Computer Intensive Statistical Methods

Martin Sköld

Lecture notes for FMS091/MAS221
October 2005, 2nd printing August 2006

Lund University

Centre for Mathematical Sciences
Mathematical Statistics

Centrum Scientiarum Mathematicarum

Computer Intensive Statistical Methods

Lecture notes for FMS091/MAS221.

Martin Sköld

Contents

Part I

Simulation and Monte-Carlo Integration

Chapter 1

Introduction

1.1 Scope of course

The term 'computer intensive methods' means different things to different people. It is also a dynamic subject: what requires intensive computing today may be solvable with a pocket calculator tomorrow. Not so long ago, the calculation of normal probabilities to reasonable accuracy would have required considerable CPU time. An initial classification of computer intensive methods as applied to statistics is the following:

1. Computers for graphical data exploration.
2. Computers for data modelling.
3. Computers for inference.

There is obviously some overlap in these three, but in this course I intend to focus mostly on the third of the above. That is, we shall aim at understanding computer techniques which require innovative algorithms to apply standard inferences.

I see two roles of this type of course. The first is to gain some understanding and knowledge of the techniques and tools which are available (so long as you've got the computing power). The second is that many of the techniques (if not all of them) are themselves clever applications or interpretations of the interplay between probability and statistics. So, understanding the principles behind the different algorithms can often lead to a better understanding of inference, probability and statistics generally. In short, the techniques are not just a means to an end. They have their own intrinsic value as statistical exercises.

This is not a course on computing itself. We won't get into the details of programming. Furthermore, this is not a course which will deal with specialised statistical packages often used in statistical computing. All the examples will be handled using simple Matlab functions - far from the most efficient way of implementing the various techniques. It is important to recognise that high-dimensional complex problems do require more efficient programming (commonly in C or Fortran). However the emphasis of this

course is to illustrate the various methods and their application on relatively simple examples. A basic familiarity with Matlab will be assumed.

1.2 Computers as inference machines

It is something of cliché to point out that computers have revolutionized all aspects of statistics. In the context of inference there have really been two substantial impacts: the first has been the freedom to make inferences without the catalogue of arbitrary (and often blatantly inappropriate) assumptions which standard techniques necessitate in order to obtain analytic solutions — Normality, linearity, independence etc. The second is the ability to apply standard type models to situations of greater data complexity — missing data, censored data, latent variable structures.

1.3 References

I've stolen quite heavily from the following books for this course:

- *Stochastic simulation*, B. Ripley.
- *An introduction to the bootstrap*, B. Efron and R. Tibshirani.
- *Tools for statistical inference*, M. Tanner.

By sticking closely to specific texts it should be easy to follow up any techniques that you want to look at in greater depth. Within this course I'll be concentrating very much on developing the techniques themselves rather than elaborating the mathematical and statistical niceties. You're recommended to do this for yourselves if interested.

1.4 Acknowledgement

Much of these notes are adapted from previous course notes of one form or another. Thanks are due especially to Stuart Coles.

Gareth Roberts, 2002

The original notes, used in various British universities, have been extensively revised and extended. Some parts and many examples remain untouched; they can probably be recognised by their varied and proper use of the English language. For the reader who would like a more in-depth coverage, the following advanced books are recommended:

- *Monte Carlo Statistical Methods, 2nd ed. (2005)* by C.P. Robert and G. Casella. This is the most up-to-date and extensive book on Monte Carlo methods available, with a focus on Markov-Chain Monte-Carlo and problems in Bayesian statistics. Does not cover bootstrap methods.

- *Bootstrap Methods and their Application, (1997)* by A.C. Davison and D.V. Hinkley. In-depth coverage of Bootstrap methods with a focus on applications rather than theory.

For the theoretical statistical background, the choice is more difficult. You might want to try *Statistical Inference, 2nd ed. (2001)*, by G. Casella and R.L. Berger, which is the book used in the course MAS207 and contains an introduction to both frequentist and Bayesian statistical theory.

Martin Sköld, 2005

Chapter 2

Simulation and Monte-Carlo integration

2.1 Introduction

In this chapter we look at different techniques for simulating from distributions and stochastic processes. Unlike all subsequent chapters we won't look much at applications here, but suffice it to say the applications of simulation are as varied as the subject of statistics itself. In any situation where we have a statistical model, simulating from that model generates realizations which can be analyzed as a means of understanding the properties of that model. In subsequent chapters we will see also how simulation can be used as a basic ingredient for a variety of approaches to inference.

2.2 Issues in simulation

Whatever the application, the role of simulation is to generate data which have (to all intents and purposes) the statistical properties of some specified model. This generates two questions:

1. How to do it; and
2. How to do it efficiently.

To some extent, just doing it is the priority, since many applications are sufficiently fast for even inefficient routines to be acceptably quick. On the other hand, efficient design of simulation can add insight into the statistical model itself, in addition to CPU savings. We'll illustrate the idea simply with a well–known example.

2.3 Buffon's Needle

We'll start with a simulation experiment which is intrinsically nothing to do with computers. Perhaps the most famous simulation experiment is

Buffon's needle, designed to calculate (not very efficiently) an estimate of π. There's nothing very sophisticated about this experiment, but for me I really like the 'mystique' of being able to trick nature into giving us an estimate of π. There are also a number of ways the experiment can be improved on to give better estimates which will highlight the general principle of *designing* simulated experiments to achieve optimal accuracy in the sense of minimizing statistical variability.

Buffon's original experiment is as follows. Imagine a grid of parallel lines of spacing d, on which we randomly drop a needle of length l, with $l \leq d$. We repeat this experiment n times, and count R, the number of times the needle intersects a line. Denoting $\rho = l/d$ and $\phi = 1/\pi$, an estimate of ϕ is

$$\hat{\phi}_0 = \frac{\hat{p}}{2\rho}$$

where $\hat{p} = R/n$.

Thus, $\hat{\pi}_0 = 1/\hat{\phi}_0 = 2\rho/\hat{p}$ estimates π.

The rationale behind this is that if we let x be the distance from the centre of the needle to the lower grid point, and θ be the angle with the horizontal, then under the assumption of random needle throwing, we'd have $x \sim U[0, d]$ and $\theta \sim U[0, \pi]$. Thus

$$\begin{aligned} p &= \Pr(\text{needle intersects grid}) \\ &= \frac{1}{\pi}\int_0^\pi \Pr(\text{needle intersects } |\theta = \phi) d\phi \\ &= \frac{1}{\pi}\int_0^\pi (\frac{2}{d} \times \frac{l}{2}\sin\phi) d\phi \\ &= \frac{2l}{\pi d} \end{aligned}$$

A natural question is how to optimise the relative sizes of l and d. To address this we need to consider the variability of the estimator $\hat{\phi}_0$. Now, $R \sim \text{Bin}(n, p)$, so $\text{Var}(\hat{p}) = p(1-p)/n$. Thus $\text{Var}(\hat{\phi}_0) = 2\rho\phi(1-2\rho\phi)/4\rho^2 n = \phi^2(1/2\rho\phi - 1)/n$ which is minimized (subject to $\rho \leq 1$) when $\rho = 1$. That is, we should set $l = d$ to optimize efficiency.

Then, $\hat{\phi}_0 = \frac{\hat{p}}{2}$, with $\text{Var}(\hat{\phi}_0) = (\phi/2 - \phi^2)/n$. It follows that $\text{Var}(\hat{\phi}) \approx \pi^4 \text{Var}(\hat{\phi}_0) \approx 5.63/n$.

Figure 2.1 gives 2 realisations of Buffon's experiment, based on 5000 simulations each. The figures together with an estimate `pihat` of π can be produced in Matlab by

```
pihat=buf(5000);
```

where the m-file `buf.m` contains the code

```
function pihat=buf(n);
x=rand(1,n);
theta=rand(1,n)*pi;
I=(cos(theta)>x);
```

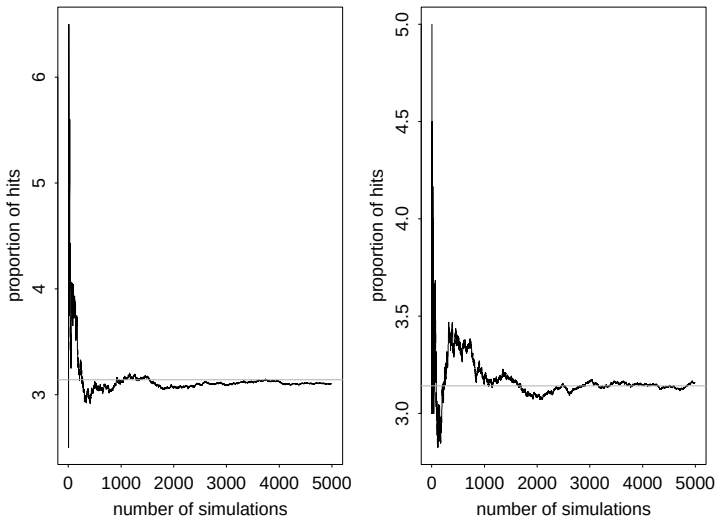

Figure 2.1: Two sequences of realisations of Buffon's experiment

```
R=cumsum(I);
phat=R./(1:n);
if R(n)>0
    plot(find(phat>0),1./phat(phat>0))
    xlabel('proportion of hits')
    ylabel('number of simulations')
    pihat=n/R(n);
else
    disp('no successes')
    pihat=0;
end
```

Thus I've used the computer to simulate the physical simulations. You might like to check why this code works.

There are a catalogue of modifications which you can use which might (or might not) improve the efficiency of this experiment. These include:

1. Using a grid of rectangles or squares (which is best?) and basing estimate on the number of intersections with either or both horizontal or vertical lines.

2. Using a cross instead of a needle.

3. Using a needle of length longer than the grid separation.

So, just to re–iterate, the point is that simulation can be used to answer interesting problems, but that careful design may be needed to achieve even moderate efficiency.

2.4 Raw ingredients

The raw material for any simulation exercise is random digits: transformation or other types of manipulation can then be applied to build simulations of more complex distributions or systems. So, how can random digits be generated?

It should be recognised that any algorithmic attempt to mimic randomness is just that: a mimic. By definition, if the sequence generated is deterministic then it isn't random. Thus, the trick is to use algorithms which generate sequences of numbers which would pass all the tests of randomness (from the required distribution or process) despite their deterministic derivation. The most common technique is to use a *congruential generator*. This generates a sequence of integers via the algorithm

$$x_i = ax_{i-1}(\text{mod } M) \tag{2.1}$$

for suitable choices of a and M. Dividing this sequence by M gives a sequence u_i which are regarded as realisations from the Uniform $U[0,1]$ distribution. Problems can arise by using inappropriate choices of a and M. We won't worry about this issue here, as any decent statistics package should have had its random number generator checked pretty thoroughly. The point worth remembering though is that computer generated random numbers aren't random at all, but that (hopefully) they look random enough for that not to matter.

In subsequent sections then, we'll take as axiomatic the fact that we can generate a sequence of numbers $u_1, u_2, \ldots, u_n$ which may be regarded as n independent realisations from the $U[0,1]$ distribution.

Chapter 3

Simulating from specified distributions

In this chapter we look at ways of simulating data from a specified distribution function F, on the basis of a simulated sample $u_1, u_2, \ldots, u_n$ from the distribution $U[0,1]$.

3.1 Transforming uniforms

We start with the case of constructing a draw x from a random variable $X \in \mathbf{R}$ with a continuous distribution F on the basis of a single u from $U[0,1]$. It is natural to try a simple transformation $x = h(u)$, but how should we choose h? Let's assume h is increasing with inverse $h^{-1} : \mathbf{R} \mapsto [0,1]$. The requirement is now that

$$\begin{aligned} F(v) &= P(X \leq v) = P(h(U) \leq v,) \\ &= P(h^{-1}(h(U)) \leq h^{-1}(v)) = P(U \leq h^{-1}(v)) \\ &= h^{-1}(v), \end{aligned}$$

for all $v \in \mathbf{R}$ and where in the last step we used that the distribution function of the $U[0,1]$ distribution equals $P(U \leq u) = u, u \in [0,1]$. The conclusion is clear, we should choose $h = F^{-1}$. If F is not one-to-one, as is the case for discrete random variables, the above argument remains valid if we define the inverse as

$$F^{-1}(u) = \inf\{x; F(x) \geq u\}. \tag{3.1}$$

The resulting algorithm for drawing from F is *the Inversion Method*:

Algorithm 3.1 (The Inversion Method)**.**

1. Draw u from $U[0,1]$.
2. $x = F^{-1}(u)$ can now be regarded a draw from F.

Figure 3.1 illustrates how this works. For example, to simulate from the

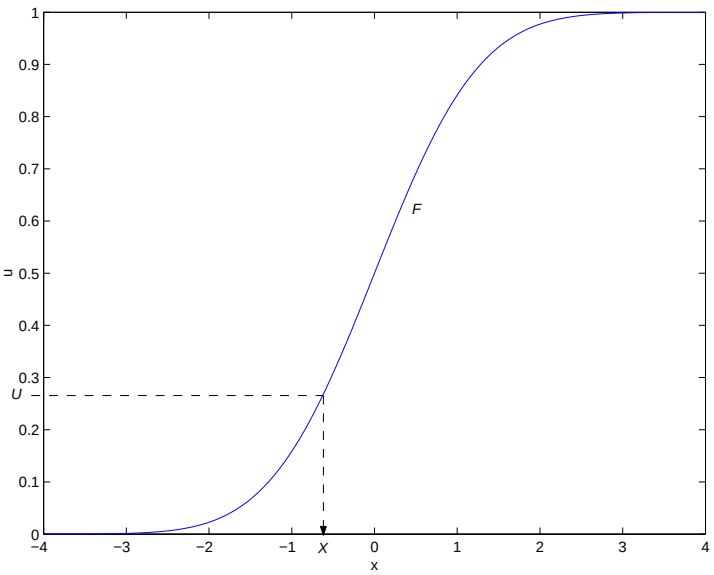

Figure 3.1: Simulation by inversion; the random variable $X = F^{-1}(U)$ will have distribution F if U is uniformly distributed on $[0, 1]$.

exponential distribution we have $F(x) = 1 - \exp(-\lambda x)$, so

$$F^{-1}(u) = -\lambda^{-1} \log(1 - u).$$

Thus with

```
u=rand(1,n);
x=-(log(1-u))/lambda;
```

we can simulate `n` independent values from the exponential distribution with parameter `lambda`. Figure 3.2 shows a histogram of 1000 standard ($\lambda = 1$) exponential variates simulated with this routine.

For discrete distributions, the procedure then simply amounts to searching through a table of the distribution function. For example, the distribution function of the Poisson(2) distribution is

```
x     F(x)
------------
0   0.1353353
1   0.4060058
2   0.6766764
3   0.8571235
4   0.9473470
5   0.9834364
6   0.9954662
```

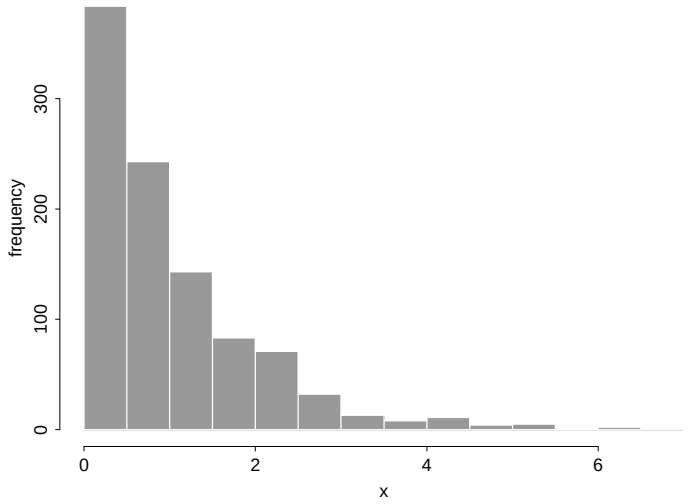

Figure 3.2: Histogram of 1000 simulated unit exponential variates

```
 7  0.9989033
 8  0.9997626
 9  0.9999535
10  0.9999917
```

so, we generate a sequence of standard uniforms $u_1, u_2, \ldots, u_n$ and for each u_i obtain a Poisson(2) variate x_i where $F(x_i - 1) < u_i \leq F(x_i)$. So, for example, if $u_1 = 0.7352$ then $x_1 = 3$.

The limitation on the efficiency of this procedure is due to the necessity of searching through the table, and there are various schemes to optimize this aspect.

