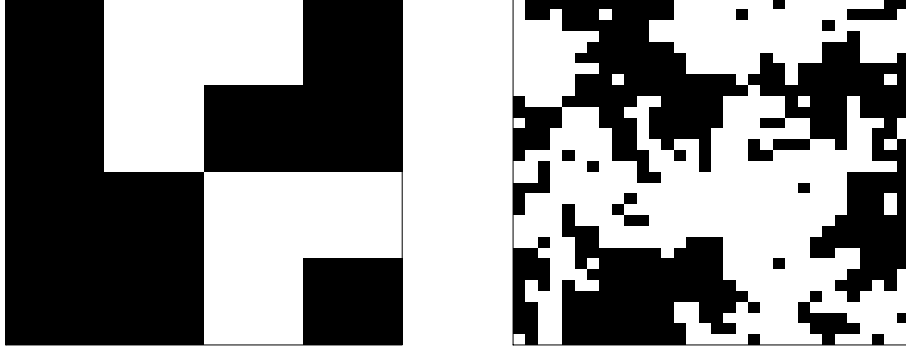


Figure 13: Left: Example 4-by-4 configuration with $\#x = 12$; right: realisation of the Ising model in $m = 32$, $\theta = 0.8$.

Example 6.37 (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.

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

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

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

- (a) Draw random indices $I_k \sim \mathcal{U}\{1, \dots, \ell\}$, $J_k \sim \mathcal{U}\{1, \dots, m\}$.
- (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

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

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 6.39. 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*.

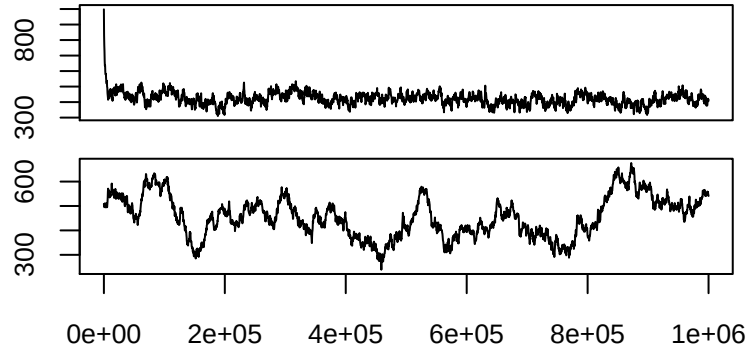


Figure 14: Trajectories of $f(X_k) = \#X_k$ (top) and $w(X_k)$ (bottom).

Remark 6.40. Note that in practice one should not re-calculate $\#(y, z^{(-i,j)})$ and $\#(x, z^{(-i,j)})$, but only their difference.

```
m <- 32; theta <- 0.8; n <- 20e3
hashdiff <- function(y, X, i, j, m) {
  i_ind <- pmax(pmin(c(i-1,i+1), m), 1) # Handle boundary
  j_ind <- pmax(pmin(c(j-1,j+1), m), 1) # indices
  Xneigh = c(X[i_ind,j], X[i,j_ind]) # Vector of neighbours
  hash_y <- sum(y != Xneigh)
  hash_x <- sum(X[i,j] != Xneigh)
  return(hash_y - hash_x)
}
X <- matrix(sample.int(2, size=m^2, replace=T), nrow=m)-1
for (k in 1:n) {
  i <- sample.int(m,1); j <- sample.int(m,1)
  y <- 1-X[i,j]
  if (runif(1) < exp(-theta*hashdiff(y, X, i, j, m))) {
    X[i,j] <- y
  }
}
```

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 $w(x) = \sum_{i,j} \mathbf{1}(x^{(i,j)} = 1)$, that is, the total number of white pixels. What is $\mathbb{E}_p[w(X)]$?

Example 6.41 (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

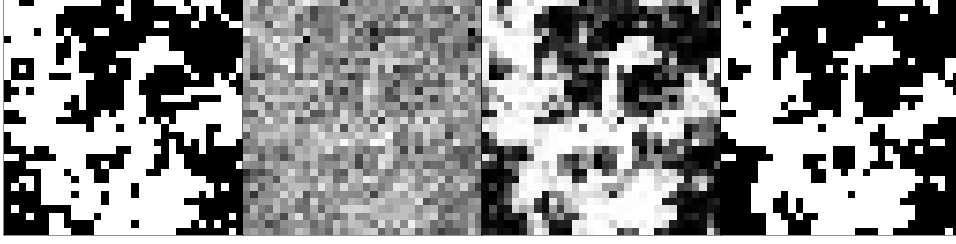


Figure 15: From left: Simulation of the Ising model X with $\theta = 0.8$; the noisy observations O of X with $\sigma^2 = 1$; the posterior mean approximation by MCMC; the MAP approximation by MCMC, with ten million samples.

pixel $i = 1, 2, \dots, m^2$ we observe

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

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

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

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

$$\log p_u(x) = -\frac{|x - o|^2}{2\sigma^2} - \theta \#x \quad \text{where} \quad |x - o|^2 = \sum_i (x_i - o_i)^2.$$

We will simulate $X_1, \dots, X_n \sim p$ with MCMC and use the samples to approximate the posterior mean and maximum a Posteriori (MAP) of each cell $i = 1, \dots, m^2$

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

In order to implement the MCMC, we can recycle the implementation in Example 6.38 only modifying the acceptance probability $\alpha(y \mid x)$ to incorporate $-|x - o|^2/(2\sigma^2)$ factor.

Variants of Metropolis-within-Gibbs

Algorithm 6.32 introduced earlier is only one way to implement (Metropolis-within-)Gibbs sampling, in terms how I_k are chosen.

Random scan means we choose I_k at random, as in Algorithm 6.32. It is customary to take $I_k \sim \mathcal{U}\{1, \dots, d\}$, but I_k can be chosen also from a non-uniform distribution over $\{1, \dots, d\}$.

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

$$K_1 K_2 \cdots K_d.$$

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

Random sweep is less common alternative, in which we choose a random orders (permutations of $\{1, \dots, d\}$).

Remark 6.42. 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 6.43. *Suppose that the Metropolis-within-Gibbs chain is p -irreducible and that starting from any $x \in \text{supp}(p)$, 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 6.25).*

Proof. Theorem 12 of [22] shows that the chain is Harris recurrent¹¹, and the SLLN is implied by [14, Theorem 17.0.1 (i)]. $\square$

Remark 6.44 ().* Theorem 6.43 adds one natural (and practically non-restrictive) condition over the irreducibility condition of Theorem 6.25, which only avoids some pathological scenarios (like if $x \in \text{supp}(p)$ but the conditionals are well-defined. . .).

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

Corollary 6.45. *Any p -irreducible Gibbs sampler satisfies the SLLN.*

We give next natural sufficient conditions which enable to check the p -irreducibility of a (Metropolis-within-)Gibbs.

Definition 6.46 (Positivity of p). The distribution p satisfies the *positivity condition* if the marginal distributions $p_i(x)$ satisfy for all $x \in \mathbb{R}$

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

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

Proposition 6.47. *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.*

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

About BUGS (*)

The BUGS (Bayesian inference Using Gibbs Sampling) software [24] is an implementation of Gibbs sampling (and sometimes also other Metropolis-within-Gibbs updates). It is a black-box inference engine, meaning that the user supplies only the model and the data, and the 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 ‘data’ in the BUGS model means only that some of the variables are ‘observed,’ and their values are fixed to the observed values. For instance, if the first two variables $X^{(1)} = x_*^{(1)}$ and $X^{(2)} = x_*^{(2)}$ were observed, and others not, then the MCMC targets the posterior distribution of $X^{(3:d)} \mid X_{(1:2)} = x_*^{(1:2)}$ which satisfies

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

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

6.7 About ‘gradient-informed’ proposals (*)

One way to construct proposal distributions $q(x, y)$ in the Metropolis-Hastings algorithm is to use $\nabla \log p(x)$ to ‘inform’ about the shape of p around x . The simplest such proposal is of the ‘Langevin’ type, where

$$Y_k = X_{k-1} + \frac{\tau}{2} \nabla \log p(X_{k-1}) + \sqrt{\tau} Z, \quad Z \sim N(0, \Sigma), \quad (14)$$

for some parameters $\tau \in (0, \infty)$ and covariance $\Sigma \in \mathbb{R}^{d \times d}$.¹² This algorithm is known as the *Metropolis adjusted Langevin* algorithm (MALA).

In recent years, a so-called *Hamiltonian Monte Carlo* (HMC) algorithm has gained attention. Its proposal is based on a physics-motivated continuous-time process (*Hamiltonian dynamics*) involving an auxiliary *momentum* random vector. More specifically, consider an auxiliary target $\tilde{p}(x, m) = p(x)q(m)$, where $q(m) = N(m; 0, \Sigma)$. The related ‘Hamiltonian’ can be written as

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

12. The proposal (14) is justified because it corresponds to an Euler discretisation of the (over-damped) Langevin diffusion of the form $dX_t = \frac{1}{2} \nabla \log p(X_t) + dB_t$, which admits p as its stationary distribution.

where $U(x) := -\log p(x)$ and $K(m) := \frac{1}{2}m^T \Sigma^{-1}m$ (up to a constant). The proposal is based on the following ODE:

$$\mathrm{d}x_t/\mathrm{d}t = \nabla U(x_t) \qquad \mathrm{d}m_t/\mathrm{d}t = \Sigma^{-1}m_t \qquad (15)$$

more specifically, drawing an independent $M_{k-1} \sim q$, and then a numerical approximation¹³ of (15) starting from $(x_0, m_0) = (X_{k-1}, M_{k-1})$, and setting $(\hat{X}_k, \hat{M}_k) \leftarrow (x_\tau, m_\tau)$. As the used integrator is required to satisfy reversibility¹⁴, the acceptance probability of $\hat{X}_k$ reduces to

$$\min \left\{ 1, \frac{\tilde{p}(\hat{X}_k, \hat{M}_k)}{\tilde{p}(X_{k-1}, M_{k-1})} \right\} = \min \left\{ 1, \exp \left(H(X_{k-1}, M_{k-1}) - H(\hat{X}_{k-1}, \hat{M}_{k-1}) \right) \right\}.$$

There exists an ergonomic implementation of HMC (based) algorithm in Stan [5]. It has a similar interface with BUGS, allowing to build model from blocks. Stan outperform BUGS in many interesting scenarios, but if you intend to use Stan, there are certain inherent restrictions that come with its ‘gradient-based’ algorithm, which are good to know:

- Discrete variables cannot be unknowns.
- Unknowns need to be (easily transformable) to $\mathbb{R}$ (or $\mathbb{R}^d$).¹⁵

It may also be worth pointing out, that HMC (like methods) may be sensitive to the tail behaviour of p , and even though they have showed great promise, they may not always provide a reliable outcome.¹⁶

7 MCMC convergence and mixing

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

- 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 can be significant for smaller k . This can induce significant *bias* to the estimator.
- It is not straightforward to assess the reliability of MCMC averages, due to the dependence of the random variables (X_k) . The dependence usually adds *variance* to the estimator, when compared against simple Monte Carlo averages.

13. Usually with so-called *leapfrog integrator*; see [15].

14. In the sense that starting from (x_0, m_0) we end up in (x_τ, m_τ) , and starting from $(x_\tau, -m_\tau)$ we get back to (x_0, m_0) , and because $K(m) = K(-m)$.

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

16. The HMC is still somewhat ‘immature’, in the sense that not much can be said about its behaviour in general. This is in contrast with Gibbs sampling and random-walk proposals, which are rather well understood by now (including their weaknesses).

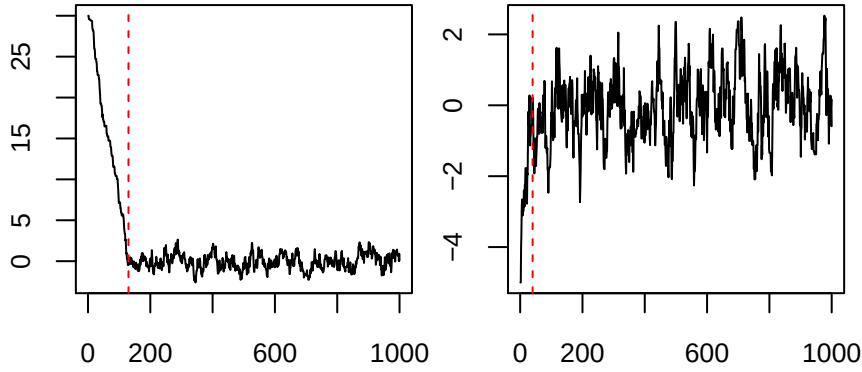


Figure 16: The first 1000 samples simulated from Example 6.26 with $a = 1$ and with $x_0 = 30$ (left) and $x_0 = -5$ (right). The red vertical line indicates the ‘burn-in time’.

7.1 Burn-in bias

MCMC Markov chain X_n converges in distribution to p as $n \rightarrow \infty$ (under an aperiodicity condition, cf. Theorem 6.7). The common practice with MCMC is to discard b first values of the Markov chain $X_0, \dots, X_b$, 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 $X_0, \dots, X_b$ 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*

$$\sqrt{n} [I_{p,q,\text{MH}}^{(n)}(f) - \mathbb{E}_p[f(X)]] \xrightarrow{n \rightarrow \infty} N(0, \sigma_{\text{MH}}^2),$$

with $\sigma_{\text{MH}}^2 = \lim_{n \rightarrow \infty} n \text{Var}(I_{p,\text{MH}}^{(n)}(f)) < \infty$.

Remark 7.3. The CLT does hold more generally, *under technical regularity conditions*. As there exist no single set of easily verifiable conditions, we shall not state more general forms of the CLT in detail.

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 $X_0, X_1, \dots$ be a stationary Markov chain, that is, $X_0 \sim \pi$, where π is the invariant distribution. Suppose $f : \mathbb{S} \rightarrow \mathbb{R}$ such that $\mathbb{E}_\pi[f^2(X)] < \infty$ and denote $Y_k = f(X_k)$.*

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

$$\lim_{n \rightarrow \infty} n \text{Var} \left(\frac{1}{n} \sum_{k=1}^n f(X_k) \right) = \text{Var}_\pi(f(X)) \left(1 + 2 \sum_{k=1}^{\infty} \rho_k \right).$$

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

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

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

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, $n^{-1} \sum_{k=1}^n k x_k \xrightarrow{n \rightarrow \infty} 0$.*

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

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

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

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

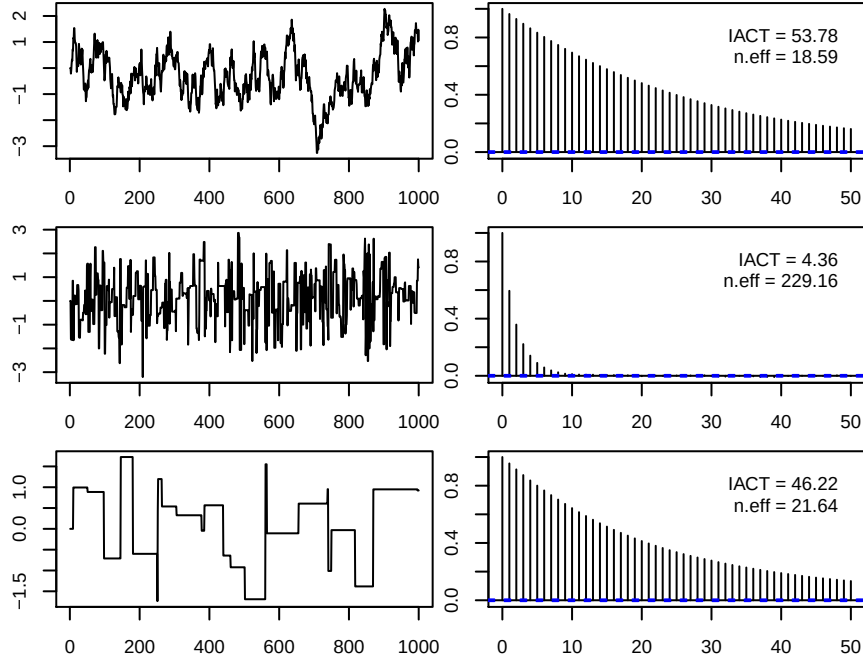


Figure 17: Sample paths and correlations of MH in Example 6.26 with $a = 0.5$ (top), $a = 5$ (middle) and $a = 50$ (bottom); here $f(x) := x$.