Returning to the continuous case, it may seem that the inversion method is sufficiently universal to be the only method required. In fact, there are many situations in which the inversion method is either (or both) complicated to program or excessively inefficient to run. The inversion method is only really useful if the inverse distribution function is easy to program and compute. This is not the case, for example, with the Normal distribution function for which the inverse distribution function, Φ^{-1}, is not available analytically and slow to evaluate numerically. An even more serious limitation is that the method only applies for generating draws from univariate random variables. To deal with such cases, we turn to a variety of alternative schemes.

3.2 Transformation methods

The inversion method is a special case of more general transformation methods. The following theorem can be used to derive the distribution of $Z = h(X)$ for a more general class of real-valued random variables X.

Theorem 3.1 (Transformation theorem)**.** *Let $X \in \mathcal{X} \subseteq \mathbf{R}$ have a continuous density f and h a function with differentiable inverse $g = h^{-1}$. Then the random variable $Z = h(X) \in \mathbf{R}$ has density*

$$f(g(z))|g'(z)|, \tag{3.2}$$

for $z \in h(\mathcal{X})$ and zero elsewhere.

Proof. The proof is a direct application of the change-of-variable theorem for integrals. Note that two random variables X and Z have the same distribution iff $P(X \in A) = P(Z \in A)$ for all sets A. □

This device is used extensively in simulation, for example when we want to generate a $N(\mu, \sigma^2)$ variate y, it is common to first draw x from $N(0, 1)$ and then set $y = \sigma x + \mu$. Use Theorem 3.1 to show that this works. Sums of random variables can also be useful in creating new variables. Recall that

Theorem 3.2. *Let $X \in \mathbf{R}$ and $Y \in \mathbf{R}$ be independent with densities f and g respectively, then the density of $Z = X + Y$ equals $f * g(z) = \int f(t - z)g(t)\, dt$.*

This can be used to generate Gamma random variables. A random variable X has a Gamma$(a, 1)$ distribution if its density is proportional to $x^{a-1} \exp(-x)$, $x > 0$. Using Theorem 3.2 we can show that if X and Y are independent Gamma$(a, 1)$ and Gamma$(a', 1)$ respectively, then $Z = X + Y$ has a Gamma$(a + a', 1)$ distribution. Since Gamma$(1, 1)$ (i.e. Exponential(1)) variables are easily generated by inversion, a Gamma$(k, 1)$ variable Z, for integer values k, is generated by

$$z = \sum_{i=1}^{k} -\log(u_k) \tag{3.3}$$

using independent draws of uniforms $u_1, \ldots, u_n$. As an alternative we can use a combination of Theorems 3.1 and 3.2 to show that

$$z = \sum_{i=1}^{2k} x_i^2/2 \tag{3.4}$$

is a draw from the same distribution if $x_1, \ldots, x_{2k}$ are independent standard Normal draws.

Example 3.1 (The Box-Muller transformation)**.** This is a special trick to simulate from the Normal distribution. In fact it produces two independent variates in one go. Let u_1, u_2 be two independently sampled U$[0, 1]$ variables, then it can be shown that

$$x_1 = \sqrt{-2\log(u_2)}\cos(2\pi u_1) \text{ and } x_2 = \sqrt{-2\log(u_2)}\sin(2\pi u_1)$$

are two independent N$(0, 1)$ variables.

Below we give the multivariate version of Theorem 3.1.

Theorem 3.3 (Multivariate transformation theorem)**.** *Let $X \in \mathcal{X} \subseteq \mathbf{R}^d$ have a continuous density f and $h : \mathcal{X} \mapsto \mathbf{R}^d$ a function with differentiable inverse $g = h^{-1}$. Further write $J(z)$ for the determinant of the Jacobian matrix of $g = (g_1, \ldots, g_d)$,*

$$J(x) = \begin{vmatrix} dg_1(z)/dz_1 & \ldots & dg_1(z)/dz_d \\ \vdots & \ddots & \vdots \\ dg_d(z)/dz_1 & \ldots & dg_d(z)/dz_d \end{vmatrix}. \tag{3.5}$$

Then the random variable $Z = h(X) \in \mathbf{R}^d$ has density

$$f(g(z))|J(z)|, \tag{3.6}$$

for $z \in h(\mathcal{X})$ and zero elsewhere.

Example 3.2 (Choleski method for multivariate Normals)**.** The Choleski method is a convenient way to draw a vector z from the multivariate Normal distribution $N_n(0, \Sigma)$ based on a vector of n independent $N(0, 1)$ draws $(x_1, x_2, \ldots, x_n)$. Choleski decomposition is a method for computing a matrix C such that $CC^T = \Sigma$, in Matlab the command is `chol`. We will show that $z = Cx$ has the desired distribution. The density of X is $f(x) = (2\pi)^{-d/2} \exp(-x^T x/2)$ and the Jacobian of the inverse transformation, $x = C^{-1}z$, equals $J(z) = |C^{-1}| = |C|^{-1} = |\Sigma|^{-1/2}$. Hence, according to Theorem 3.3, the density of Z equals

$$\begin{aligned} f(C^{-1}z)|\Sigma|^{-1/2} &= (2\pi)^{-d/2} \exp(-(C^{-1}z)^T(C^{-1}z)/2)|\Sigma|^{-1/2} \\ &= (2\pi)^{-d/2} \exp(-z^T\Sigma^{-1}z/2)|\Sigma|^{-1/2}, \end{aligned}$$

which we recognise as the density of a $N_n(0, \Sigma)$ distribution. Of course, $z + \mu$, $\mu \in \mathbf{R}^d$ is a draw from $N_n(\mu, \Sigma)$.

3.3 Rejection sampling

The idea in rejection sampling is to simulate from one distribution which is easy to simulate from, but then to only accept that simulated value with some probability p. By choosing p correctly, we can ensure that the sequence of accepted simulated values are from the desired distribution.

The method is based on the following theorem:

Theorem 3.4. *Let f be the density function of a random variable on $\mathbf{R}^d$ and let $Z \in \mathbf{R}^{d+1}$ be a random variable that is uniformly distributed on the set $A = \{z; 0 \leq z_{d+1} \leq Mf(z_1, \ldots, z_d)\}$ for an arbitrary constant $M > 0$. Then the vector $(Z_1, \ldots, Z_d)$ has density f.*

Proof. First note that

$$\begin{aligned}\int_A dz &= \int_{\mathbf{R}^d} \Big(\int_0^{Mf(z_1,\ldots,z_d)} dz_{d+1} \Big)\, dz_1 \cdots dz_d \\ &= M \int f(z_1, \ldots, z_d)\, dz_1 \cdots dz_d = M.\end{aligned}$$

Hence, Z has density $1/M$ on A. Similarily, with $B \subseteq \mathbf{R}^d$, we have

$$\begin{aligned}P((Z_1, \ldots, Z_d) \in B) &= \int_{\{z; z\in A\} \cap \{z; (z_1,\ldots,z_d)\in B\}} M^{-1}\, dz \\ &= M^{-1} \int_B Mf(z_1, \ldots, z_d)\, dz_1 \cdots dz_d \\ &= \int_B f(z_1, \ldots, z_d)\, dz_1 \cdots dz_d,\end{aligned}$$

and this is exactly what we needed to show. □

The conclusion of the above theorem is that we can construct a draw from f by drawing uniformly on an appropriate set and then drop the last coordinate of the drawn vector. Note that the converse of the above theorem is also true, i.e. if we draw $(z_1, \ldots, z_d)$ from f and then z_{d+1} from $U(0, Mf(z_1, \ldots, z_d))$, $(z_1, \ldots, z_{d+1})$ will be a draw from the uniform distribution on $A = \{z; 0 \leq z_{d+1} \leq Mf(z_1, \ldots, z_d)\}$. The question is how to draw uniformly on A without having to draw from f (since this was our problem in the first place); the rejection method solves this by drawing uniformly on a larger set $B \supset A$ and rejecting the draws that end up in $B \backslash A$. A natural choice of B is $B = \{z; 0 \leq z_{d+1} \leq Kg(z_1, \ldots, z_d)\}$, where g is another density, the *proposal density*, that is easy to draw from and satisfies $Mf \leq Kg$.

Algorithm 3.2 (The Rejection Method)**.**

1. Draw $(z_1, \ldots, z_d)$ from a density g that satisfies $Mf \leq Kg$.
2. Draw z_{d+1} from $U(0, Kg(z_1, \ldots, z_d))$.
3. Repeat steps 1-2 until $z_{d+1} \leq Mf(z_1, \ldots, z_d)$.
4. $x = (z_1, \ldots, z_d)$ can now be regarded as a draw from f.

It might seem superflous to have two constants M and G in the algorithm. Indeed, the rejection method is usually presented with $M = 1$. We include M here to illustrate the fact that you only need to know the density up to a constant of proportionality (i.e. you know Mf but not M or f). This situation is very common, especially in applications to Bayesian statistics.

The efficiency of the rejection method depends on how many points are rejected, which in turn depends on how close Kg is to Mf. The probability of accepting a particular draw $(z_1, \ldots, z_d)$ from g equals

$$\begin{aligned} &P(Z_{d+1} \le Mf(Z_1, \ldots, Z_d)) \\ &\quad = \int \Big(\int_0^{Mf(z_1,\ldots,z_d)} (Kg(z_1, \ldots, z_d))^{-1} \, dz_{d+1} \Big) g(z_1, \ldots, z_d) \, dz_1 \cdots dz_d \\ &\quad = \frac{M}{K} \int f(z_1, \ldots, z_d) \, dz_1 \cdots dz_d = \frac{M}{K}. \end{aligned}$$

For large d it becomes increasingly difficult to find g and K such that M/K is large enough for the algorithm to be useful. Hence, while the rejection method is not strictly univariate as the inversion method, it tends to be practically useful only for small d.

The technique is illustrated in Figure 3.3.

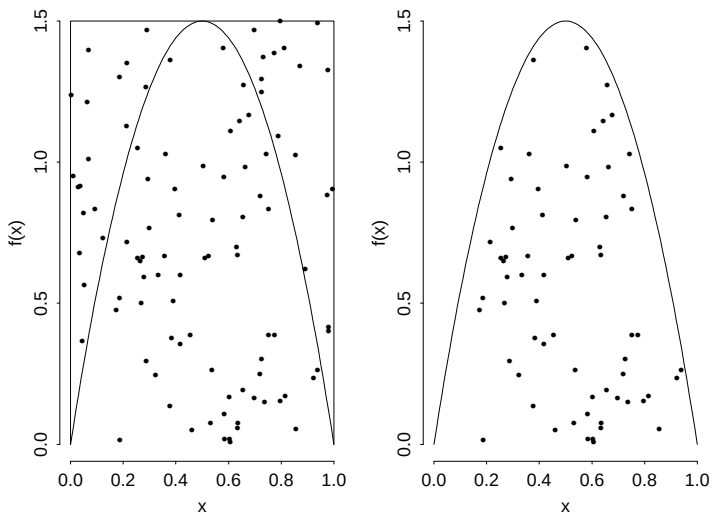

Figure 3.3: Simulation by rejection sampling from an $U[0, 1]$ distribution (here $M = 1$ and $G = 1.5$); the x–coordinates of the points in the right panel constitute a sample with density f

As an example, consider the distribution with density

$$f(x) \propto x^2 e^{-x}; \quad 0 \le x \le 1, \tag{3.7}$$

a truncated gamma distribution. Then, since $f(x) \le e^{-x}$ everywhere, we can set $g(x) = \exp(-x)$ and so simulate from an exponential distribution, rejecting according to the above algorithm. Figure 3.4 shows both $f(x)$ and $g(x)$. Clearly in this case the envelope is very poor so the routine is highly inefficient (though statistically correct). Applying this to generate a sample of 100 data by

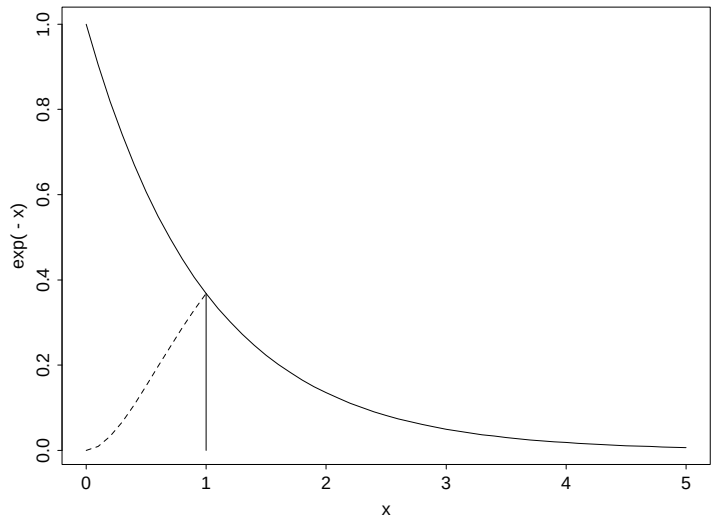

Figure 3.4: Scaled density and envelope

```
 [x,m]=rejsim(100);
 hist(x);
```

using the following code

```
function [x,m]=rejsim(n)
m=0;
for i=1:n
    acc=0;
    while(~acc)
        m=m+1;
        z1=-log(rand);
        z2=rand*exp(-z1);
        if (z2<z1^2*exp(-z1)*(z1<1))
            acc=1;
            x(i)=z1;
        end
    end
end
```

gave the histogram in Figure 3.5. The variable `m` contains the number of random variate pairs (Z_1, Z_2) needed to accept 100 variables from the correct distribution, in our simulation `m=618` suggesting that the algorithm is rather poor. What values of M and G did we use in this example?

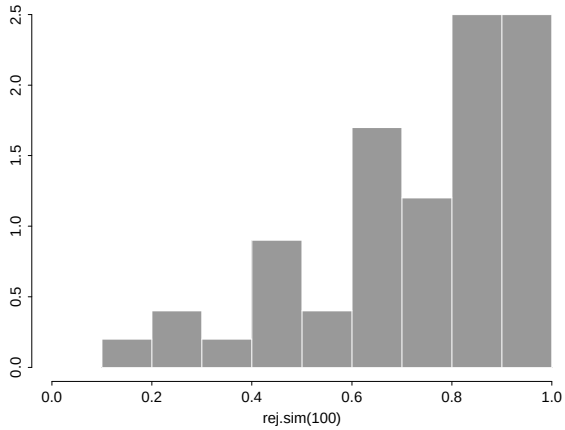

Figure 3.5: Histogram of simulated data

3.4 Conditional methods

The inversion method is strictly univariate, since the inverse F^{-1} is not well-defined for functions $F : \mathbf{R}^d \mapsto [0, 1]$ when $d > 1$. The rejection method is not limited to $d = 1$, but for large d it becomes increasingly difficult to find a bounding function $Kg(x)$ that preserves a reasonably high acceptance rate. A general technique to simulate from a multivariate distribution, using steps of univariate draws, is suggested by a factorization argument. Any d-variate density function f can be factorised recursively as

$$f(x_1, \ldots, x_d) = f_1(x_1) f_2(x_2|x_1) f_3(x_3|x_2, x_1) \cdots f_d(x_d|x_{d-1}, \ldots, x_1). \quad (3.8)$$

Given the above factorisation, a draw from f can now be produced recursively by

Algorithm 3.3.

1. Draw x_1 from the distribution with density $f_1(\cdot)$.
2. Draw x_2 from the distribution with density $f_2(\cdot|x_1)$.
3. Draw x_3 from the distribution with density $f_3(\cdot|x_1, x_2)$.

   $\vdots$

d. Draw x_d from the distribution with density $f_d(\cdot|x_1, x_2, \ldots, x_{d-1})$.

$(x_1, \ldots, x_d)$, is now a draw from $f(x_1, \ldots, x_d)$ in (3.8).

In the above algorithm, each step could be performed with an univariate method. The problem is that, commonly, the factorisation in (3.8) is not explicitly available. For example, deriving f_1 involves the integration

$$f_1(x_1) = \int f(x_1, \ldots, x_d)\, dx_2 \cdots dx_d,$$

which we might not be able to perform analytically.