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

$$\text{Var}\left(\frac{1}{\lfloor n_{\text{eff}} \rfloor} \sum_{k=1}^{\lfloor n_{\text{eff}} \rfloor} f(Z_k)\right) = \frac{1}{\lfloor n_{\text{eff}} \rfloor} \text{Var}_p(f(X)).$$

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

Remark 7.9. Estimation of IACT and then n_{eff} can be done, for example, by summing some amount of sample autocorrelations, and truncating the sum at a chosen boundary. In R, if y contains $(Y_1, \dots, Y_n) = (f(X_1), \dots, f(X_n))$, you can use

```
> require(coda)
> effectiveSize(y)
```

to estimate the effective sample size.

Remark 7.10. Note that a MCMC sample $(X_k)_{k=1, \dots, n}$ 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 $f_1, \dots, f_m : \mathbb{X} \rightarrow \mathbb{R}$, you need to calculate $n_{\text{eff}}^{(1)}, \dots, n_{\text{eff}}^{(m)}$! 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 (`acf` in R).
- (iv) Calculate n_{eff} (either from ACF or with `coda`) and check that it is reasonably large. Use it to construct a CI:

$$\left[I_{p,q,\text{MH}}(f) \pm \alpha \frac{\hat{\sigma}_n}{\sqrt{n_{\text{eff}}}} \right],$$

where $\hat{\sigma}_n^2 := (n-1)^{-1} \sum_{k=1}^n [f(X_k) - I_{p,q,\text{MH}}(f)]^2 \xrightarrow{n \rightarrow \infty} \text{Var}_p(f(X))$ and α 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 [19], Tierney [26]). *Suppose that P and Q are transition probabilities both reversible wrt. common distribution π . Suppose that*

$$\sum_{x,y \in \mathbb{X}} \pi(x)P(x,y)[f(x) - f(y)]^2 \geq \sum_{x,y \in \mathbb{X}} \pi(x)Q(x,y)[f(x) - f(y)]^2, \quad (16)$$

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

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

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

$$P(x,y) \geq Q(x,y) \quad \text{for all } x, y \in \mathbb{X}, x \neq y, \quad (17)$$

implies (16). The condition (17) is referred to as the *off-diagonal order* or the *Peskun order* and (16) is known as the *covariance order*.

Remark 7.13 (*). In the continuous case, if P and Q are in the form (13) with $k_P(x,y)$ and $k_Q(x,y)$, respectively, then the covariance order (16) corresponds to

$$\iint \pi(x)k_P(x,y)[f(x) - f(y)]^2 dx dy \geq \iint \pi(x)k_Q(x,y)[f(x) - f(y)]^2 dx dy,$$

which holds if the analogous off-diagonal order holds:

$$k_P(x, y) \geq k_Q(x, y) \quad \text{for all } x, y \in \mathbb{X}, x \neq y.$$

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

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

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

Example 7.14. In the Ising model Example 6.38, 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 (17).

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

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

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, which implies that 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 an algorithm which operates on a *sequence* of distributions $\pi_1, \pi_2, \dots, \pi_T$, which gradually evolve towards the distribution of interest $p = \pi_T$. The algorithm can be viewed as producing a Monte Carlo approximation of each π_t , by a sequential update of the Monte Carlo samples that form an approximation of π_{t-1} . The samples are often called *particles* in this context, and the algorithm 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.

In this section, we denote for $a \leq b$ the vector $x_{a:b} = (x_a, \dots, x_b)$, with the convention that $x_{a:b}$ is omitted if $a > b$. We also exceptionally denote ‘time’ indices in subscript (not Monte Carlo samples as before), and superscript contain *sample indices* (not coordinates as before).

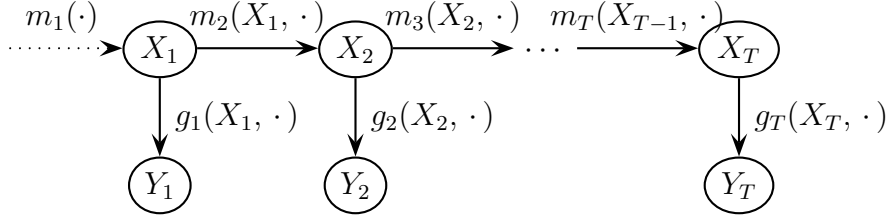


Figure 18: General state-space model.

8.1 General state-space models/hidden Markov models

Figure 18 illustrates a general state-space model. It consists of two parts:

- ‘Latent’ Markov chain $(X_t)_{t \geq 1}$ evolving in $\mathbb{S}$ with the (time-inhomogeneous) Markov dynamic $X_t | X_{t-1} \sim m_t(X_{t-1}, \cdot)$ and initial density $X_1 \sim m_1$.
- Conditionally independent observed process $(Y_t)_{t \geq 1}$ following the observation model $Y_t | X_t \sim g_t(X_t, \cdot)$.

More precisely, the model defines the joint density of the form $\prod_{t=1}^T m_t(x_{t-1}, x_t) g_t(x_t, y_t)$, with the convention $m_1(x_0, x_1) := m_1(x_1)$.

We are interested in Bayesian inference of $X_{1:T}$ having observed $Y_{1:T} = y_{1:T}$, that is,

$$p(x_{1:T}) \propto \prod_{t=1}^T m_t(x_{t-1}, x_t) g_t(x_t, y_t). \quad (18)$$

Example 8.1 (Noisy AR(1) process). Let $\sigma_1^2, \sigma_x^2, \sigma_y^2 \in (0, \infty)$ and $\rho \in \mathbb{R}$ be parameters. Then, let $m_1 = N(0, \sigma_1^2)$ and for $k \geq 2$, assume $(Z_k)_{k \geq 1}, (W_k)_{k \geq 1} \stackrel{\text{i.i.d.}}{\sim} N(0, 1)$, and define

$$\begin{aligned} X_k &:= \rho X_{k-1} + \sigma_x Z_k \\ Y_k &:= X_k + \sigma_y W_k. \end{aligned}$$

This corresponds to setting

$$\begin{aligned} m_k(x_{k-1}, x_k) &:= N(x_k; \rho x_{k-1}, \sigma_x^2) \\ g_k(x_k, y_k) &:= N(y_k; x_k, \sigma_y^2). \end{aligned}$$

In other words, $(X_k)_{k \geq 1}$ is an AR(1) process.¹⁷ Given a realisation of the process $(X_1, \dots, X_T)$, the observations are conditionally independent and perturbed by Gaussian increments with variance σ_y^2 . Figure 19 shows an example realisation of the process.

Remark 8.2. What we call state-space models (SSM), some other authors call *hidden Markov models* (HMM) [e.g. 4, 11]. 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.

17. Stationary iff $|\rho| < 1$ and $\sigma_1^2 = \frac{\sigma_x^2}{1-\rho^2}$.

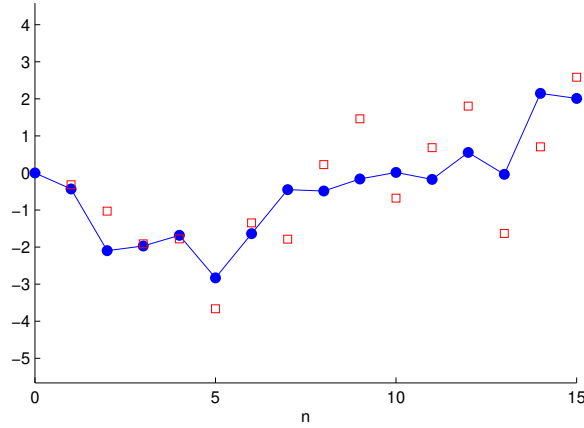


Figure 19: Sample path of the noisy AR(1) process in Example 8.1 with $\rho = 1$ and $\sigma_1^2 = \sigma_x^2 = 1 = \sigma_y^2$: The Markov chain $X_{1:15}$ in blue and the noisy observations $Y_{1:15}$ in red.

Remark 8.3. The generic methods such as IS and MCMC are, in theory, directly applicable in the SSM context. However, when T is large, the space $\mathbb{S}^T$ is high-dimensional, and ‘blind’ methods tend to work badly. The SMC can be much more efficient, because it is tailored to take advantage of the model structure.

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, exact inference is possible through the *forward-backward* algorithm.
- If $\mathbb{S} = \mathbb{R}^d$ 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.

8.2 First attempt: Sequential importance sampling

Before presenting the particle filter algorithm let us first consider the application of 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 some density, with support covering that of $p(x_{1:T})$, we could just draw $X_{1:T}^{(k)} \stackrel{\text{i.i.d.}}{\sim} q$ and approximate

$$\mathbb{E}_{\pi_T}[f(X_{1:T})] \approx \frac{\sum_{k=1}^n w_u(X_{1:T}^{(k)}) f(X_{1:T}^{(k)})}{\sum_{j=1}^n w_u(X_{1:T}^{(j)})}, \quad \text{where} \quad w_u(x_{1:T}) := \frac{p_T(x_{1:T}, y_{1:T})}{q(x_{1:T})}.$$

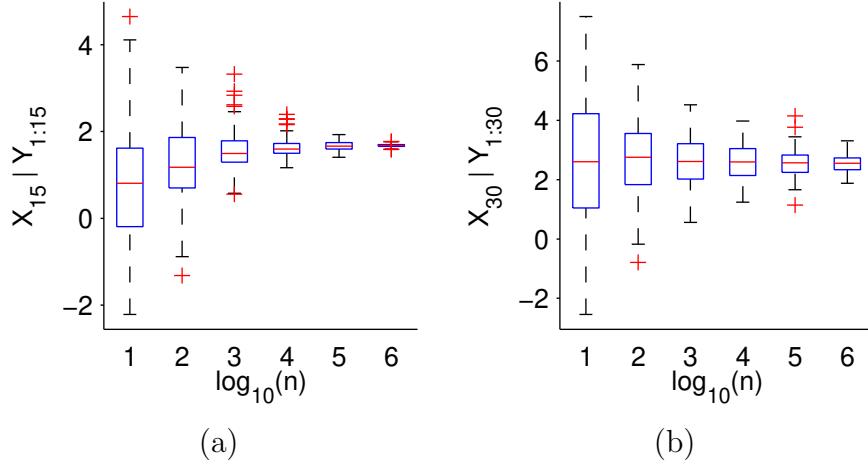


Figure 20: Box plot of estimates from Example 8.6 with up to one million samples, and 100 repetitions. (a) $\mathbb{E}[X_{15} | Y_{1:15} = y_{1:15}]$ (true value: 1.685) (b) $\mathbb{E}[X_{30} | Y_{1:30} = y_{1:30}]$ (true value: 2.508).

Remark 8.5. Note that the notation differs here from the notation in Section 4.3: we write the index of the sample in superscript.

Example 8.6 (Noisy AR(1) with prior as q). Consider Example 8.1 and let q be the prior of $X_{1:T}$, that is,

$$q(x_{1:T}) = \prod_{t=1}^T m_t(x_{t-1}, x_t).$$

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 20 and 21 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.

Example 8.7 (Noisy AR(1) with a ‘one-step optimal’ q). Consider Example 8.6, but choose now

$$q(x_{1:T}) = \prod_{t=1}^T q_t(x_t | x_{t-1}), \quad q_t(x_t | x_{t-1}) = N\left(x_t, \frac{x_{t-1} + y_t}{2}, \frac{1}{2}\right).$$

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 construct q_t based on $y_{1:t}$. It is direct to check that the unnormalised weights $w(z_{1:T})$ resulting from this choice are also bounded (exercise).

Figures 22 and 23 show simulation results corresponding Example 8.7.

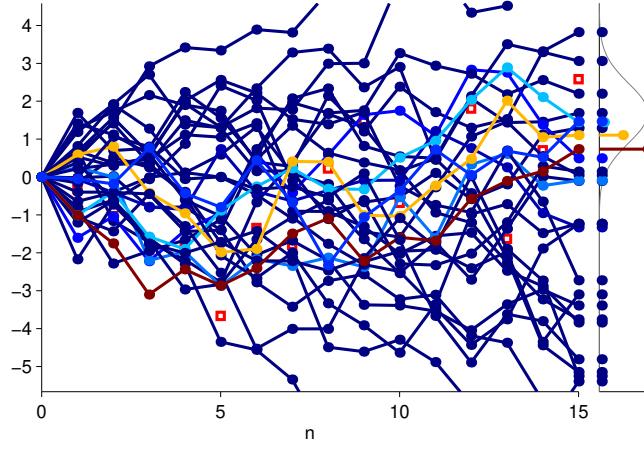


Figure 21: Some samples corresponding Example 8.6. Note that the weight distribution is very unequal. The true posterior density is shown on the right.

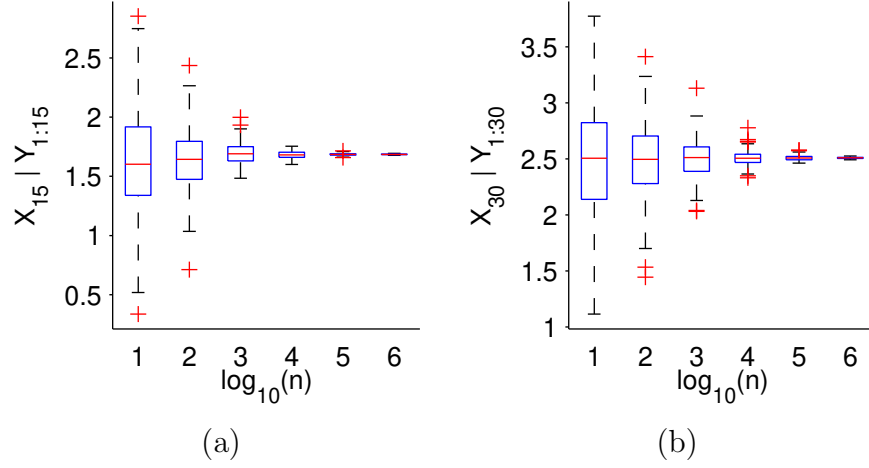


Figure 22: Box plot of estimates from Example 8.7; compare with 20.

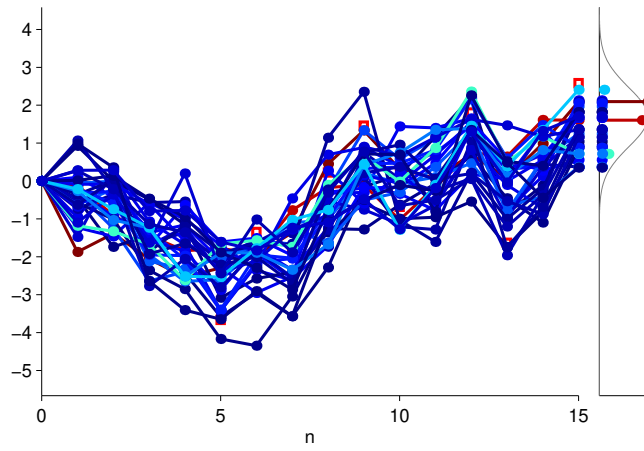


Figure 23: Some samples corresponding Example 8.7; compare with Figure 21.

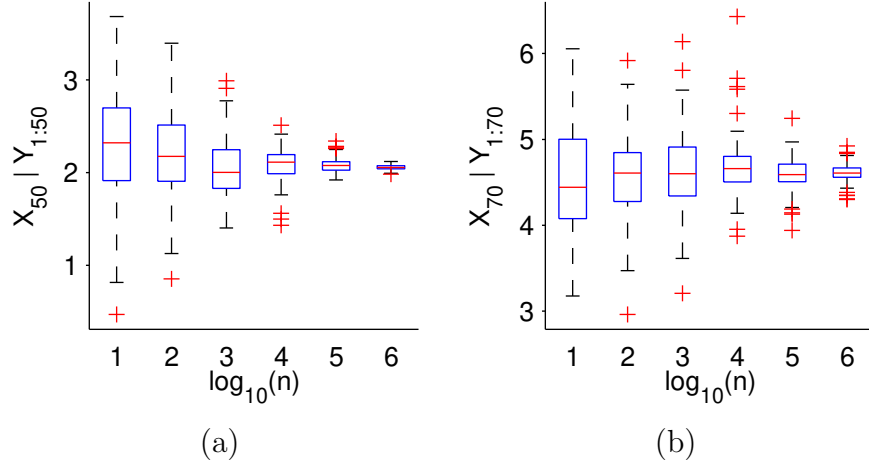


Figure 24: Box plot of estimates from Example 8.7 with $T = 50$ (true: 2.058) and $T = 70$ (true: 4.606).