Example 3.3 (Simulating a Markov Chain)**.** Recall that a Markov Chain is a stochastic process $(X_0, X_1, \ldots, X_n)$ such that, conditionally on $X_{i-1} = x_{i-1}$, X_i is independent of the past $(X_0, \ldots, X_{i-2})$. Assuming x_0 is fixed, the factorisation (3.8) simplifies to

$$f(x_1, \ldots, x_n) = f_1(x_1|x_0) f(x_2|x_1) \cdots f(x_n|x_{n-1}),$$

for a common transition density $f(x_i|x_{i-1})$ of $X_i|X_{i-1} = x_{i-1}$. Thus, to simulate a chain starting from x_0, we proceed recursively as follows

1. Draw x_1 from the distribution with density $f(\cdot|x_0)$.
2. Draw x_2 from the distribution with density $f(\cdot|x_1)$.

$\vdots$

n. Draw x_n from the distribution with density $f(\cdot|x_{n-1})$.

Example 3.4 (Bivariate Normals)**.** Another application of the factorisation argument is useful when generating draws from the bivariate Normal distribution. If

$$(X_1, X_2) \sim N_2\left((\mu_1, \mu_2), \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}\right),$$

we obviously have that $X_1 \sim N(\mu_1, \sigma_1^2)$ and it is a straightforward exercise to show that $X_2|X_1 = x_1 \sim N(\mu_2 + \rho\sigma_2(x_1 - \mu_1)/\sigma_1, \sigma_2^2(1 - \rho^2))$.

Chapter 4

Monte-Carlo integration

Many quantities of interest to statisticians can be formulated as integrals,

$$\tau = E(\phi(X)) = \int \phi(x) f(x)\, dx, \tag{4.1}$$

where $X \in \mathbf{R}^d$, $\phi : \mathbf{R}^d \mapsto \mathbf{R}$ and f is the probability density of X. Note that probabilities correspond to ϕ being an indicator function, i.e.

$$P(X \in A) = \int \mathbf{1}\{x \in A\} f(x)\, dx,$$

where

$$\mathbf{1}\{x \in A\} = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{else} \end{cases} \tag{4.2}$$

When dimension n is large and/or ϕf complicated, the integration in (4.1) can often not be performed analytically. Monte-Carlo integration is a numerical method for integration based on the *Law of Large Numbers* (LLN). The algorithm goes as follows:

Algorithm 4.1 (Basic Monte-Carlo Integration)**.**

1. Draw N values $x_1, \ldots, x_N$ independently from f.
2. Approximate $\tau = E(\phi(X))$ by

$$t_N = t(x_1, \ldots, x_N) = \frac{1}{N} \sum_{i=1}^{N} \phi(x_i).$$

As an example of this, suppose we wish to calculate $P(X < 1, Y < 1)$ where (X, Y) are bivariate Standard Normal with correlation 0.5. This can be written as

$$\int \mathbf{1}\{x < 1, y < 1\} f(x, y)\, dx\, dy \tag{4.3}$$

where f is the bivariate normal density. Thus, provided we can simulate from the bivariate normal, we can estimate this probability as

$$n^{-1}\sum_{i=1}^{n}\mathbf{1}\{x_i<1,y_i<1\} \tag{4.4}$$

which is simply the proportion of simulated points falling in the set defined by $\{(x,y);x<1,y<1\}$. Here we use the approach from Example 3.4 for simulating bivariate Normals. Matlab code to achieve this is

```
function [x,y]=bvnsim(n,m,s,r);
x=randn(1,n)*s(1)+m(1);
y=randn(1,n)*s(2)*sqrt(1-r^2)+m(2)+(r*s(2))/s(1)*(x-m(1));
```

To obtain an estimate of the required probability on the basis of, say, 1000 simulations, we simply need

```
[x,y]=bvnsim(1000,[0,0],[1,1],.5);
mean((x<1)&(y<1))
```

I got the estimate 0.763 doing this. A scatterplot of the simulated values is given in Figure 4.1.

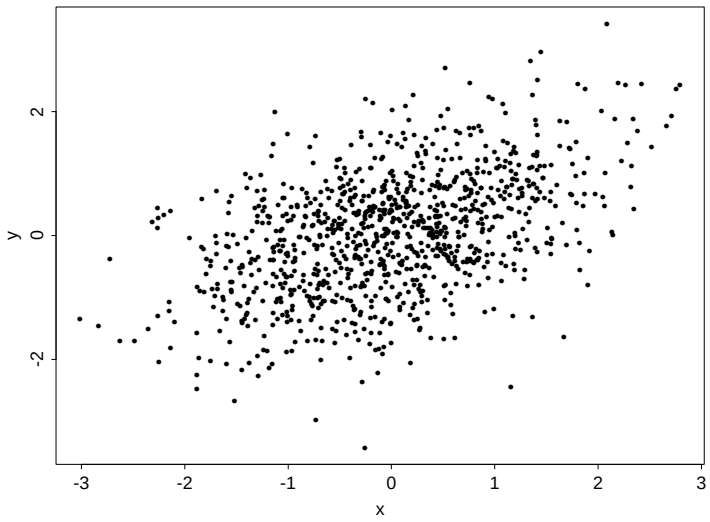

Figure 4.1: Simulated bivariate normals

Example 4.1. For a non-statistical example, say we want to estimate the integral

$$\begin{aligned}\tau &= \int_0^{2\pi} x\sin[1/\cos(\log(x+1))]^2\,dx \\ &= \int (2\pi x\sin[1/\cos(\log(x+1))]^2)(\mathbf{1}\{0\le x\le 2\pi\}/(2\pi))\,dx,\end{aligned}$$

where, of course, the second term of the integrand is the $U[0, 2\pi]$ density function. The integrand is plotted in Figure 4.2, and looks to be a challenge for many numerical methods.

Monte-Carlo integration in Matlab proceeds as follows:

```
x=rand(1,10000)*2*pi;
tn=mean(2*pi*x.*sin(1./cos(log(x+1))).^2)

tn =

    8.6775
```

Maple, using `evalf` on the integral, gave 8.776170832. A larger run of the Monte-Carlo algorithm shows that this might be an overestimate and that the true value is close to 8.756.

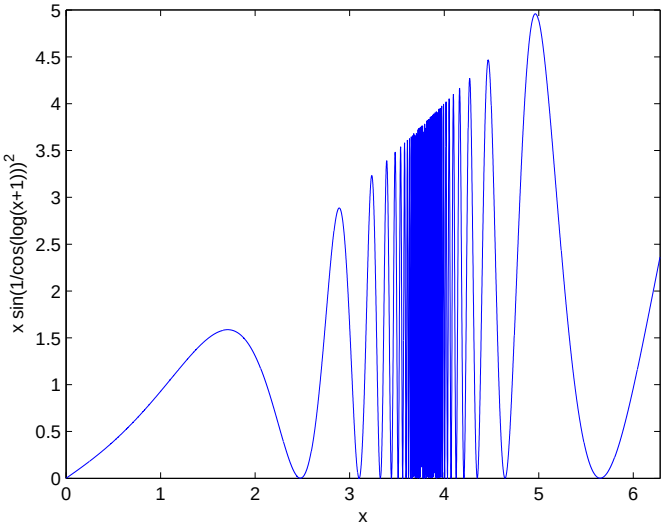

Figure 4.2: An attempt at plotting $x \sin(1/\cos(\log(x+1)))^2$.

We suggested the motivation comes from the LLN. There are many versions of this celebrated theorem, we will provide a simple mean-square version. First note that if $X_1, \ldots, X_n$ is a sequence of random variables and $T_n = t(X_1, \ldots, X_n)$ for a function t, we say that T_n *converges in the mean square sense* to a fixed value τ if

$$E(T_n - \tau)^2 \to 0 \text{ as } n \to \infty.$$

Theorem 4.1 (A Law of Large Numbers)**.** *Assume $Z_1, \ldots, Z_n$ is a sequence of independent random variables with common means $E(Z_i) = \tau$ and variances $Var(Z_i) = \sigma^2$. If $T_n = n^{-1} \sum_{i=1}^{n} Z_i$, we have*

$$E(T_n - \tau)^2 = \frac{\sigma^2}{n} \to 0 \text{ as } n \to \infty. \tag{4.5}$$

Proof. Simple and straightforward; exercise. □

The above theorem tells us that with $Z_i = \phi(X_i)$ where X_i are independent with density f, the arithmetic mean of $Z_1, \ldots, Z_n$ converges in mean square error to $\tau = E(g(X))$. Moreover, it gives the precise rate of the error: $(E(T_n - \tau)^2)^{1/2} = O(n^{-1/2})$ and this rate is *independent of dimension d.* This is in contrast to deterministic methods for numerical integration, like the trapezoidal rule and Simpson's rule, that have errors of $O(n^{-2/d})$ and $O(n^{-4/d})$ respectively. Monte-Carlo integration is to be preferred in high dimensions (greater than 4 and 8 respectively). Another advantage is that we can reuse the drawn values $x_1, \ldots, x_N$ to estimate other expectations with respect to f without much extra effort.

More precise information on the Monte-Carlo error $(T_n - \tau)$ is given by celebrated result no. 2: the *Central Limit Theorem* (CLT).

Theorem 4.2 (Central Limit Theorem)**.** *Assume $Z_1, \ldots, Z_n$ is a sequence of i.i.d. random variables with common means $E(Z_i) = \tau$ and variances $Var(Z_i) = \sigma^2$. If $T_n = n^{-1} \sum_{i=1}^n Z_i$, we have*

$$P\Big(\frac{\sqrt{n}(T_n - \tau)}{\sigma} \le x\Big) \to \Phi(x) \text{ as } n \to \infty, \tag{4.6}$$

where Φ is the distribution function of the $N(0,1)$ distribution.

Proof. Almost as simple, but somewhat less straightforward than LLN. Look it up in a book. □

Slightly less formally, the CLT tells us that the difference $T_n - \tau$ has, at least for large n, approximately an $N(0, \sigma^2/n)$ distribution. With this information we can approximate probabilities like $P(|T_n - \tau| > \epsilon)$, and perhaps more importantly find ϵ such that $P(|T_n - \tau| > \epsilon) = 1 - \alpha$ for some specified confidence level α. To cut this discussion short, the random interval

$$[T_n - 1.96\hat{\sigma}/\sqrt{n}, T_n + 1.96\hat{\sigma}/\sqrt{n}] \tag{4.7}$$

will cover the true value τ with approximately 95% probability. Here $\hat{\sigma}$ is your favourite estimate of standard deviation, e.g. based on

$$\hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^n (z_i - \bar{z})^2, \tag{4.8}$$

and 1.96 is roughly $\Phi^{-1}(0.95)$, the standard Normal 95% quantile.

A similar result to the central limit theorem also holds for the median and general sample quantiles:

Theorem 4.3. *Assume $Z_1, \ldots, Z_n$ is a sequence of i.i.d. random variables with distribution function $F(z - \tau)$ such that $F(0) = \alpha$ and that at zero F has density $f(0) > 0$. Then*

$$P(\sqrt{C_\alpha n}(Z_{(\lceil n\alpha \rceil)} - \tau) \le x) \to \Phi(x) \text{ as } n \to \infty, \tag{4.9}$$

where $C_\alpha = \alpha(1-\alpha)f^2(0)$ and Φ is the distribution function of the $N(0,1)$ distribution.

Bias and the Delta method

It is not always we can find a function t_n such that $E(T_n) = \tau$. For example we might be interested in $\tau = h(E(X))$ for some specified smooth function h. If $\bar{X}$ again is the arithmetic mean, then a natural choice is $T_n = h(\bar{X})$. However, unless h is linear, $E(T_n)$ is not guaranteed to equal τ. This calls for a definition: the *bias* of t (when viewed as an estimator of τ), $T_n = t(X_1, \ldots, X_n)$ is

$$\text{Bias}(t) = E(T_n) - \tau. \tag{4.10}$$

The concept of bias allows us to more fully appreciate the concept of mean square error, since

$$E(T_n - \tau)^2 = \text{Var}(T_n) + \text{Bias}^2(t), \tag{4.11}$$

(show this as an exercise). The mean square error equals variance plus squared bias. In the abovementioned example, a Taylor expansion gives an impression of the size of the bias. Roughly we have with $\mu = E(X)$

$$\begin{aligned} E(T_n - \tau) &= E[h(\bar{X}) - h(\mu)] \\ &\approx E(\bar{X} - \mu)h'(\mu) + \frac{E(\bar{X} - \mu)^2}{2}h''(\mu) \\ &= \frac{\text{Var}(X)}{2n}h''(\mu). \end{aligned} \tag{4.12}$$

And it is reassuring that (4.12) suggests a small bias when sample size n is large. Moreover, since variance of T_n generally is of order $O(n^{-1})$ it will dominate the $O(n^{-2})$ squared bias in (4.11) suggesting that bias is a small problem here (though it can be a serious problem if the above Taylor expansions are not valid).

We now turn to the variance of T_n. First note that while $\text{Var}(\bar{X})$ is easily estimated by e.g. (4.8), estimating $\text{Var}(h(\bar{X}))$ is not so straightforward. An useful result along this line is the *Delta Method*

Theorem 4.4 (The Delta method)**.** *Let r_n be an increasing sequence and S_n a sequence of random variables. If there is μ such that h is differentiable at μ and*

$$P(r_n(S_n - \mu) \leq x) \to F(x), \text{ as } n \to \infty$$

for a distribution function F, then

$$P(r_n(h(S_n) - h(\mu)) \leq x) \to F(x/|h'(\mu)|).$$

Proof. Similar to the Taylor expansion argument in (4.12). □

This theorem suggests that if $S_n = \bar{X}$ has variance σ^2/r_n, then the variance of $T_n = h(S_n)$ will be approximately $\sigma^2 h'(\mu)^2/r_n$ for large n. Moreover, if S_n is asymptotically Normal, so is T_n.

4.1 Importance sampling

Importance sampling is a technique that might substantially decrease the variance of the Monte-Carlo error. It can also be used as a tool for estimating $E(\phi(X))$ in cases where X can not be sampled easily.

We want to calculate

$$\tau = \int \phi(x) f(x) dx \tag{4.13}$$

which can be re–written

$$\tau = \int \psi(x) g(x) dx \tag{4.14}$$

where $\psi(x) = \phi(x) f(x)/g(x)$. Hence, if we obtain a sample $x_1, x_2, \ldots, x_n$ from the distribution of g, then we can estimate the integral by the unbiased estimator

$$t_n = n^{-1} \sum_{i=1}^{n} \psi(x_i), \tag{4.15}$$

for which the variance is

$$\mathrm{Var}(T_n) = n^{-1} \int \{\psi(x) - \tau\}^2 g(x) dx. \tag{4.16}$$

This variance can be very low, much lower than the variance of an estimate based on draws from f, provided g can be chosen so as to make ψ nearly constant. Essentially what is happening is that the simulations are being concentrated in the areas where there is greatest variation in the integrand, so that the informativeness of each simulated value is greatest. Another important advantage of importance sampling comes in problems where drawing from f is difficult. Here draws from f can be replaced by draws from an almost arbitrary density g (though it is essential that $\phi f/g$ remain bounded).

This example illustrates the idea. Suppose we want to estimate the probability $P(X > 2)$, where X follows a Cauchy distribution with density function

$$f(x) = \frac{1}{\pi(1 + x^2)} \tag{4.17}$$

so we require the integral

$$\int \mathbf{1}\{x > 2\} f(x) dx. \tag{4.18}$$

We could simulate from the Cauchy directly and apply basic Monte-Carlo integration, but the variance of this estimator is substantial. As with the bivariate Normal example, the estimator is the empirical proportion of exceedances; exceedances are rare, so the variance is large compared to its mean. Put differently, we are spending most of our simulation budget on

estimating the integral of $\mathbf{1}\{x > 2\}f(x)$ over an area (i.e. around the origin) where we know it equals zero.

Alternatively, we observe that for large x, $f(x)$ is similar in behaviour to the density $g(x) = 2/x^2$ on $x > 2$. By inversion, we can simulate from g by letting $x_i = 2/u_i$ where $u_i \sim U[0, 1]$. Thus, our estimator becomes:

$$t_n = n^{-1} \sum_{i=1}^{n} \frac{x_i^2}{2\pi(1 + x_i^2)}, \tag{4.19}$$

where $x_i = 2/u_i$. Implementing this with the Matlab function

```
function [tn,cum]=impsamp(n);
x=2./rand(1,n);
psi=x.^2./(2*pi*(1+x.^2));
tn=mean(psi);
cum=cumsum(psi)./(1:n);
```

and processing

```
[tn,cum]=impsamp(1000);
plot(cum)
```

gave the estimate $t_n = .1478$. The exact value is $.5 - \pi^{-1} \tan 2 = .1476$. In Figure 4.3 the convergence of this sample mean to the true value is demonstrated as a function of n by plotting the additional output vector `cum`.