Using a better proposal distribution in Example 8.7 improved significantly. It made reliable inference possible for up to $\mathbb{E}[X_{30} | Y_{1:30} = y_{1:30}]$ with around one million samples. This is achieved by better approximation of p by q , which shows in Figure 23 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 24. 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 SIS

Suppose now that $M_t(x_t | x_{1:t-1})$ 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:

$$\text{Assumption: } \prod_{t=1}^T M_t(x_t | x_{1:t-1}) G_t(x_{1:t}) \equiv \prod_{t=1}^T m_t(x_{t-1}, x_t) g_t(x_t, y_t). \quad (19)$$

Remark 8.8. Note that (19) is equivalent with $q(x_{1:T}) = \prod_{t=1}^T M_t(x_t | x_{1:t-1})$ satisfying the SNIS support condition (9) and G_t forming a *factorisation* of the unnormalised importance weight:

$$\prod_{t=1}^T G_t(x_{1:t}) = w_u(x_{1:T}) = \frac{\prod_{t=1}^T m_t(x_{t-1}, x_t) g_t(x_t, y_t)}{\prod_{t=1}^T M_t(x_t | x_{1:t-1})}, \quad \text{when } q(x_{1:T}) > 0.$$

Common choice of $G_t(x_{1:t}) = \frac{m_t(x_{t-1}, x_t) g_t(x_t, y_t)}{M_t(x_t | x_{1:t-1})}$ satisfies (19), but other choices may be preferable with the particle filter, which we introduce later [cf. 9].

Remark 8.9 ()*. The ingredients $M_{1:T}$ and $G_{1:T}$ form a so-called *Feynman-Kac* model [7].

Algorithm 8.10 (Sequential importance sampling). In each line of the algorithm, $i = 1, \dots, n$:

(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 $t = 2, \dots, T$, 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. Consider Algorithm 8.10, and let $t \in \{1:T\}$ such that $\int \prod_{u=1}^t M_u(x_u \mid x_{1:u-1}) G_u(x_{1:u}) dx_{1:T} < \infty$, and let $\pi_t(x_{1:t}) \propto \prod_{u=1}^t M_u(x_u \mid x_{1:u-1}) G_u(x'_{1:u})$ be a probability density. Then, denoting $V_t^{(i)} := \prod_{t'=1}^t \omega_{t'}^{(i)}$,

$$\frac{\sum_{i=1}^n V_t^{(i)} f(\mathbf{X}_t^{(i)})}{\sum_{j=1}^n V_t^{(j)}} \xrightarrow{n \rightarrow \infty} \mathbb{E}_{\pi_t}[f(X_{1:t})] \quad (\text{in probability}),$$

whenever the expectation is well-defined and finite.

Proof. This is self-normalised IS, because $\mathbf{X}_t \sim q_t(x_{1:t}) = \prod_{t=1}^T M_t(x_t \mid x_{1:t-1})$ and $V_t^{(i)} = w_u(\mathbf{X}_t) \propto \pi_t(\mathbf{X}_t)/q_t(\mathbf{X}_t)$. The result follows from Theorem 4.19. $\square$

Corollary 8.12. If assumption (19) holds, then the output of Algorithm 8.10 satisfies:

$$\text{SIS}_{M_{1:T}, G_{1:T}}^{(n)}(f) := \frac{\sum_{k=1}^n V^{(k)} f(\mathbf{X}^{(k)})}{\sum_{j=1}^n V^{(j)}} \xrightarrow{n \rightarrow \infty} \mathbb{E}_p[f(X_{1:T})] \quad (\text{in probability})$$

Proof. Direct application of Proposition 8.11, because $p = \pi_T$ and $V^{(i)} = V_T^{(i)}$. $\square$

Remark 8.13. When $\pi_1, \dots, \pi_T$ 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 probability*. We state the result using the latter, because we regard now Algorithm 8.15 to be run with a fixed n — and therefore ‘adding samples’ does not make sense, but the algorithm is just run with a higher $n \dots$

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 $A_{t-1}^{(i)}$ independently with $\mathbb{P}(A_{t-1}^{(i)} = j) = \bar{\omega}_{t-1}^{(j)}$.

(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 $V^{(j)} := \prod_{t=1}^T (\frac{1}{n}\omega_t^*)\bar{\omega}_T^{(j)}$ and $\mathbf{X}^{(j)} := \mathbf{X}_T^{(j)}$.
(In case $\omega_t^* = 0$, the algorithm is terminated with $V^{(i)} = 0$ and with arbitrary $\mathbf{X}^{(i)} \in \mathcal{S}^T$.)

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 does not sense to explicitly maintain $\mathbf{X}_t^{(i)}$, because $\omega_t^{(i)} = G_t(X_{t-1}^{(i)}, X_t)$ and $\mathbf{X}_T^{(i)}$ 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 $G_t(x_{1:t}) = G_t(x_t)$:

```
pf <- function(gen.M, log.G, n, TT) {
  A <- X <- matrix(nrow=n, ncol=TT)
  X[,1] <- x <- gen.M(NA, 1, n)
  nw <- norm.logw(log.G(x, 1)) # log(mean(omega)) & bar{omega}
  V_ <- nw$c; omega_ <- nw$w
  for (t in 2:TT) {
    A[,t-1] <- a <- sample(1:n, n, prob=omega_, replace=T)
    X[,t] <- x <- gen.M(x[a], t, n)
    nw <- norm.logw(log.G(x, t))
    V_ <- V_ + nw$c; omega_ <- nw$w
  }
  # Trace back the 'paths' \mathbf{X}_T^{(i)}
  a <- 1:n; for (t in rev(1:(TT-1))) {a <- A[a,t]; X[,t] <- X[a,t]}
  return(list(log.V=V_+log(omega_), X=X))
}
```

Application in Example 8.6, with an estimate for $E_p[X_T]$:

```
set.seed(42); TT <- 50; x0 <- 0; rho <- sigma.x <- sigma.y <- 1
y <- filter(rnorm(TT, sd=sigma.x), rho, method="r", init=x0) +
  rnorm(TT, sd=sigma.y)
log.G <- function(x, t) {dnorm(x, sd=sigma.y, mean=y[t], log=T)}
gen.M <- function(x, t, n) {
  if (t==1) {x <- x0}; rho*x + rnorm(n, sd=sigma.x)
}
norm.logw <- function(logw) {
  m<-max(logw); u<-exp(logw-m); list(c=m+log(mean(u)), w=u/sum(u))
}
out <- pf(gen.M, log.G, 100, TT)
ts.plot(t(out$X)); points(1:TT, y, pch="o", col="red")
m.T <- sum(norm.logw(out$log.V)$w * out$X[,TT])
```

Under (19), and under certain technical assumptions [cf. 7]:

$$\text{PF}_{M_{1:T}, G_{1:T}}^{(n)}(f) := \frac{\sum_{k=1}^n V^{(k)} f(\mathbf{X}^{(k)})}{\sum_{j=1}^n V^{(j)}} = \sum_{k=1}^n \bar{\omega}_T^{(i)} f(\mathbf{X}_T^{(k)}) \xrightarrow{n \rightarrow \infty} \mathbb{E}_p[f(X_{1:T})], \quad (20)$$

in probability.

Remark 8.18. While (20) holds quite generally, the estimator $\text{PF}_{M_{1:T}, G_{1:T}}^{(n)}(f)$ 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 SSM context, the conditional distributions of X_t given the observations $y_{1:t}$ 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, increasing n may not lead to useful results, but the PF may be iterated several times with fixed n — this is roughly what particle MCMC aims for.

Remark 8.19 ()*. The step (iii) in Algorithm 8.15 is called *resampling* or *selection*. Algorithm 8.15 was introduced for SSMs in [8], 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 in genetic algorithms, which aim for (global) optimisation.

In fact, the *multinomial resampling* (iii) may be replaced with another procedure drawing $A_{t-1}^{(1:n)}$, which remains *unbiased*:

$$\mathbb{E} \left[\frac{1}{n} \sum_{i=1}^n \mathbf{1} \left(A_{t-1}^{(i)} = j \right) \mid X_1^{(1:n)}, X_{t-1}^{(1:n)}, A_1^{(1:n)}, \dots, A_{t-2}^{(1:n)} \right] = \bar{\omega}_{t-1}^{(j)}.$$

For instance, the stratified resampling procedure of problem sheet 3, problem 3(a) 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 $\text{PF}_{M_{1:T}, G_{1:T}}^{(n)}(f)$, even though this is commonly observed empirically. . .)

Example 8.20. Let us revisit Example 8.6 with the PF. Figure 25 shows the samples produced by the particle filter. It is clear that the resampling helps to concentrate paths (compare with Figure 21). Figure 26 shows a summary of the filtering estimates, analogous to Figure 20, and Figure 27 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 better for bigger T . . .)

8.5 Unbiasedness of PF estimator

We shall not pursue a detailed proof of (20), but instead focus on the following *non-asymptotic* unbiasedness property of the PF [cf. 7, Theorem 7.4.2], which is key for particle MCMC [3] which we discuss later.

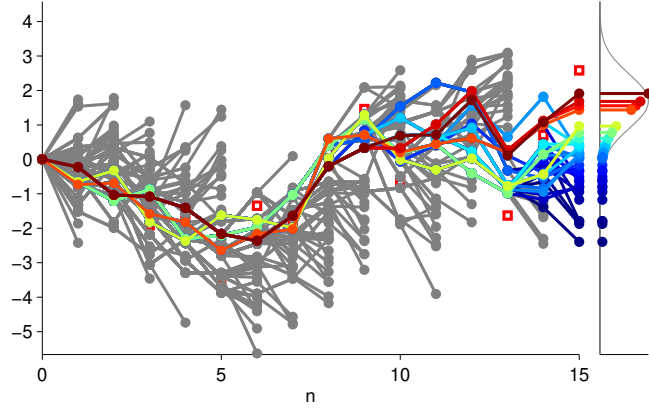


Figure 25: Some samples corresponding to the PF in Example 8.20. The grey paths show those ‘dead’ samples that were not selected in resampling.

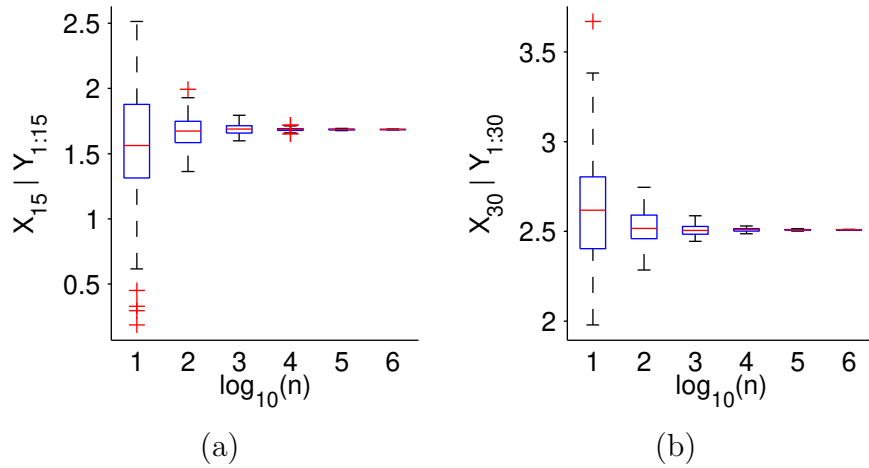


Figure 26: Box plot of the PF estimates with M_t corresponding to the prior, Example 8.6. Compare with 20. The estimates outperform also SIS with the ‘optimal’ proposal density in Example 8.7; see Figure 22.

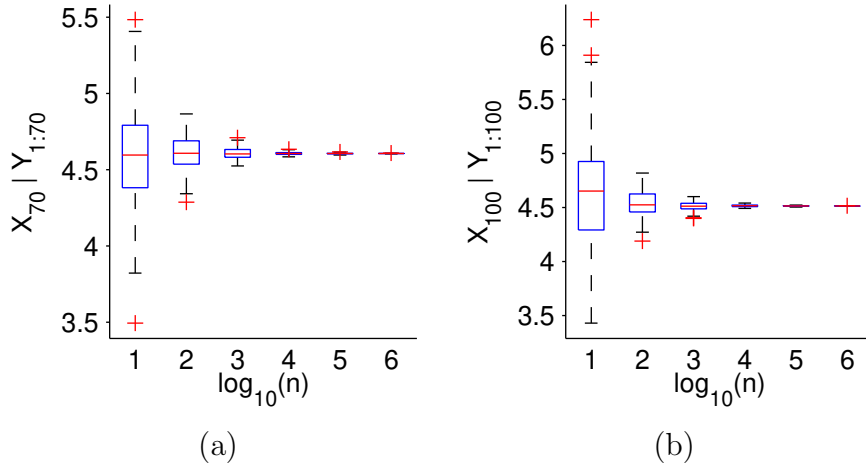


Figure 27: Box plot of the particle filter estimates with M_t corresponding to the prior, Compare with Figure 24. True value for $\mathbb{E}[X_{100} | Y_{1:100} = y_{1:100}] \approx 4.514$.

Theorem 8.21. *Under assumption (19), for any $f : \mathbb{S}^T \rightarrow \mathbb{R}$ with $\mathbb{E}_p[f(X)] < \infty$, and any $n \in \mathbb{N}$, the following holds for the output of Algorithm 8.15:*

$$\mathbb{E} \left[\sum_{k=1}^n V^{(k)} f(\mathbf{X}^{(k)}) \right] = \int \prod_{t=1}^T m_t(x_{t-1}, x_t) g_t(x_t, y_t) f(x_{1:T}) dx_{1:T}.$$

Proof. Define the functions $f_T(x_{1:T}) := f(x_{1:T})$, and for $t = T, \dots, 2$

$$f_{t-1}(x_{1:t-1}) := \int f_t(x_{1:t}) M_t(x_t | x_{1:t-1}) G_t(x_{1:t}) dx_t.$$

Assumption (19) implies that $f_0 := \int M_1(x_1) G_1(x_1) f_1(x_1) dx_1$ 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 $t = 2, \dots, T$ and $i \in \{1:n\}$,

$$\begin{aligned} & \mathbb{E}[\omega_t^* \bar{\omega}_t^{(i)} f_t(\mathbf{X}_t^{(i)}) | X_{1:t-1}^{(*)}, A_{1:t-2}^{(*)}] \\ &= \mathbb{E} \left[\mathbb{E}[\omega_t^{(i)} f_t(\mathbf{X}_{t-1}^{(A_{t-1}^{(i)})}, X_t^{(i)}) | X_{1:t-1}^{(*)}, A_{1:t-1}^{(*)}] \mid X_{1:t-1}^{(*)}, A_{1:t-2}^{(*)} \right] \\ &= \mathbb{E} \left[\int M_t(x_t | \mathbf{X}_{t-1}^{(A_{t-1}^{(i)})}) G_t(\mathbf{X}_{t-1}^{(A_{t-1}^{(i)})}, x_t) f_t(\mathbf{X}_{t-1}^{(A_{t-1}^{(i)})}, x_t) dx_t \mid X_{1:t-1}^{(*)}, A_{1:t-2}^{(*)} \right] \\ &= \sum_{j=1}^n \mathbb{P}(A_{t-1}^{(i)} = j | X_{1:t-1}^{(*)}, A_{1:t-2}^{(*)}) f_{t-1}(\mathbf{X}_{t-1}^{(j)}), \end{aligned}$$

so we may conclude that

$$\mathbb{E} \left[\left(\frac{1}{n} \omega_t^* \right) \sum_{i=1}^n \bar{\omega}_t^{(i)} f_t(\mathbf{X}_t^{(i)}) \mid X_{1:t-1}^{(*)}, A_{1:t-2}^{(*)} \right] = \sum_{j=1}^n \bar{\omega}_{t-1}^{(j)} f_{t-1}(\mathbf{X}_{t-1}^{(j)}). \quad (21)$$