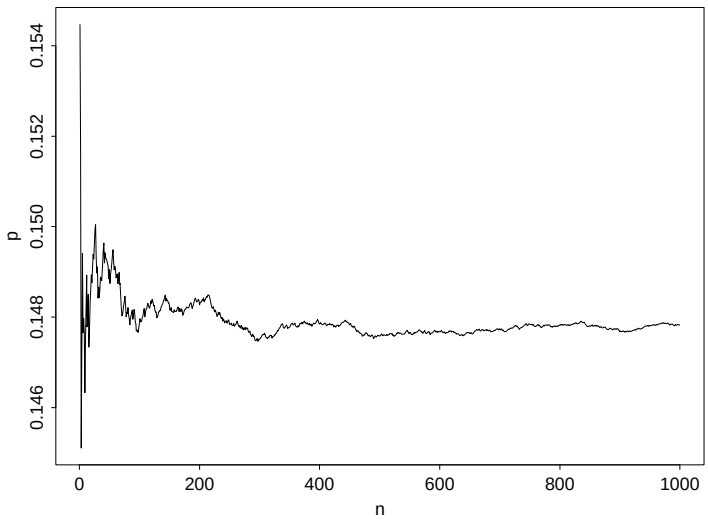

Figure 4.3: Convergence of importance sampled mean

For comparison, in Figure 4.4, we show how this compares with a sequence of estimators based on the sample mean when simulating directly

from a Cauchy distribution. Clearly, the reduction in variability is substantial (the importance sampled estimator looks like a straight line).

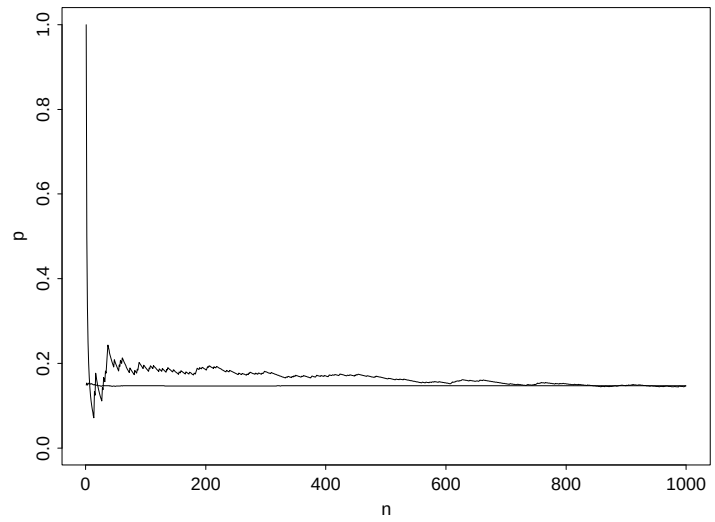

Figure 4.4: Comparison of importance sampled mean with standard estimator

4.1.1 Unknown constant of proportionality

To be able to use the above importance sampling techniques, we need to know $f(x)$ explicitly. Just knowing Mf for an unknown constant of proportionality M is not sufficient. However, importance sampling can also be used to approximate M. Note that,

$$M = \int Mf(x)\,dx = \int \frac{Mf(x)}{g_2(x)} g_2(x)\,dx, \tag{4.20}$$

for a density g_2. Thus, based on a sample $x_1, x_2, \ldots, x_N$ from g_2, we can approximate M by

$$t_N = \frac{1}{N}\sum_{i=1}^{N} \frac{Mf(x_i)}{g_2(x_i)}. \tag{4.21}$$

It should be noted that this approximation puts some restrictions on the choice of g_2. To have a finite variance, we need (with $X' \sim g_2$)

$$E\left(\frac{(Mf(X'))^2}{g_2(X')^2}\right) = \int \frac{(Mf(x))^2}{g_2(x)}\,dx,$$

to be finite, i.e. f^2/g_2 is integrable. Hence, a natural requirement is that f/g_2 is bounded. This can now be used to approximate $\tau = E(\phi(X))$ using sequences $x_1, x_2, \ldots, x_N$ from g_2 and $y_1, y_2, \ldots, y_N$ from g through

$$\frac{t'_N}{t_N} = \left(\sum_{i=1}^{N} \frac{\phi(y_i) M f(y_i)}{g(y_i)} \right) \Big/ \left(\sum_{i=1}^{N} \frac{M f(x_i)}{g_2(x_i)} \right), \tag{4.22}$$

where the numerator approximates $M\tau$ and denominator M. Of course we could use $g = g_2$ and $x_i = y_i$ in (4.22), but this is not usually the most efficient choice.

Chapter 5

Markov Chain Monte-Carlo

Today, the most-used method for simulating from complicated and/or high-dimensional distributions is *Markov Chain Monte Carlo* (MCMC). The basic idea of MCMC is to construct a Markov Chain that has f as stationary distribution, where f is the distribution we want to simulate from. In this chapter we introduce the algorithms, more applications will be given later.

5.1 Markov chains with continuous state-space

The theory for Markov chains that take values in a continuous state-space is complex. Much more so than the theory for finite/countable state-spaces, that you might have already been exposed to. We will here not pretend to give a full account of the underlying theory, but be happy with formulating a number of sufficient conditions under which the methods work. Links to more advanced material will be provided on the course web-page.

In this chapter, we will consider Markov-Chains $X_1, X_2, \ldots, X_N$ defined through a starting value x_0 and a transition density $\tilde{q}$. The transition density $x_i \mapsto \tilde{q}(x_{i-1}, x_i) = g(x_i|x_{i-1})$ is the density of $X_i|X_{i-1} = x_{i-1}$, and this determines the evolution of the Markov-chain across the state-space. We would ideally like to draw the starting value x_0 and choose $\tilde{q}$ in such a way that the following realisations $x_1, x_2, \ldots, x_N$ are independent draws from f, but this is a too ambitious task in general. In contrast to the previous chapter, here our draws will neither be independent nor *exactly* distributed according to f. What we will require is that f is the *asymptotic distribution* of the chain, i.e. if X_i takes values in $\mathcal{X}$,

$$P(X_n \in A) \to \int_A f(x)\, dx, \tag{5.1}$$

as $n \to \infty$, for all subsets A of $\mathcal{X}$ and independently of the starting value $x_0 \in \mathcal{X}$.

A condition on the transition density that ensures the Markov chain has a stationary distribution f, is that it satisfies the *global balance condition*

$$f(y)\tilde{q}(y, x) = f(x)\tilde{q}(x, y). \tag{5.2}$$

This says roughly that the flow of probability mass is equal in both directions (i.e. from x to y and vice versa). Global balance is not sufficient for the stationary distribution to be unique (hence the Markov chain might converge to a different stationary distribution). Sufficient conditions for uniqueness are irreducibility and aperiodicity of the chain, a simple sufficient condition for this is that the support of $f(y)$ is contained in the support of $y \mapsto \tilde{q}(x, y)$ for all x (minimal *necessary* conditions for $\tilde{q}$ to satisfy (5.1) are not known in general, however our *sufficient* conditions are far from the weakest known in literature).

5.2 Markov chain Monte-Carlo integration

Before going into the details of how to construct a Markov chain with specified asymptotic distribution, we will look at Monte-Carlo integration under the new assumption that draws are neither independent not exactly from the target distribution. Along this line, we need a Central Limit Theorem for Markov chains. First we observe that if $X_1, X_2, \ldots, X_n$ is a Markov chain on $\mathbf{R}^d$ and $\phi : \mathbf{R}^d \mapsto \mathbf{R}$, then with $Z_i = \phi(X_i)$, the sequence $Z_1, Z_2, \ldots, Z_n$ forms a Markov chain on $\mathbf{R}$. As before, we want to approximate $\tau = E(Z) = E(\phi(X))$ by $t_N = N^{-1} \sum_{i=1}^{N} z^{(i)}$, for a sequence $z^{(i)} = \phi(x^{(i)})$ of draws from the chain.

5.2.1 Burn-in

An immediate concern is that, unless we can produce a single starting value x^0 with the correct distribution, our sampled random variables will only *asymptotically* have the correct distribution. This will induce a bias in our estimate of τ.

In general, given some starting value $x^{(0)}$, there will be iterations $x^{(i)}$, $i = 1, \ldots, k$, before the distribution of $x^{(i)}$ can be regarded as "sufficiently close" to the stationary distribution $f(x)$ in order to be useful for further analysis (i.e. the value of $x^{(i)}$ is still strongly influenced by the choice of $x^{(0)}$ when $i \leq k$). The values $x^{(0)}, \ldots, x^{(k)}$ are then referred to as the *burn-in* of the chain, and they are usually discarded in the subsequent output analysis.

For example when estimating $\int \phi(x) f(x)\, dx$, it is common to use

$$\frac{1}{N-k} \sum_{i=k+1}^{N} \phi(x^{(i)}).$$

An illustration is given in Figure 5.1. We now provide the CLT for Markov Chains.

Theorem 5.1. *Suppose a geometrically ergodic Markov chain X_i, $i = 1, \ldots, n$, on $\mathbf{R}^d$ with stationary distribution f and a real-valued function $\phi : \mathbf{R}^d \mapsto \mathbf{R}$ satisfies $E(\phi^{2+\epsilon}(X)) \leq \infty$ for some $\epsilon > 0$, then with $\tau = E(\phi(X))$ and $T_n = n^{-1} \sum_{i=1}^{n} \phi(X_i)$*

$$P\big(\frac{\sqrt{n}(T_n - \tau)}{\sigma} \leq x\big) \to \Phi(x) \tag{5.3}$$

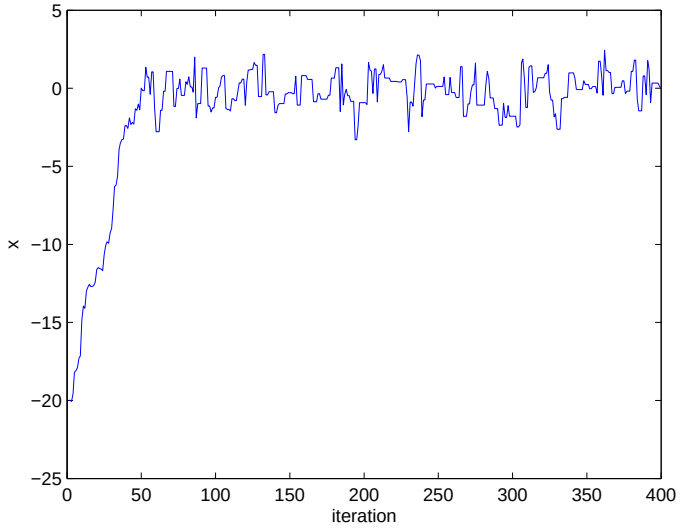

Figure 5.1: A Markov chain starting from $x^{(0)} = -20$ that seems to have reached its stationary distribution after roughly 70 iterations

where Φ *is the distribution function of the* $N(0,1)$ *distribution and*

$$\sigma^2 = r(0) + 2\sum_{i=1}^{\infty} r(i) \tag{5.4}$$

where $r(i) = \lim_{k\to\infty} Cov\{\phi(X_k), \phi(X_{k+i})\}$*, the covariance function of a stationary version of the chain.*

Note that the above holds for an arbitrary starting value x_0. The error due to "wrong" starting value, i.e. the bias $E(T_n) - \tau$, is of $O(n^{-1})$. Hence squared bias is dominated by variance, and asymptotically negligible. This does not suggest that it is unnecessary to discard burn-in, but rather that it is unnecessary to put too much effort into deciding on how long it should be. A visual inspection usually suffices.

5.2.2 After burn-in

If we manage to simulate our starting value $x^{(0)}$ from the target distribution f, all subsequent values will also have the correct distribution. Would our problems then be solved if we were given that single magic starting value with the correct distribution? As the above discussion suggests, the answer is no. The "big problem" of MCMC is that (5.4) can be very large, especially in problems where the dimension d is large. It is important to understand that converging to the stationary distribution (or getting close enough) is just the very beginning of the analysis. After that we need to do sufficiently many iterations in order for the variance σ^2/n of T_n to be small.

It is actually a good idea to choose starting values $x^{(0)}$ that are *far away* from the main support of the distribution. If the chain then takes a long time to stabilize you can expect that even after reaching stationarity, the autocorrelation of the chain will be high and σ^2 in (5.4) large. Here we give a rough guide for estimating σ^2, which is needed want we to produce a confidence interval:

1. We assume your draws $x^{(i)}$, $i = 1, \ldots, N$ are d-variate vectors. First make a visual inspection of each of the d trajectories, and decide a burn-in $1, \ldots, k$ where *all* of them seems to have reached stationarity. Throw away the first k sample vectors.

2. Now you want to estimate $E(\phi(X))$. Compute $z_i = \phi(x^{(i+k)})$, $i = 1, \ldots, N-k$ and estimate the autocorrelation function of the sequence z_i, i.e. the function $\rho(t) = \mathrm{Corr}(Z_1, Z_t)$ over a range of values $t = 1, \ldots, L$, this can be done with Matlabs `xcorr` function (though it uses the range $-L, \ldots, L$). A reasonable choice of L is around $(N-k)/50$ (estimates of $\rho(L)$ tends to be too unreliable for larger values). If the estimated autocorrelation function does not reach zero in the range $1, \ldots, L$, go back to the first step and choose a larger value for N.

3. Divide your sample into m batches of size l, $ml = N-k$. Here l should be much larger than the time it took for the autocorrelation to reach zero in the previous step. The batches are $(z_1, \ldots, z_l)$,$(z_{l+1}, \ldots, z_{2l})$, ..., $(z_{m(l-l)+1}, \ldots, z_{ml})$. Now compute the arithmetic mean $\bar{z}_j$ of each batch, $j = 1, \ldots, m$. Estimate τ by the mean of the batch means and σ^2 by s^2/m, where s^2 is the empirical variance of the batch means.

This procedure is based on the fact that if batches are sufficiently large, their arithmetic means should be approximately uncorrelated. Hence, since our estimate is the mean of m approximately independent batch means it should have variance σ_b^2/m, where σ_b^2 is the variance of a batch mean.

We now turn to the actual construction of the Markov chains.

5.3 The Metropolis-Hastings algorithm

How do you choose transition density $\tilde{q}$ in order to satisfy (5.1)? The idea behind the Metropolis-Hastings algorithm is to start with an (almost) arbitrary transition density q. This density will not give the correct asymptotic distribution f, but we could try to repair this by rejecting some of the moves it proposes. Thus, we construct a new transition density $\tilde{q}$ defined by

$$\tilde{q}(x, y) = \alpha(x, y)q(x, y) + (1 - \alpha(x, y))\delta_x(y), \tag{5.5}$$

where $\delta_x(y)$ is a point-mass at x. This implies that we stay at the level x if the proposed value is rejected and we reject a proposal y^* with probability $1 - \alpha(x, y^*)$. Simulating from the density $y \mapsto \tilde{q}(x, y)$ works as follows

1. Draw y^* from $q(x, \cdot)$.

2. Draw u from $U(0,1)$.

3. If $u < \alpha(x, y^*)$ set $y = y^*$,
   else set $y = x$.

We now have to match our proposal density q with a suitable acceptance probability α. The choice of the *Metropolis-Hastings algorithm*, based on satisfying the global balance equation (5.2), is

$$\alpha(x, y) = \min(1, \frac{f(y)q(y, x)}{f(x)q(x, y)}), \tag{5.6}$$

you might want to check that this actually satisfies (5.2).

Algorithm 5.1 (The Metropolis-Hastings algorithm)**.**

1. Choose a starting value $x^{(0)}$.
2. Repeat for $i = 1, \ldots, N$:
   - i.1 Draw y^* from $q(x^{(i-1)}, \cdot)$.
   - i.2 Draw u from $U(0,1)$.
   - i.3 If $u < \alpha(x^{(i-1)}, y^*)$ set $x^{(i)} = y^*$, else set $x^{(i)} = x^{(i-1)}$.
3. $x^{(1)}, x^{(2)}, \ldots, x^{(N)}$ is now a sequence of dependent draws, approximately from f.

There are three general types of Metropolis-Hastings candidate generating densities q used in practise; the *Gibbs sampler*, *independence sampler* and the *random walk sampler*. Below we will discuss their relative merits and problems.