We may apply (21) recursively with $t = T, \dots, 2$, yielding

$$\begin{aligned}\mathbb{E}\left[\sum_{k=1}^n V^{(k)} f(\mathbf{X}_T^{(k)})\right] &= \mathbb{E}\left[\left(\prod_{t=1}^{T-1} \frac{1}{n} \omega_t^*\right) \mathbb{E}\left[\left(\frac{1}{n} \omega_T^*\right) \sum_{i=1}^n \bar{\omega}_T^{(i)} f_T(\mathbf{X}_T^{(i)}) \mid X_{1:T-1}^{(*)}, A_{1:T-2}^{(*)}\right]\right] \\ &= \mathbb{E}\left[\left(\prod_{t=1}^{T-1} \frac{1}{n} \omega_t^*\right) \sum_{i=1}^n \bar{\omega}_{T-1}^{(i)} f_{T-1}(\mathbf{X}_{T-1}^{(i)})\right] = \dots \\ &= \frac{1}{n} \sum_{i=1}^n \mathbb{E}\left[\omega_1^* \bar{\omega}_1^{(i)} f_1(X_1^{(i)})\right],\end{aligned}$$

which equals to f_0 by a similar calculation as above. $\square$

The unbiasedness allows us to *combine* easily the output of independent particle filters, and deduce a consistent estimator as in self-normalised IS:

Corollary 8.22. *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*

$$(i) \quad E_{M_{1:T}, G_{1:T}}^{(N,n)}(f) := \frac{\sum_{i=1}^N \zeta_i(f)}{\sum_{j=1}^N \zeta_j(1)} \xrightarrow{N \rightarrow \infty} \mathbb{E}_p[f(X)] \text{ a.s.}$$

(ii) *If $\mathbb{E}[|\zeta(f) - \zeta(1)\mathbb{E}_p[f(X)]|^2 + |\zeta(1)|^2] < \infty$, then for any $\alpha \in (0, \infty)$,*

$$\begin{aligned}\mathbb{P}\left(\mathbb{E}_p[f(X)] \in \left[E_{M_{1:T}, G_{1:T}}^{(N,n)}(f) \pm \alpha \sqrt{\hat{v}^{(N,n)}}\right]\right) &\xrightarrow{N \rightarrow \infty} 1 - 2\Phi(-\alpha), \quad \text{where} \\ \hat{v}^{(N,n)} &:= \frac{\sum_{i=1}^N (\zeta_i(f) - \zeta_i(1) E_{M_{1:T}, G_{1:T}}^{(N,n)}(f))^2}{\left(\sum_{j=1}^N \zeta_j(1)\right)^2}.\end{aligned}$$

Proof. (i) follows from Theorem 8.21, 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 $N \rightarrow \infty$,

$$N \hat{v}^{(N,n)} = \frac{\frac{1}{N} \sum_{i=1}^N (\zeta_i(f) - \zeta_i(1) E_{M_{1:T}, G_{1:T}}^{(N,n)}(f))^2}{\left(\frac{1}{N} \sum_{j=1}^N \zeta_j(1)\right)^2} \rightarrow \frac{\mathbb{E}[(\zeta(f) - \zeta(1)\mathbb{E}_p[f(X)])^2]}{\mathbb{E}_p[\zeta(1)]^2}.$$

$\square$

Remark 8.23 (*). Suppose $\text{PF}_{M_{1:T}, G_{1:T}}^{(n,i)}(f)$ are independent realisations of $\text{PF}_{M_{1:T}, G_{1:T}}^{(n)}(f)$ in (20), then, unlike $E_{M_{1:T}, G_{1:T}}^{(N,n)}(f)$, the naive combination $\frac{1}{N} \sum_{i=1}^N \text{PF}_{M_{1:T}, G_{1:T}}^{(n,i)}(f)$ is *not* consistent, because $\mathbb{E}[\text{PF}_{M_{1:T}, G_{1:T}}^{(n)}(f)] \neq \mathbb{E}_p[f(X)]$ in general (even though, under general conditions, $\mathbb{E}[\text{PF}_{M_{1:T}, G_{1:T}}^{(n)}(f)] \rightarrow \mathbb{E}_p[f(X)]$ as $n \rightarrow \infty$). 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 a particle MCMC algorithm introduced in the seminal paper [3]. It is based on a combination of MCMC and particle filter, in a way that allows for Bayesian inference in a parameterised SSM.

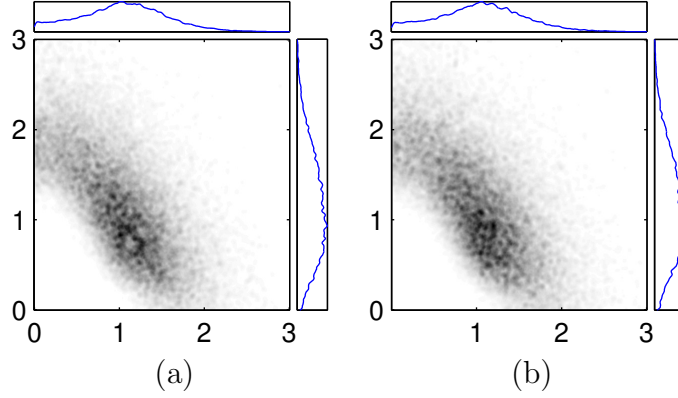


Figure 28: Density estimate using 100'000 samples from the marginal distribution of $\theta = (\sigma_x, \sigma_y)$, where $y_{1:15}$ correspond to Figure 19, and the model is from Example 8.1, with $\rho = 1$, $\sigma_1 = \sigma_x$ and a flat (improper) prior on θ . (a) Marginal (MCMC+Kalman) algorithm (b) PMMH with 1000 particles.

9.1 Bayesian state-space model

Suppose now that we have a family of SSMs, determined by a (hyper)parameter $\theta \in \mathbb{T}$:

$$p^{(\theta)}(x_{1:T}, y_{1:T}) = \prod_{t=1}^T m_t^{(\theta)}(x_{t-1}, x_t) g_t^{(\theta)}(x_t, y_t).$$

We also have a prior distribution $\text{pr}(\theta)$ of θ . The distribution of interest is the conditional distribution of $(\Theta, X_{1:T})$ given $Y_{1:T} = y_{1:T}$, which satisfies

$$p(\theta, x_{1:T}) \propto \text{pr}(\theta) p^{(\theta)}(x_{1:T}, y_{1:T}).$$

9.2 Particle marginal Metropolis-Hastings algorithm

In what follows, $n \in \mathbb{N}$ is fixed, and we have the following:

- $q(\theta, \theta')$ is a proposal distribution on $\mathbb{T}$.
- For each $\theta \in \mathbb{T}$, $M_{1:T}^{(\theta)}$ and $G_{1:T}^{(\theta)}$ satisfy assumption (19) with respect to $m_t^{(\theta)}$ and $g_t^{(\theta)}$.

Algorithm 9.1 (Particle marginal Metropolis-Hastings). Let $\Theta_0 \in \mathbb{T}$, and $V_0^{(i)} \in (0, \infty)$, $\mathbf{X}_0^{(i)} \in \mathbb{S}^T$ for $i \in 1:n$, and for $k = 1, 2, \dots, N$ iterate:

- Sample $\hat{\Theta}_k \sim q(\Theta_{k-1}, \cdot)$.
- Run Algorithm 8.15 with $(n, M_{1:T}^{(\hat{\Theta}_k)}, G_{1:T}^{(\hat{\Theta}_k)})$, and call its output $(\hat{V}_k^{(1:n)}, \hat{\mathbf{X}}_k^{(1:n)})$.
- With probability

$$\min \left\{ 1, \frac{\text{pr}(\hat{\Theta}_k) q(\hat{\Theta}_k, \Theta_{k-1}) \sum_{i=1}^n \hat{V}_k^{(i)}}{\text{pr}(\Theta_{k-1}) q(\Theta_{k-1}, \hat{\Theta}_k) \sum_{j=1}^n V_{k-1}^{(j)}} \right\},$$

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 $\mathbb{E}_p[f(X)]$:

$$I_{\text{PMMH}}^{(N,n)}(f) := \frac{1}{N} \sum_{k=1}^N \frac{\sum_{i=1}^n V_k^{(i)} f(\Theta_k, \mathbf{X}_k^{(i)})}{\sum_{j=1}^n V_k^{(j)}}.$$

Theorem 9.2. Suppose that $q(\theta, \theta') > 0$ for all $\theta, \theta' \in \mathbb{T}$. Then, for any fixed $n \in \mathbb{N}$,

$$I_{\text{PMMH}}^{(N,n)}(f) \xrightarrow{N \rightarrow \infty} \mathbb{E}_p[f(X)] \quad a.s.,$$

whenever the expectation is finite.