5.4 The Gibbs-sampler

The Gibbs-sampler is often viewed as a separate algorithm rather than a special case of the Metropolis-Hastings algorithm. It is based on partitioning the vector state-space $\mathbf{R}^d = \mathbf{R}^{d_1} \times \cdots \mathbf{R}^{d_m}$ into blocks of sizes d_i, $i = 1, \ldots, m$ such that $d_1 + \cdots + d_m = d$. We will write $z = (z_1, \ldots, z_m) = (x_1, \ldots, x_d)$, where $z_i \in \mathbf{R}^{d_i}$ (e.g. $z_1 = (x_1, \ldots, x_{d_1})$. With this partition, the Gibbs-sampler with target f is now:

Algorithm 5.2 (The Gibbs-sampler)**.**

1. Choose a starting value $z^{(0)}$.
2. Repeat for $i = 1, \ldots, N$:
   - i.1 Draw $z_1^{(i)}$ from $f(z_1|z_2^{(i-1)}, \ldots, z_m^{(i-1)})$.
   - i.2 Draw $z_2^{(i)}$ from $f(z_2|z_1^{(i)}, z_3^{(i-1)}, \ldots, z_m^{(i-1)})$.
   - i.3 Draw $z_3^{(i)}$ from $f(z_3|z_1^{(i)}, z_2^{(i)}, z_4^{(i-1)}, \ldots, z_m^{(i-1)})$.
   - $\vdots$
   - i.m Draw $z_m^{(i)}$ from $f(z_m|z_1^{(i)}, z_2^{(i)}, \ldots, z_{m-1}^{(i)})$.
3. $z^{(1)}, z^{(2)}, \ldots z^{(N)}$, is now a sequence of dependent draws approximately from f.

This corresponds to an MH-algorithm with a particular proposal density and $\alpha = 1$. What is the proposal density q?

Note the similarity with Algorithm 3.3. The difference is that here we draw each z_i conditionally on all the others which is easier since these conditional distributions are much easier to derive. For example,

$$z_1 \mapsto f(z_1|z_2, \ldots, z_m) = \frac{f(z_1, \ldots, z_m)}{f(z_2, \ldots, z_m)} \propto f(z_1, \ldots, z_m).$$

Hence, if we know $f(z_1, \ldots, z_m)$, we also know all the conditionals needed (up to a constant of proportionality). The Gibbs-sampler is the most popular MCMC algorithm and given a suitable choice of partition of the state-space it works well for most applications to Bayesian statistics. We will have the opportunity to study it more closely in action in the subsequent part on Bayesian statistics of this course. Poor performance occurs when there is a high dependence between the components Z_i. This is due to the fact that the Gibbs-sampler only moves along the coordinate axes of the vector $(z_1, \ldots, z_m)$, illustrated by Figure 5.2. One remedy to this problem is to merge the dependent components into a single larger component, but this is not always practical.

Example 5.1 (Bivariate Normals)**.** Bivariate Normals can be drawn with the methods of Examples 3.4 or 3.2. Here we will use the Gibbs-sampler instead. We want to draw from

$$(X_1, X_2) \sim N_2\left(0, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right),$$

and choose partition $(X_1, X_2) = (Z_1, Z_2)$. The conditional distributions are given by

$$Z_1|Z_2 = z_2 \sim N(\rho z_2, \sqrt{1-\rho^2}) \text{ and } Z_2|Z_1 = z_1 \sim N(\rho z_1, \sqrt{1-\rho^2}).$$

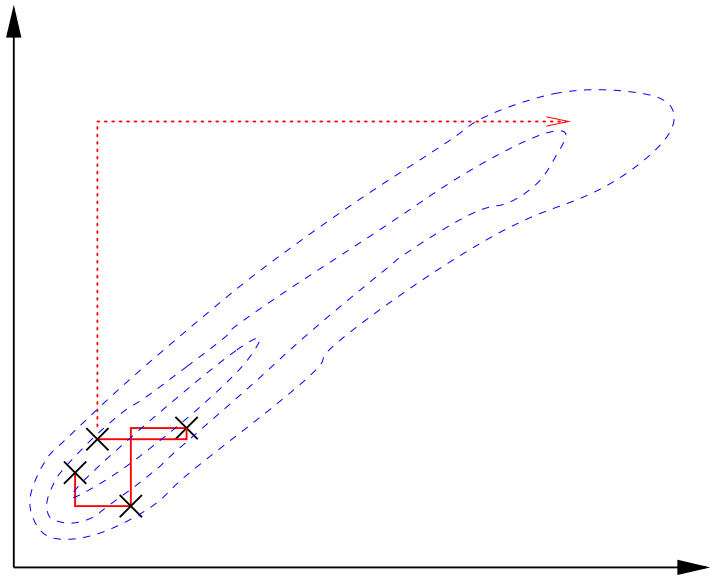

Figure 5.2: For a target (dashed) with strongly dependent components the Gibbs sampler will move slowly across the support since "big jumps", like the dashed move, would involve first simulating a highly unlikely value from $f(z_1|z_2)$.

The following script draws 1000 values starting from $(z_1^{(0)}, z_2^{(0)}) = (0, 0)$.

```
function z=bivngibbs(rho)
z=zeros(1001,2);
for i=2:1000
    z(i,1)=randn*sqrt(1-rho^2)+rho*z(i-1,2);
    z(i,2)=randn*sqrt(1-rho^2)+rho*z(i,1);
end
z=z(2:1001,:);
```

In Figure 5.3 we have plotted the output for $\rho = 0.5$ and $\rho = 0.99$. Note the strong dependence between successive draws when $\rho = 0.99$. Also note that each individual panel constitute approximate draws from the same distribution, i.e. the $N(0, 1)$ distribution.

5.5 Independence proposal

The independence proposal amounts to proposing a candidate value y^* *independently* of the current position of the Markov Chain, i.e. we choose $q(x, y) = q(y)$. A necessary requirement here is that $\text{supp}(f) \subseteq \text{supp}(q)$; if this is not satisfied, some parts of the support of f will never be reached. This candidate is then accepted with probability

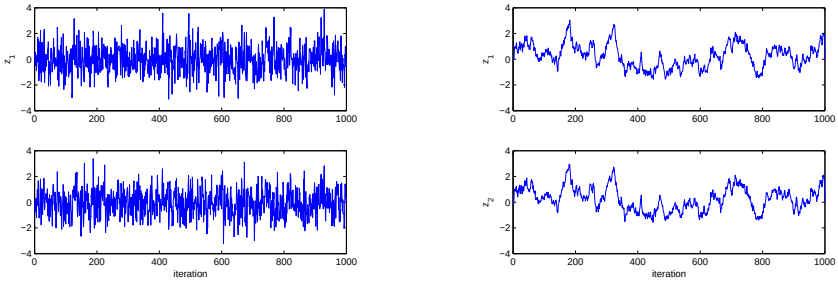

Figure 5.3: Gibbs-draws from Example 5.1, left with $\rho = 0.5$ and right with $\rho = 0.99$.

$$\alpha(x, y^*) = \begin{cases} \min\{\frac{f(y^*)q(x)}{f(x)q(y^*)}, 1\} & \text{if } f(x)q(y^*) > 0 \\ 1 & \text{if } f(x)q(y^*) = 0 \end{cases}$$

And we immediately see that if $q(y) = f(y)$, $\alpha(x, y) \equiv 1$, i.e. all candidates are accepted. Of course, if we really could simulate a candidate directly from $q(y) = f(y)$ we would not have bothered about implementing an MH algorithm in the first place. Still, this fact suggests that we should attempt to find a candidate generating density $q(y)$ that is as good an approximation to $f(y)$ as possible, similarily to the rejection sampling algorithm. The main difference is that we don't need to worry about deriving constants M or K such that $Mf < Kq$ when we do independence sampling. To ensure good performance of the sampler it is advisable to ensure that such constants *exists*, though we do not need to derive it explicitly. If it does not exist, the algorithm will have problems reaching parts of the target support, typically the extreme tail of the target. This is best illustrated with an example; assume we want to simulate from $f(x) \propto 1/(1+x)^3$ using an Exp(1) MH independence proposal. A plot of (unnormalised) densities q and $1/(1+x)^3$ in Figure 5.4 does not indicate any problems — the main support of the two densities seem similar. The algorithm is implemented as follows

```
function [x,acc]=indsamp(m,x0)
x(1:m)=x0;
acc=0;
for i=1:m
    y=gamrnd(1,1);
    a=min(((y+1)^(-3))*gampdf(x(i),1,1)...
                /gampdf(y,1,1)/((x(i)+1)^(-3)),1);
    if (rand<a)
        x(i+1)=y;
        acc=acc+1;
    else
        x(i+1)=x(i);
    end
end
```

```
acc=acc/m;
```

and 1000 simulated values are shown in the left panel of Figure 5.5 with starting value `x0=1`. Looking at the output does not immediately indicate any problems either. However, a second run, now with starting value `x0=15`, is shown in the right panel of the same figure; 715 proposed moves away from the tail are rejected.

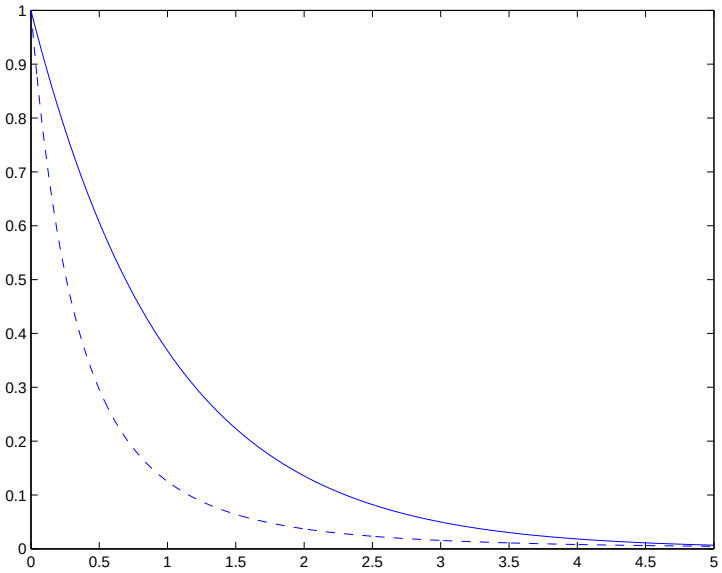

Figure 5.4: Unnormalized target $1/(1+x)^3$ (dashed) and Exp(1) independence proposal (solid).

The problem here is that since the light-tailed proposal density generates large values too seldom, the chain needs to stay in the tail for a very long time once it gets there to preserve stationarity. As a consequence this induces large autocorrelation which reduces the information contained in the output.

The main problem with the independence sampler is that unless q is a good approximations of f (and *especially so in high dimensional problems*), most proposals will be rejected and as a consequence autocorrelations high.

5.6 Random walk proposal

While the independence sampler needs careful consideration when choosing a candidate generating kernel q, random walk kernels are more "black box". As a drawback they will never be quite as efficient as a finely tuned independence proposal.

Random walk proposals are characterised by $q(x,y) = g(|x-y|)$, i.e. they are symmetric centered around the current value, and as a consequence α

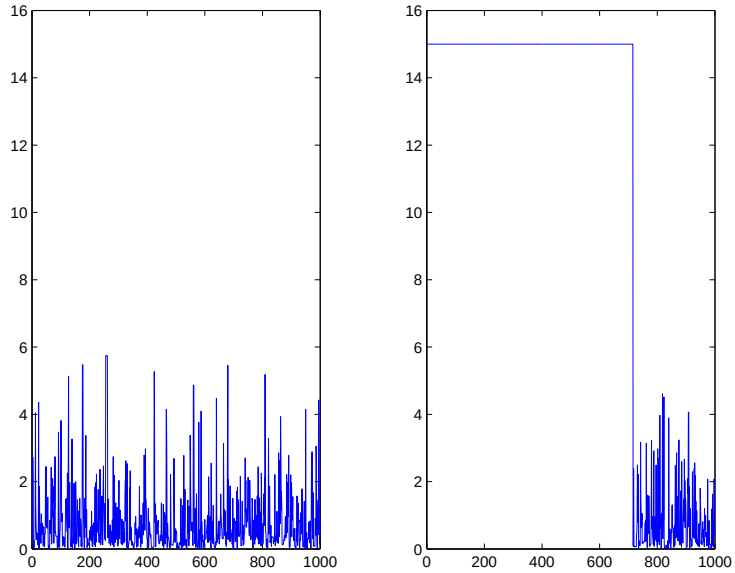

Figure 5.5: Output from independence sampler. Left with small starting value and right starting from the tail.

simplifies to

$$\alpha(x, y) = \min\{f(y)/f(x), 1\}.$$

Note especially that moves to areas with higher density, $f(y^*) > f(x)$, are always accepted.

A common choice is to let g be an $\mathrm{N}(0, s^2\Sigma)$ density with a suitably chosen covariance matrix Σ and a scaling constant s. In this case proposals are generated by adding a zero-mean Gaussian vector to the current state, $y^* = x + s\epsilon$ where $\epsilon \in N(0, \Sigma)$. What remains for the practitioner in this case is the choice of scaling constant s and covariance matrix Σ. The latter choice is difficult and in practise Σ is often chosen to be a diagonal matrix of ones. For s some general rules of thumb can be derived though. Lets first look at a simple example:

The function `rwmh.m` implements a Gaussian random-walk for a standard Gaussian target density given input `N` number of iterations, `x0` starting value and `s` scaling constant:

```
function [x,acc]=rwmh(N,x0,s);
x(1:N)=x0;
acc=0;
for i=1:N-1
    y=x(i)+s*randn;
    a=min(exp(-y^2/2+x(i)^2/2),1);
    if rand<a
        x(i+1)=y;
```

```
        acc=acc+1;
    else
        x(i+1)=x(i);
    end
end
```

In Figure 5.6 we have plotted outputs from 1000 iterations with `x0=0` and scales `s` set to 0.3, 30 and 3. For the smallest scaling constant almost all of the proposed moves are accepted (908 out of 1000) but they are too small and the chain travels slowly across the support of f. For $s = 30$ very few proposals are accepted (48 out of 1000), but they are all very "innovative". Finally the result using $s = 3$ looks most promising. The methods can be compared by estimating the auto-correlation functions of the output as is done in Figure 5.7. Here it is clear that if our goal is to estimate the mean, this will be most efficient for $s = 3$.

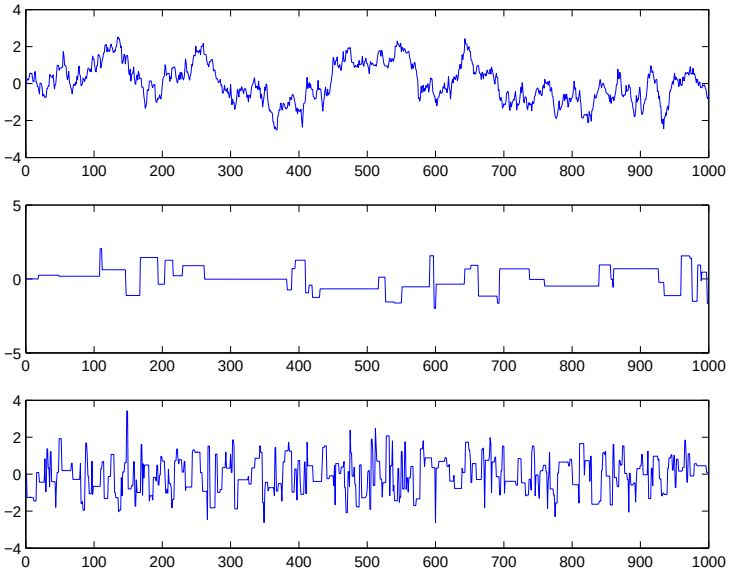

Figure 5.6: Random walks with too small s (top), too large s (middle) and well-tuned s (bottom)

Are there any general rules of thumb on how to choose random-walk scale s? The answer is yes, at least asymptotically. It turns out monitoring the acceptance rate is the key, and that for a large class of densities $f : \mathbf{R}^d \mapsto \mathbf{R}$, asymptotically, as $d \to \infty$, it is optimal to choose s in such a way that 23.4...% of the proposed moves are accepted (when $d = 1$ a slightly higher acceptance rate is often favourable). This is optimal for any fixed Σ.

Returning to the choice of Σ, note in Figure 5.8 that if we set Σ to be a diagonal matrix of ones, the random-walk sampler will have similar problems with dependent components as the Gibbs-sampler.

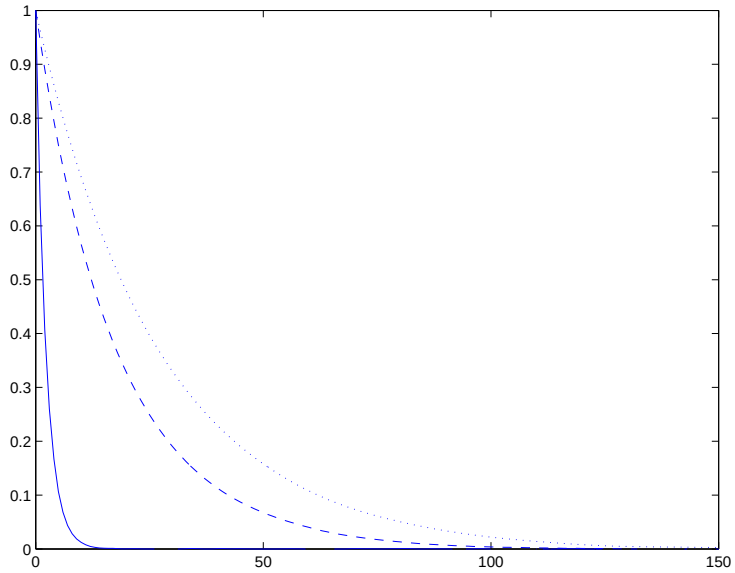

Figure 5.7: Autocorrelation functions for the output with $s = .3$ (dotted) $s = 30$ (dashed) and $s = 3$ (solid)

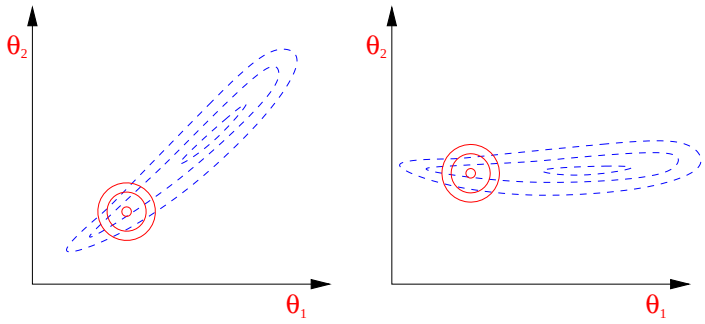

Figure 5.8: Left panel: With a symmetric random-walk proposal (solid contours) tuned to the optimal acceptance rate, movement will be slow if correlation of the posterior (dashed contours) is strong or variances different.

In addition, orthogonalising does not help unless we also standardise variances. A good choice of Σ is one that is similar to the covariance matrix of the target (up to a proportionality constant). An approximation that is often useful is to let Σ be proportional to the Hessian matrix of $\log f$ (i.e. $H(x)$ with entries $H_{i,j} = d(\log f)^2/(dx_i\, dx_j)$) evaluated at e.g. a mode of f if this can be found.

5.6.1 Multiplicative random walk

If the target has a very heavy tail, the random walk proposal will generally perform poorly. In a similar fashion to the independence sampler with a light-tailed proposal, since it takes the chain a long time to travel all the way to the tail, it will stay there for a long time when it reaches it.

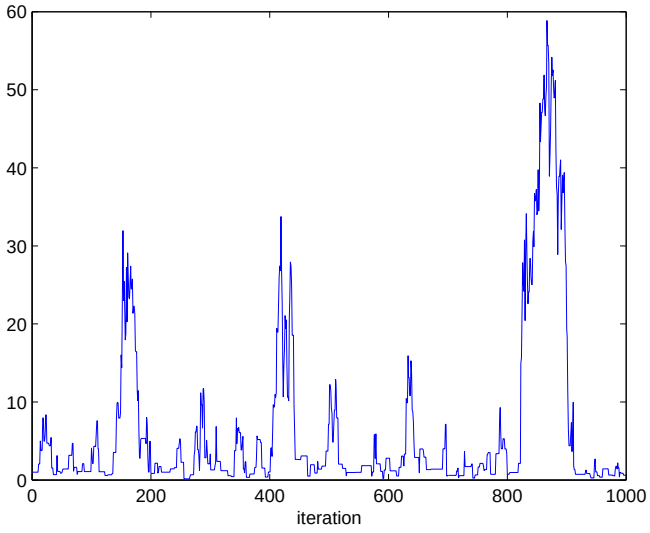

Figure 5.9: Typical behaviour of a random walk Metropolis Hastings on a heavytailed target. Seemingly stable behaviour is exchanged with long excursions in the tails.

In this situation, a *Multiplicative random-walk proposal* is often more efficient. The random-walk proposal is formed by adding an independent component, $y^* = x^{(i-1)} + \epsilon$, the *multiplicative* random-walk instead multiplies with an independent component, $y^* = x^{(i-1)}\epsilon$. If we denote the density of ϵ by g, the proposal-generating density is $q(x, y) = g(y/x)/x$ and we accept/reject with probability

$$\alpha(x, y) = \min(1, \frac{f(y)g(x/y)x}{f(x)g(y/x)y}).$$

5.7 Hybrid strategies

In statistical problems there is often a natural choice of partition for the Gibbs-sampler, however one or more of the conditional distributions in Algorithm 5.2 might be difficult to sample from directly. In this case, an exact draw from the tricky conditionals can be replaced by one iteration of the Metropolis-Hastings algorithm. This strategy is often necessary in complex

problems and is sometimes referred to as the Metropolis–within–Gibbs algorithm.

Part II

Applications to classical inference

Chapter 6

Statistical models

... all models are wrong, but some are useful. (G.E.P. Box)

Suppose we have observed *data* $y \in \mathcal{Y}$. $\mathcal{Y}$ is the *sample space*, typically a finite or countable set, or a subset of $\mathbf{R}^n$. Our data y is assumed to be a *realisation* of a random variable Y with probability distribution P_0. I.e. P_0 assigns probabilities to sets $A \subseteq \mathcal{Y}$, denoted $P_0(Y \in A)$. The statistical problem is that P_0 is unknown to us, and as a statistician you would like to say something about the properties of this distribution. A *statistical model* is a set $\mathcal{P}$ of probability distributions on $\mathcal{Y}$ that contains P_0. The most general model is then $\mathcal{P} = \{P; P \text{ is a probability distribution on } \mathcal{Y}\}$, but this tends to be too generous. The space is simply too large and its elements are difficult to distinguish in a practical meaningful manner. A common way to restrict the size of $\mathcal{P}$ is to assume that P_0 belongs to a *parametric family* of distributions indexed by a finite-dimensional parameter $\theta \in \Theta$, $\mathcal{P} = \{P_\theta; \theta \in \Theta\}$, here Θ is the *parameter space* and we write $P_0 = P_{\theta_0}$. We will insist that there is a one-to-one correspondence between Θ and $\mathcal{P}$, i.e. each $\theta \in \Theta$ corresponds to an *unique* probability distribution $P_\theta \in \mathcal{P}$.

Example 6.1 (Independent Normals)**.** If we assume $y = (y_1, \ldots, y_n) \in \mathbf{R}^n = \mathcal{Y}$ is a vector of independent draws from $N(\mu_0, \sigma_0^2)$, $\theta_0 = (\mu_0, \sigma_0^2) \in \mathbf{R} \times \mathbf{R}^+ = \Theta$ and $\mathcal{P}$ is the set of distributions with corresponding distribution functions

$$F_\theta(y) = \prod_{i=1}^{n} \Phi((y_i - \mu)/\sigma), \quad \theta = (\mu, \sigma) \in \Theta.$$

In the above example, introducing a parametric family restricted the model of all probability distributions on $\mathbf{R}^n$ to a model of dimension 2. Sometimes, you do not want to restrict yourself to a family of distributions indexed by a *finite*-dimensional parameter, but still consider only a small subset of the "most general" model:

Example 6.2 (Independent identically distributed (i.i.d.) random variables)**.** Here we assume $y = (y_1, \ldots, y_n) \in \mathbf{R}^n = \mathcal{Y}$ is a vector of independent draws from a distribution function $F_0 : \mathbf{R} \mapsto [0, 1]$, but we make no assumptions on the shape of F_0. If we denote by $\mathcal{F}$ the set of all univariate

distribution functions, $\Theta = \mathcal{F}$ and the model is $\mathcal{P} = \{P_F; F \in \Theta\}$ where P_F is the probability distribution with corresponding distribution function $F_n(y) = \prod_{i=1}^n F(y_i)$.

Here, by the i.i.d. assumption, the dimension of $\mathcal{P}$ was reduced from that of the set of distribution functions on $\mathbf{R}^n$ to that of the corresponding set on $\mathbf{R}$. While the model in Example 6.2 could be seen as a parametric model with parameter-space $\mathcal{F}$, it is usually referred to as a *non-parametric* model due to the infinite dimension of $\mathcal{F}$. A hybrid between a parametric and a non-parametric model, i.e. a model $\mathcal{P} = \{P_{\theta,F}; (\theta, F) \in \Theta_1 \times \Theta_2\}$, is referred to as a *semi-parametric* model. From a conceptual point of view, the distinction parametric/non-parametric/semi-parametric is not important. We make it here because of methodological differences.

6.1 Functions of data

Definition 6.1 (Statistic)**.** A *statistic* is a mapping $t : \mathcal{Y} \mapsto \mathcal{X}$ from the sample-space $\mathcal{Y}$ to some arbitrary space $\mathcal{X}$.

Put simply, a statistic t is virtually any function of data. Some examples are

- The ordered sample: if $y = (y_1, \ldots, y_n) \in \mathbf{R}^n$, $t(y) = (y_{(1)}, \ldots, y_{(n)})$ is the mapping that arranges the values of y according to their size, i.e. $y_{(1)} = \min(y)$, $y_{(2)} = \min(y \backslash y_{(1)})$ and so on.
- The arithmetic mean: $t(y) = n^{-1} \sum_{i=1}^n y_i$.
- The maximum-likelihood estimator: $t(y) = \text{argmax}_\theta L_y(\theta)$ where the *likelihood* $L_y(\theta) = f_\theta(y)$, the density of Y evaluated at observed data y.
- ...

We will write t for $t(y)$ and T for $t(Y)$, with $Y \sim P_0$. Often, a statistic t provides a summary of data, as for example the arithmetic mean provides a real-valued summary measure of location of a data-set y. We will later give precise definitions of when a statistic *completely* summarises the information contained in data about an unknown parameter θ, in such a case we call t a *sufficient statistic* for θ.

6.2 Point estimation

Now you have defined your model $\mathcal{P}$, observed your data $y \in \mathcal{Y}$, and want to make inference about some real-valued property $\tau = \tau(P_0) \in \mathbf{R}$ of the "true" distribution P_0 that generated data. With a slight abuse of notation, we will write $\tau = \tau(P_0) = \tau(\theta_0)$ or $\tau(P_0) = \tau(F_0)$ in a nonparametric model (which is fair since we insisted on a one-to-one correspondence between $\mathcal{P}$ and Θ). In Example 6.1 we might be interested in $\tau = \tau(\theta_0) = \mu_0$ and in Example 6.2 it might be $\tau = \tau(F_0) = E(Y_1) = \int u \, dF_0(u)$.

Two immediate questions here are:

1. How should I construct a data-based estimate $t = t(y)$ of τ?
2. How should I assess the uncertainty/accuracy of this estimate?

We will leave the former question open to the user for now, this construction can be performed in a multitude of ways, for example based on the maximum-likelihood principle. Instead we focus on the latter.

We would like to report a probabilistic measure of the accuracy of $t(y)$, when viewed as an estimate of τ. The problem is that, after observing y there is no randomness left and we can not make any interesting probabilistic statements about the relation between two fixed (non-random) values $t(y)$ and τ. We can however make statements about the accuracy of the estimator $t(\cdot)$ "in general". On a philosophical level, we imagine the experiment can be repeated under exactly the same circumstances. The data in this "imaginary" experiment is, of course, unobserved and hence it makes sense to model it as a random variable Y distributed accurding to P_0 (rather than just a realization of Y).

The conclusion of the classical/frequentist statistician is that the accuracy of $t(y)$ should be summarized by the distribution of the random variable $\Delta(Y) = t(Y) - \tau = T - \tau$, $Y \sim P_0$. After all, the actual error we are making when reporting $t(y)$, $\Delta(y) = t(y) - \tau$, is a realisation of this random variable. This is one of the central questions of classical statistics; how do we obtain information about the distribution F_Δ of $\Delta(Y)$.

6.3 Asymptotic results

The traditional (as in "before the computer age"), and still commonly used technique of approximating F_Δ is through asymptotic expansions. It turns out that most frequently used estimators $t(\cdot)$ enjoy an asymptotic normal distribution. More specifically, if $T_n = t_n(Y)$, $Y \in \mathbf{R}^n$, it often holds in practical situations that

$$P(\sqrt{n}(T_n - \tau) \leq u) \to \Phi(u/\sigma_t), \tag{6.1}$$

as $n \to \infty$. Hence, $\sqrt{n}(\Delta(Y) = T_n - \tau)$ might be regarded as $N(0, \sigma_t^2)$-distributed for large n. Provided we can also estimate σ_t, an estimate of $F_\Delta(u)$ is given by $\Phi(u/(\hat{\sigma}_t/\sqrt{n}))$.

Example 6.3. Assume $(Y_1, \ldots, Y_n)$ is a vector of independent random variables with common density function f_{θ_0}, $\theta_0 \in \Theta$, a compact subset of $\mathbf{R}$. If $t(y)$ is the Maximum-Likelihood estimate of θ_0,

$$t(y) = \text{argmax}_\theta L_y(\theta) = \text{argmax}_\theta \prod_{i=1}^{n} f_\theta(y_i),$$

then under certain regularity conditions on the mapping $\theta \mapsto f_\theta$ we have that

$$P(\sqrt{n}(T_n - \theta_0) \leq u) \to \Phi(u I_{\theta_0}^{1/2}), \tag{6.2}$$

as $n \to \infty$ and where

$$I_\theta = E\left(\frac{d \log f_\theta(Y_1)}{d\theta}\right)^2 \tag{6.3}$$

is the *Fisher information* of θ contained in Y_1.

Results like the above has played a very important role in the history of statistics, and they still provide some theoretical insight. However, in this course the only thing approaching infinity will be the number of iterations in the Monte-Carlo algorithms.

6.4 Interval estimates

Confidence intervals are tools for making meaningful (?) statements about unknown parameters τ. In a statement "I am 95% confident that the true value τ lies between 3 and 5", "[3,5]" is the *confidence interval* and 95% is the *confidence level*. A common misconception is that "τ lies in [3,5] with probability 0.95", but this statement is meaningless unless you assume τ is a random variable (see the section on Bayesian statistics though) and this is why the less precise notion of "confidence" is used. The theory of confidence intervals was developed by Jerzy Neyman ([**?**]). Note that a "point estimate" is a "point" that estimates a "point" while an "interval estimate" is an "interval" that estimates a "point" (i.e. we are not primarily trying to estimate a fixed interval!). Being a "mathematical" statistician, it is often convenient to disregard the philosophical framework and actual interpretation of the interval [3,5] and the above statement. We will follow this dubious tradition by hiding behind the abstract formulation of our statistical model. The more philosophically inclined reader may consult e.g. [**?**].

A confidence interval for τ given data y is an interval $C = C(y)$ such that $P(\tau \in C(Y)) = 1 - \alpha$ where Y is distributed according to P_0, hence it is an "observation of a random interval" (some authors instead define the object $C(Y)$ or even $C(\cdot)$ to be the confidence interval, others (e.g. Blom et.al. [**?**]) remain slightly confused about the issue). $1 - \alpha$ is the *confidence level* of the interval, often choosen to equal 0.95 or 0.99.

Given an estimator $t(\cdot)$ of τ, a two-sided confidence interval $C(y) = [L(y), U(y)]$ for τ can be derived as follows, first note that

$$\begin{aligned} 1 - \alpha = P(L(Y) \le \tau \le U(Y)) &= P(-U(Y) \le -\tau \le -L(Y)) \\ &= P(T - U(Y) \le \Delta(Y) \le T - L(Y)), \end{aligned}$$

where we in the last step have added $T = t(Y)$ to each side of the inequality. This equality is satisfied if we choose $T - U(Y) = F_\Delta^{-1}(\alpha/2)$ and $T - L(Y) = F_\Delta^{-1}(1 - \alpha/2)$, i.e.

$$C(y) = [L(y), U(y)] = [t(y) - F_\Delta^{-1}(1 - \alpha/2), t(y) - F_\Delta^{-1}(\alpha/2)]. \tag{6.4}$$

Note that it is the upper quantile of $\Delta(Y)$ that appears in lower limit of the interval and the lower quantile in the upper limit. If T enjoys an asymptotic Normal distribution like in (6.1), we might want to use the approximate confidence interval

$$C'(y) = [t(y) - \sigma_t \Phi^{-1}(1-\alpha/2), t(y) - \sigma_t \Phi^{-1}(\alpha/2)]. \tag{6.5}$$

Here the approximation lies in the fact that $P(\tau \in C'(Y))$ will in general not equal $1-\alpha$ exactly, but will be close for "sufficiently large" sample sizes. Moreover, in practise σ_t is usually unknown and needs to be replaces by yet another estimate. The observation to be made is that deriving F_Δ is essential here (we already know $t(y)$).