Proof. Let $Q_\theta(v^{(1:n)}, \mathbf{x}^{(1:n)})$ stand for the distribution[†] of $(V^{(1:n)}, \mathbf{X}^{(1:n)})$ of Algorithm 8.15 with $(n, M_{1:T}^{(\theta)}, G_{1:T}^{(\theta)})$, and define a distribution (sic!)

$$\pi(\theta, v^{(1:n)}, \mathbf{x}^{(1:n)}) := \frac{\text{pr}(\theta) Q_\theta(v^{(1:n)}, \mathbf{x}^{(1:n)}) \sum_{k=1}^n v^{(k)}}{\int \text{pr}(\hat{\theta}) Q_{\hat{\theta}}(\hat{v}^{(1:n)}, \hat{\mathbf{x}}^{(1:n)}) \left(\sum_{k=1}^n \hat{v}^{(k)} \right) d\hat{\theta} d\hat{v}^{(1:n)} d\hat{\mathbf{x}}^{(1:n)}}.$$

By Theorem 8.21, $\int Q_\theta(v^{(1:n)}, \mathbf{x}^{(1:n)}) \left(\sum_{k=1}^n v^{(k)} \right) dv^{(1:n)} = \int p^{(\theta)}(x_{1:T}, y_{1:T}) dx_{1:T}$, so indeed π is well-defined. Algorithm 9.1 turns out to be a Metropolis-Hastings with target π and proposal $\tilde{q}(\theta, v^{(1:n)}, \mathbf{x}^{(1:n)}; \hat{\theta}, \hat{v}^{(1:n)}, \hat{\mathbf{x}}^{(1:n)}) = q(\theta, \hat{\theta}) Q_{\hat{\theta}}(\hat{v}^{(1:n)}, \hat{\mathbf{x}}^{(1:n)})$, and by Theorem 8.21,

$$\mathbb{E}_\pi \left[\frac{\sum_{k=1}^n V^{(k)} f(\Theta, \mathbf{X}^{(k)})}{\sum_{j=1}^n V^{(j)}} \right] = \frac{\int \text{pr}(\theta) \left[\int p^{(\theta)}(x_{1:T}, y_{1:T}) f(\theta, x_{1:T}) dx_{1:T} \right] d\theta}{\int \text{pr}(\hat{\theta}) p^{(\hat{\theta})}(\hat{x}_{1:T}, y_{1:T}) d\hat{x}_{1:T} d\hat{\theta}},$$

which equals $\mathbb{E}_p[f(\Theta, X_{1:T})]$. The proof is complete once we believe that the Markov chain $(\Theta_i, V_i^{(1:n)}, \mathbf{X}_i^{(1:n)})_{i \geq 1}$ is Harris, which follows because it is π -irreducible Metropolis-Hastings. $\square$

Remark 9.3 (*). [†] The proof is written assuming $(V^{(1:n)}, \mathbf{X}^{(1:n)})$ have a density. This is not necessary, and the proof can be generalised directly for $Q_\theta(\cdot)$ being a probability measure instead.

Remark 9.4. The PMMH Algorithm 8.15 may be seen as an instance of a *pseudo-marginal* Metropolis-Hastings algorithm [1]. Similar algorithm was discovered in the physics literature slightly earlier [12].

9.3 Conditional particle filter/particle Gibbs (*)

The paper [3] contained also another scheme for PF inference, called the *conditional* PF (CPF) (or conditional SMC). It defines a Markov chain on $\mathbb{S}^T$, with transition defined essentially by the standard particle filter Algorithm 8.15, but with the one of the samples $X_{1:T}^{(1)}$ fixed to the previous state, and the state drawn amongst $\mathbf{X}^{(i)}$, with probability proportional to $V^{(i)}$. The Markov kernel defined by the CPF is p -reversible (where p is given in (18)).

Algorithm 9.5 (Conditional particle filter). Input $\mathbf{X}^* = X_{1:T}^*$:

- (i) Set $X_{1:T}^{(1)} := X_{1:T}^*$ and $\mathbf{X}_t^{(1)} := X_{1:t}^{(1)}$ for $t = 1:T$.
- (ii) Sample $X_1^{(2:n)} \stackrel{\text{i.i.d.}}{\sim} M_1$ and set $\mathbf{X}_1^{(2:n)} = X_1^{(2:n)}$.

(iii) 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 $A_{t-1}^{(2:n)}$ independently with $\mathbb{P}(A_{t-1}^{(i)} = j) = \bar{\omega}_{t-1}^{(j)}$, $j \in 1:n$.
 (iv) Sample $X_t^{(i)} \sim M_t(\cdot \mid \mathbf{X}_{t-1}^{(A_{t-1}^{(i)})})$ and set $\mathbf{X}_t^{(i)} = (\mathbf{X}_{t-1}^{(A_{t-1}^{(i)})}, X_t^{(i)})$ for $i = 2:n$.
 (v) Calculate $\omega_t^{(i)} := G_t(\mathbf{X}_t^{(i)})$ and set $\bar{\omega}_t^{(i)} := \omega_t^{(i)} / \omega_t^*$ where $\omega_t^* = \sum_{j=1}^n \omega_t^{(j)}$.
 Output $\mathbf{X}_T^{(I)}$ where $\mathbb{P}(I = j) = \bar{\omega}_T^{(j)}$, $j = 1:n$.

Remark 9.6. When $M_t(x_t \mid x_{1:t}) = M_t(x_t \mid x_{t-1})$ and $G_t(x_{1:t}) = G_t(x_{t-1:t})$, the CPF may be enhanced by applying it together with so-called *backward sampling* [28]. That is, instead of $\mathbf{X}_T^{(I)}$, the output is $(X_1^{(B_1)}, \dots, X_{T-1}^{(B_{T-1})}, X_T^{(B_T)})$, where $B_T = I$ and for $t = T-1, \dots, 1$:

$$\mathbb{P}(B_t = i \mid B_{t+1} = j) \propto \omega_t^{(i)} M_{t+1}(X_{t+1}^{(j)} \mid X_t^{(i)}) G_{t+1}(X_t^{(i)}, X_{t+1}^{(j)}). \quad (22)$$

Also the backward sampling version of CPF is p -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 (22).)

Remark 9.7. The authors of [3] suggest using the CPF algorithm in the Bayesian SSM context, as part of a Metropolis-within-Gibbs scheme, which alternates between sampling θ and $x_{1:T}$. The latter algorithm was called *particle Gibbs* in [3], but ‘particle Gibbs’ is also commonly used to refer to the CPF alone.

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] 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.
- [9] P. Guarniero, A. M. Johansen, and A. Lee. The iterated auxiliary particle filter. *J. Amer. Statist. Assoc.*, 112(520):1636–1647, 2017.
- [10] H. Haario, E. Saksman, and J. Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242, 2001.
- [11] A. M. Johansen, L. Evers, and N. Whiteley. Monte Carlo methods. Lecture notes, University of Bristol, 2010.

- [12] L. Lin, K. Liu, and J. Sloan. A noisy Monte Carlo algorithm. *Phys. Rev. D*, 61(7):074505, 2000.
- [13] 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.
- [14] S. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Cambridge University Press, second edition, 2009. ISBN 978-0-521-73182-9.
- [15] 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.
- [16] G. Nicholls. Part A Simulation and statistical programming. Lecture notes, University of Oxford, 2015.
- [17] E. Nummelin. MC’s for MCMC’ists. *Int. Statist. Rev.*, 70(2):215–240, 2002.
- [18] A. Penttinen. MATS442 Stokastinen simulointi. Lecture notes, University of Jyväskylä, 2010. (In Finnish).
- [19] P. H. Peskun. Optimum Monte-Carlo sampling using Markov chains. *Biometrika*, 60(3):607–612, 1973.
- [20] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, New York, 1999.
- [21] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, New York, second edition, 2004.
- [22] 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.
- [23] 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.
- [24] 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/>.
- [25] L. Tierney. Markov chains for exploring posterior distributions. *Ann. Statist.*, 22(4):1701–1728, 1994.
- [26] L. Tierney. A note on Metropolis-Hastings kernels for general state spaces. *Ann. Appl. Probab.*, 8(1):1–9, 1998.
- [27] M. Vihola. Robust adaptive Metropolis algorithm with coerced acceptance rate. *Statist. Comput.*, 22(5):997–1008, 2012.
- [28] N. Whiteley. Discussion on Particle Markov chain Monte Carlo methods. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 72(3):306–307, 2010.