Remark 6.1. Confidence intervals are not unique, the above construction is a practical example. Given independent observations $y_1, \ldots, y_{20}$ from a continuous univariate density, the "interval" that equals $[y_1, y_1]$ if $y_{20} = \max\{y_1, \ldots, y_{20}\}$ and $[-\infty, \infty]$ else, formally satisfies the requirements of a confidence interval.

6.5 Bias

The bias of an estimator $t(\cdot)$ can be approximated using a combination of an asymptotic result like (6.1) and the Delta-method (yet another asymptotic result). However, the bias is

$$E(T - \tau) = E(\Delta(Y)) = \int u \, dF_\Delta(u) \tag{6.6}$$

and again the error distribution F_Δ provides the answer.

6.6 Deriving the error distribution

Suppose now we are happy with deriving the distribution function $F_\Delta(u) = P(\Delta(Y) \leq u)$, from this we can then derive quantities like the variance $Var(\Delta(Y))$ or the bias $E(\Delta(Y))$ of $t(\cdot)$ and perhaps a confidence interval for τ. There are two problems associated with finding F_Δ, one statistical and one computational. The statistical problem is rather obvious; we do not know the distribution P_0 of Y. The computational problem is that, even if we knew P_0, F_Δ is defined through (assuming $\mathcal{Y} = \mathbf{R}^n$)

$$F_\Delta(u) = \int_{\Delta(y) \leq u} f_0(y_1, \ldots, y_n) \, dy_1 \cdots dy_n, \tag{6.7}$$

where f_0 is the density function corresponding to P_0. This is an integral over $\{y \in \mathcal{Y}; \Delta(y) \leq u\}$, a subset of $\mathbf{R}^n$ (often a rather complicated set). The expression in (6.7) can only be evaluated analytically for a few special choices of t and P_0, the classical example being the model in Example 6.1, if t is the arithmetic mean, $\Delta(y) = \bar{y} - \mu_0$. Then $\Delta(Y)$ has an $N(0, \sigma^2/n)$ distribution.

However, if we know how to simulate from P_0, we can easily construct a Monte-Carlo approximation of $F_\Delta(u)$ by applying Algorithm 4.1 as follows:

1. Draw N samples $y^{(1)}, \ldots, y^{(N)}$, from P_0.
2. Compute $\hat{F}_\Delta(u) = N^{-1} \sum_{i=1}^{N} \mathbf{1}\{\Delta(y^{(i)}) \leq u\}$.

In a practical situation, where P_0 is unknown, a natural idea is to replace P_0 by an estimate $\hat{P}_0$ in the above algorithm. This technique has been popularized as *The Bootstrap*.

Chapter 7

The Bootstrap

7.1 The plug-in principle for finding estimators

Under a parametric model $\mathcal{P} = \{P_\theta; \theta \in \Theta\}$ (or a non-parametric $\mathcal{P} = \{P_F; F \in \mathcal{F}\}$), any real-valued characteristic τ of a particular member P_θ (or P_F) can be written as a mapping from the parameter-space Θ, i.e. $\tau : \Theta \mapsto \mathbf{R}$. If your observations y comes from P_{θ_0} and you have derived an estimate $\hat{\theta} \in \Theta$ of θ_0 (for example by Maximum-Likelihood), it is natural to use $\tau(\hat{\theta})$ as an estimate of $\tau(\theta_0)$. This method for constructing estimates is commonly reffered to as *the plug-in principle*, since we "plug" the estimate $\hat{\theta}$ into the mapping $\tau(\cdot)$.

Example 7.1 (Independent normals cont.)**.** Assume you have observations as in Example 6.1, and you are interested in the probability that a new observation $Y_{n+1} \in \mathbf{R}$ from the same distribution exceeds a level u, i.e. $\tau(\theta_0) = \Phi((u - \mu_0)/\sigma_0)$. If you have derived the Maximum-Likelihood estimate $\hat{\theta} = (\hat{\mu}, \hat{\sigma}^2)$ of θ_0, $\tau(\theta_0)$ should be estimated by $\tau(\hat{\theta}) = \Phi((u - \hat{\mu})/\hat{\sigma})$, following the plug-in principle. Similarily, if you want to estimate the error distribution of $\hat{\mu} = \bar{y}$ then $\tau(\theta_0) = F_\Delta(u) = P(\bar{Y} - \mu_0 \leq u) = \Phi(u/(\sigma_0/\sqrt{n}))$ and the plug-in estimate $\tau(\hat{\theta}) = \Phi(u/(\hat{\sigma}/\sqrt{n}))$.

Maximum-Likelihood estimators generally work well for parametric models, for non-parametric models the natural choice of an estimator for F is the empirical distribution function:

Definition 7.1. The empirical distribution derived from a sample $y = (y_1, \ldots, y_n)$, is the uniform distribution on the set $\{y_1, \ldots, y_n\}$ with distribution function

$$\hat{F}(u) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{y_i \leq u\}. \tag{7.1}$$

As before we interpret the inequalities pointwise if y_i are vectors. Importantly, a random variable $Z \in \mathbf{R}$ distributed according to $\hat{F}$ is discrete and satisfies $P(Z = y_i) = 1/n$, $i = 1, \ldots, n$ if all the values in $\{y_1, \ldots, y_n\}$ are distinct.

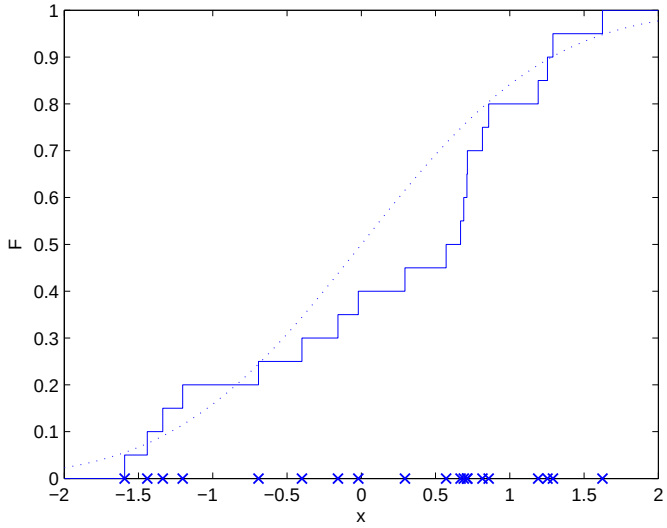

Figure 7.1: Empirical distribution function based on 20 draws from $N(0,1)$, the dotted line is true distribution, draws represented by $\times$:es on the x-axis.

Example 7.2. The plug-in principle also applies with the empirical distribution function as argument. Here we assume $y = (y_1, \ldots, y_n) \in \mathbf{R}^n$ is an observation of $Y = (Y_1, \ldots, Y_n)$, a vector of independent random variables with common distribution function F_0 and that $\hat{F}$ is the empirical distribution function.

- *The expected value:* If we want to estimate

$$\tau(F_0) = E(Y_1) = \int u \, dF_0(u),$$

the plug-in estimator is

$$\tau(\hat{F}) = \int u \, d\hat{F}(u) = n^{-1} \sum_{i=1}^{n} y_i,$$

i.e. the arithmetic mean.

- *The variance:* If we want

$$\tau(F_0) = V(Y_1) = E(E(Y_1) - Y_1)^2 = \int (\int u \, dF_0(u) - v)^2 \, dF(v),$$

the plug-in estimator is

$$\tau(\hat{F}) = \int (\int u \, d\hat{F}(u) - v)^2 \, d\hat{F}(v) = n^{-1} \sum_{i=1}^{n} (\bar{y} - y_i)^2.$$

- *Quantiles:* If we want

$$\tau(F_0) = F_0^{-1}(p) = \inf\{u; F_0(u) \geq p\},$$

the plug-in estimator is

$$\tau(\hat{F}) = \inf\{u; \hat{F}(u) \geq p\} = y_{(\lceil np \rceil)},$$

the $\lceil np \rceil$:th largest value in $\{y_1, \ldots, y_n\}$.

7.2 The plug-in principle for evaluating estimators

Now we have constructed an estimator $t(\cdot)$ for τ using the plug-in principle (or some other principle), and want to asses its uncertainty. As we have seen, estimating the error-distribution F_Δ is often crucial here. But, under model $\mathcal{P}$, F_Δ is of course uniquely determined by θ_0 (or F_0) and it is natural to estimate also this function by its plug-in estimate. I.e. since F_Δ is the distribution function of $\Delta(Y) = t(Y) - \tau(P_0)$, $Y \sim P_0$, we can write its plug-in estimate as F_{Δ^*}, the distribution function of $\Delta(Y^*) = t(Y^*) - \tau(\hat{P})$, $Y^* \sim \hat{P}$. The plug-in principle has its limitations here though. It is only for very special models (like the Normal) and simple estimators t that F_Δ^* can be computed explicitly. We have already seen in (6.7) how computing $F_\Delta(u)$ required an n-dimensional integral over a complicated set. This is where Monte-Carlo integration comes into the picture; even if we can't compute F_{Δ^*} it is often easy to simulate from $\hat{P}$ (and hence from F_{Δ^*}).

Plug-in + Monte-Carlo: The Bootstrap

Algorithm 7.1 (The Bootstrap algorithm)**.**

1. *Estimation:* Use data y to construct an estimate $\hat{P}$ of P_0.
2. *Simulation:* Draw B independent samples $y^{*b} \in \mathcal{Y}$, $b = 1, \ldots, B$ from the distribution $\hat{P}$.
3. *Approximation:* Compute $t^{*b} = t(y^{*b})$, $b = 1, \ldots, B$, and use these values to approximate e.g. one of the following plug-in estimates:
   (a) The error distribution function
   $$P(t(Y^*) - \tau(\hat{P}) \leq u) = F_{\Delta^*}(u) \approx \frac{1}{B}\sum_{b=1}^{B} \mathbf{1}\{t^{*b} - \tau(\hat{P}) \leq u\}.$$
   (b) Quantiles of the error distribution
   $$F^{-1}_{\Delta^*}(p) \approx \Delta^{*(\lceil pB \rceil)} = t^{*(\lceil pB \rceil)} - \tau(\hat{P}),$$
   the $\lceil pB \rceil$:th largest value of t^{*b},$b = 1, \ldots, B$.
   (c) The variance of $t(Y^*)$,
   $$\mathrm{Var}(t(Y^*)) \approx \frac{1}{B}\sum_{b=1}^{B}(t^{*b} - \bar{t}^*)^2.$$
   (d) The bias of t
   $$E(t(Y^*)) - \tau(\hat{P}) \approx \bar{t}^* - \tau(\hat{P})$$
   (e) ...

Most oftenly, in the above algorithm, $\tau(\hat{P}) = t(y)$. If we are unable to derive $\tau(\hat{P})$ explicitly it can be approximated with a similar procedure.

In the following sections we will discuss how to perform Steps 1-2 under some common modelling assumptions. First a basic example.

Example 7.3 (Airconditioning)**.** The file `aircon.mat` contains 12 times between failures for airconditioning equipment in a Boeing aircraft. A reasonable assumption is that data are independent with an Exp(θ) distribution. We estimate θ with the arithmetic mean $t(y) = \bar{y} = 106.4$. In the left panel of Figure 7.2 we have plotted the Exp(106.4) distribution function together with the empirical distribution function, while the fit is not perfect the deviation from the Exponential distribution does not look alarming. We want

to construct an upper 95% confidence interval for θ based on t, looking at (6.4), we want to set $L(y) = 0$, $U(y) = 106.4 - F_\Delta^{-1}(0.05)$ and hence need an estimate of $\tau = F_\Delta^{-1}(0.05)$. Using the bootstrap we proceed as follows in Matlab

```
for b=1:10000
  tstar(b)=mean(exprnd(106.4,1,12));
end
```

which draws a sample of 10000 from T^*, the arithmetic mean of twelve Exp(106.4) variates. A histogram is shown in the right panel of Figure 7.2. The plug-in estimate of θ is obviously $t = 106.4$, hence

```
Delta=tstar-106.4;
```

is a bootstrap draw from the error distribution and τ can be estimated by the empirical 5% quantile

```
quantile(Delta,.05)

ans =

  -44.8750
```

and finally, our confidence interval is given by $[0, 151.3]$. An interval based on Normal approximation is

```
106.4-106.4*norminv(.05)/12^(1/2)

ans =

  156.9217
```

where we used that the Exponential distribution has both mean and standard deviation θ.

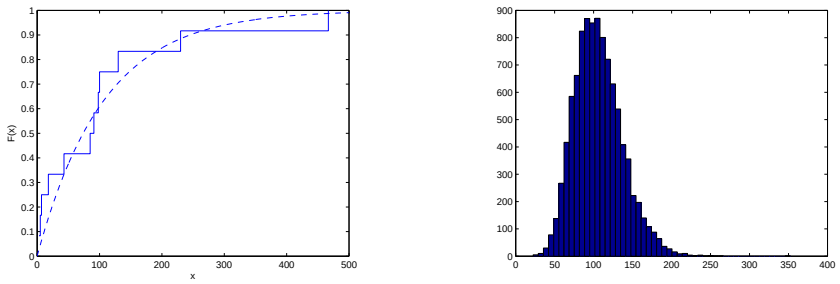

Figure 7.2: Left panel: Fitted Exponential cdf (dashed) and empirical distribution (solid). Right panel: Histogram of $t(y^*)$.

If we are interested in the failure intensity $1/\theta$, this can be estimated by $1/t(y) = 0.0094$. However, this is a biased estimate. The bias is given by $E(1/t(Y)) - 1/\theta$ and can be estimated by the plug-in estimate $E(1/t(Y^*)) - 1/t(y)$, in Matlab

```
bias=mean(1./tstar)-1/106.4

bias =

  8.9401e-004
```

and we might want to report the bias-corrected estimate 0.0094-0.0009=0.0085 rather than our original choice.

7.3 The non-parametric Bootstrap for i.i.d. observations

Here we look at the model corresponding to Example 6.2, though with possibly vector-valued y_i. That is, $y = (y_1, \ldots, y_n)$ is an observation of $Y = (Y_1, \ldots, Y_n)$, where the Y_i:s, $Y_i \in \mathbf{R}^m$, are independent random vectors with common distribution function $F_0 : \mathbf{R}^m \mapsto [0, 1]$. We make no assumptions on the shape of F_0.

We have already introduced the empirical distribution function $\hat{F}$ as the natural estimate of F. Since the distribution function of Y is $\prod_{i=1}^n F_0(u_i)$ we let $\hat{P}$ be the probability distribution with distribution function $\prod_{i=1}^n \hat{F}(u_i)$. This was Step 1. In Step 2 we need to draw samples $y^{*b} = (y_1^{*b}, \ldots, y_n^{*b})$ from this distribution, but this is easy: Firstly, the y_i^{*b}:s should be drawn independently of each other. Secondly, the empirical distribution function is the uniform distribution on $\{y_1, \ldots, y_n\}$, hence we just draw n values from this set randomly with replacement.

Algorithm 7.2 (Drawing y^* from $\prod_{i=1}^n \hat{F}(u_i)$)**.**

1. Let $\hat{F}$ be the empirical distribution function of a sample $y = (y_1, \ldots, y_n)$.
2. Draw $i_1, i_2, \ldots, i_n$ independently from the uniform distribution on the set of integers $\{1, 2, \ldots, n\}$.
3. $y^* = (y_{i_1}, y_{i_2}, \ldots, y_{i_n})$ is now a draw from $\prod_{i=1}^n \hat{F}(u_i)$.

Example 7.4 (Airconditioning cont.)**.** Lets return to the data in Example 7.3, but this time we will be reluctant to assume data are from the Exponential distribution. We will still assume the failure-times are independent and from the same distribution though, and we are still interested in the expected failure time. Hence, failure times are independent with unknown

distribution F_0. We estimate F_0 by the empirical distribution function, plotted in Figure 7.2, and proceed as in Example 7.3, with the difference that new samples are drawn from $\prod_{i=1}^{12} \hat{F}(u_i)$ (here `ac` is the Matlab vector containing data):

```
for b=1:10000
  i=ceil(12*rand(1,12));
  tstar(b)=mean(ac(i));
end
```

a histogram of the sampled means is given in Figure 7.3, it looks slightly wider than the corresponding plot in Figure 7.2 and indeed

```
Delta=tstar-106.4;
quantile(Delta,.05)

ans =

  -52.7750
```

gives a wider confidence interval $[0, 159.2]$. This is not surprising since it is more difficult to estimate the expected value under the larger non-parametric model.

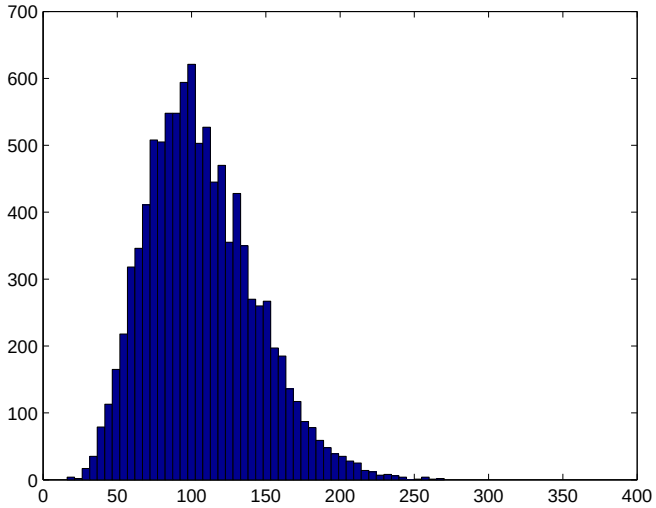

Figure 7.3: Histogram of bootstrap correlations from Example 7.4

Example 7.5. The above method applies almost as easily to slightly more complicated data sets. We consider here a set of data relating two score tests, LSAT and GPA, at a sample of 15 American law schools. Of interest is the correlation between these measurements. The data are given as

LSAT	GPA
576	3.39
635	3.30
558	2.81
578	3.03
666	3.44
580	3.07
555	3.00
661	3.43
651	3.36
605	3.13
653	3.12
575	2.74
545	2.76
572	2.88
594	2.96

and are plotted in Figure 7.4. They are stored in matrix form in the file matrix `law.mat`. Estimating the population correlation by the sample

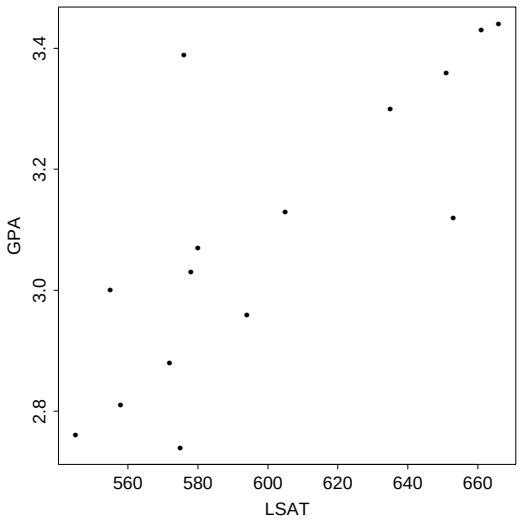

Figure 7.4: Scatterplot of law school data

correlation gives $\hat{\theta} = 0.776$, but how accurate is this estimate? Our model in this case is that the score-pairs are independent realisations of $(X, Y) \sim F_0$, which we estimate by the empirical distribution function. In Matlab, we proceed as follows

```
for b=1:10000
  i=ceil(15*rand(1,15));
```

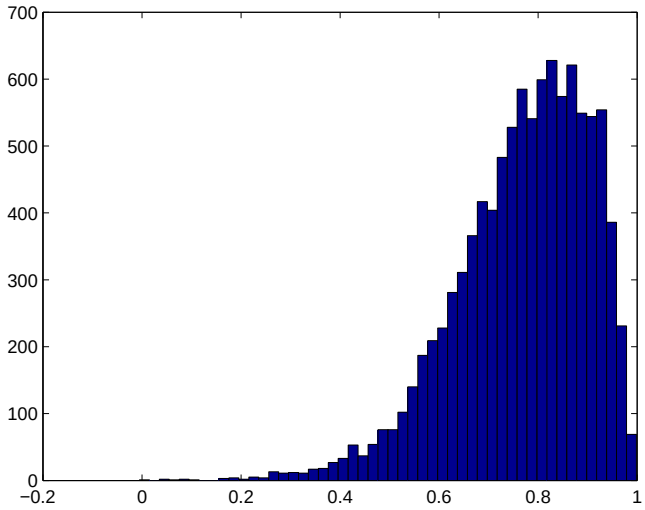

Figure 7.5: Histogram of bootstrap correlations

```
  tstar(b)=corr(law(i,1),law(i,2));
end
```

a histogram of $t(y^*)$ is given in Figure 7.5. If we want to test the hypothesis that the two scores are independent, this can be done by constructing a confidence interval of $\mathrm{Corr}(X, Y)$

```
Delta=tstar-0.776;
quantile(Delta,0.95)

ns=0.1706;
```

This gives a lower interval $[0.776 - 0.1706, 1] = [0.605, 1]$ that does not contain zero. Hence the hypothesis can be rejected.

Example 7.6 (Ph-levels)**.** Figure 7.6 gives histograms of historical and current measurements of Ph levels at each of 149 lakes in Wisconsin. The data are stored in `ph.mat` as the vectors `ph1` and `ph2` respectively. Historical data from 25 of the lakes are missing, so paired sample comparisons are not possible. A reasonable model is that historical and current measurements are all independent and with distribution functions F_0 and G_0 respectively, hence we have parameter-space $\Theta = \mathcal{F} \times \mathcal{F}$, where $\mathcal{F}$ is the set of distribution functions on $\mathbf{R}$ and P_0 is the distribution with distribution function $\prod_{i=1}^{124} F_0(y_i) \prod_{i=125}^{273} G_0(y_i)$, where $y_1, \ldots, y_{124}$ and $y_{125}, \ldots, y_{273}$ correspond to the historical and current measurements respectively. It is of some interest to examine whether the Ph-levels have increased, and hence we compute the difference in medians of the historical and current populations, $t(y) = \hat{G}^{-1}(1/2) - \hat{F}^{-1}(1/2)$ which turns out to be 0.422. This suggests

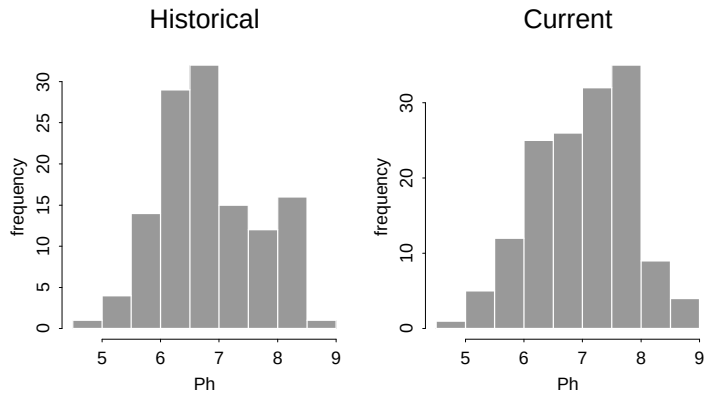

Figure 7.6: Histograms of historical and current Ph levels in Wisconsin lakes

that either there has been an increase or that $t(y)$ has overestimated the true value $G_0^{-1}(1/2) - F_0^{-1}(1/2)$, to assess the latter we want to examine the distribution of $t(Y)$ which we approximate by the distribution of $t(Y^*)$ using the Bootstrap. Each bootstrap simulation consists of simulating from the empirical distribution function of the two separate samples, obtaining the median of each and differencing.

```
for b=1:10000
    ih=ceil(124*rand(1,124));
    ic=ceil(149*rand(1,149));
    tstar(b)=median(ph2(ic))-median(ph1(ih));
end
```

which gives the histogram of bootstrap differences in Figure 7.7, the fact that all simulated values exceed 0 give some extra support for the hypothesis that Ph-levels have increased. The histogram looks rather rugged, would you get a smoother histogram if you were bootstrapping the difference in means? Why?

The limitation of non-parametric models is that we need a resonably large number of independent observations from each of the unknown distribution functions involved in the model in order to construct an accurate estimate of $\hat{P}$.

7.4 Bootstrapping parametric models

We have already seen an example the Bootstrap in parametric models in Example 7.3, here we will look at some more complicated models.

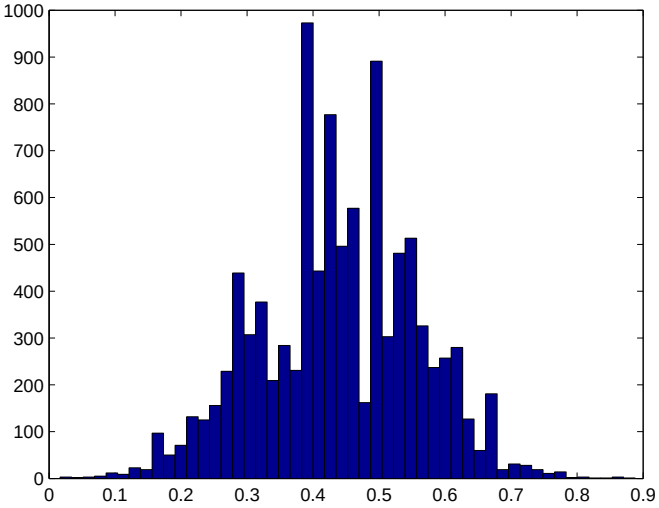

Figure 7.7: Histogram of simulated median differences

Example 7.7 (Space shuttle challenger)**.** In 1986, the space shuttle Challenger exploded during take off, killing the seven astronauts aboard. It is believed that the explosion was caused by the failure of an *O-ring* (a rubber ring that seals parts of the ship together), and that the failure was caused by the cold weather at the time of launch (31° F). In the file `oring.mat`, you will find temperature (in Fahrenheit) and failure data from 23 shuttle launches, where 1 stands for O-ring failure and 0 no failure. See Figure 7.8. We want to model the failure-data as a function of temperature, a common model in this context is the *probit* regression model that asserts that observations y_i are independent Bernoulli$(m(t_i))$, where t_i is temperature (which we will assume fixed) and $m(t) = \Phi(\beta_0 + \beta_1 t)$ (this is just a rather arbitrary way of mapping a linear regression to the unit interval). Hence, $\theta = (\beta_0, \beta_1) \in \mathbf{R}^2 = \Theta$. Further, given an estimate $\hat{\theta}$ it is straightforward to generate Bootstrap draws y^* from $P_{\hat{\theta}}$. The model is an example of a *generalised linear model* (GLM), a class of generalisations to the usual linear regression model with Normal errors. Matlab's `glmfit` can be used to estimate θ.

```
b=glmfit(chall(:,1),[chall(:,2) ones(23,1)],'binomial','probit');
```

if we are interested in the failure-probability at 65F, an estimate is given by

```
normcdf(b(1)+b(2)*65)
ans =

    0.4975
```

and to assess the accuracy, we construct a 95% confidence interval using the parametric Bootstrap

```
for i=1:1000
    ystar=binornd(1,normcdf(b(1)+b(2)*chall(:,1)));
    bstar(i,1:2)=glmfit(chall(:,1),[ystar ones(23,1)],'binomial','probit');
end
tstar=normcdf(bstar(:,1)+bstar(:,2)*65);
Delta=tstar-.4975;
C=[.4975-quantile(Delta,.975) .4975-quantile(Delta,.025)]
C =

    0.0792    0.7484
```

which turns out quite wide.

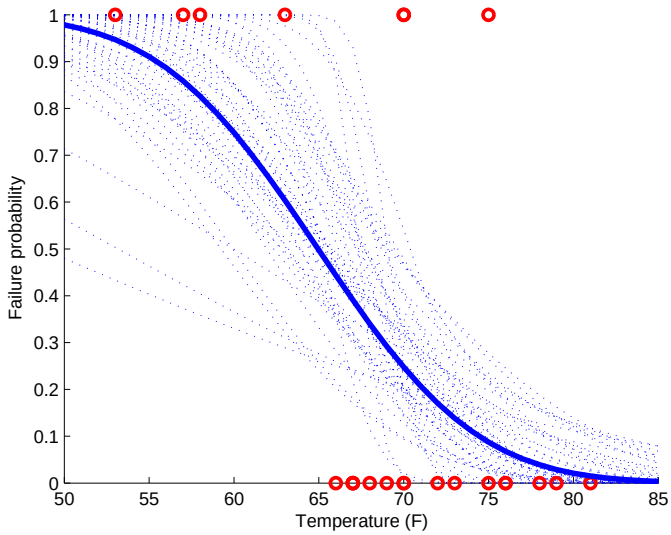

Figure 7.8: Data for the Challenger example (circles), fitted regression line (thick solid) and 50 draws from its Bootstrap distribution (dotted)

Example 7.8 (Hormone levels)**.** The file `hormone.mat` contains a time-series of lutenizing hormone level in bood-samples. A simple model for time-series is the Gaussian first-order autoregressive (AR(1)) process defined recursively through

$$Y_i = X_i + \mu, \quad X_i = \alpha X_{i-1} + \epsilon_i, \tag{7.2}$$

where ϵ_i, $i = 1, \ldots, n$ are independent $N(0, \sigma_2)$ and$X_0 \sim N(0, \sigma^2/(1-\alpha^2))$, $|\alpha| < 1$. Under this specification, $Y = (Y_1, \ldots, Y_n) \sim N((\mu, \ldots, \mu), \Sigma)$, where Σ has elements $\Sigma_{ij} = \alpha^{|i-j|}\sigma^2/(1-\alpha^2)$ and $\theta = (\mu, \sigma^2, \alpha) \in \mathbf{R} \times \mathbf{R}^+ \times (-1, 1) = \Theta$. The parameters can be estimated by their Maximum-Likelihood estimators, but we choose a simpler approach and estimate μ by the empirical mean, α by the empirical first order autocorrelation and σ^2 by the empirical variance multiplied by $1 - \hat{\alpha}$:

```
mhat=mean(y);
ahat=corr(y(1:47),y(2:48));
s2hat=var(y)*(1-ahat);
```

$\hat{P}$ is now $N(\hat{\mu}, \hat{\Sigma})$, and we can simulate from this distribution using the Choleski method (or Matlabs `mvnrnd` which is an implementation of the same). The covariance matrix $\hat{\Sigma}$ can be computed by

```
Shat=s2hat/(1-ahat)*ahat.^toeplitz(0:47,0:47);
```

If we are interested in the properties of our estimate of α, we proceed as follows

```
for b=1:1000
  ystar=mvnrnd(mhat*ones(1,48),Shat)';
  tstar(b)=corr(ystar(1:47),ystar(2:48));
end
```

In the right panel of Figure 7.9 we have plotted a histogram of T^* and in Figure 7.10 four realisations of the timeseries Y^*.

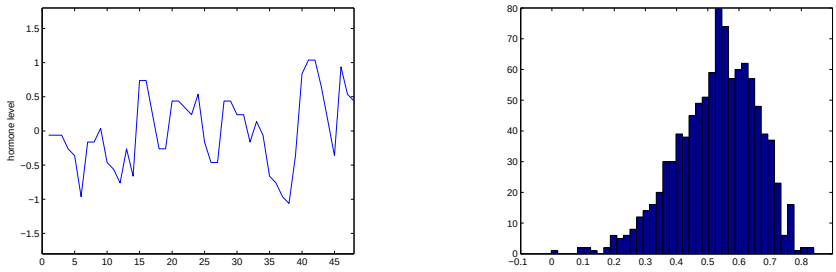

Figure 7.9: Left panel: Hormone level data-set. Right panel: Histogram of $T^* = \hat{\alpha}^*$.

7.5 Bootstrap for semi-parametric models

A semi-parametric model can be viewed as a non-parametric model with some additional parametric structure. This allows us to relieve some of the rather restrictive model assumptions in Section 7.3.

7.5.1 Regression models

A regression model tries to establish a functional relationship between a set of responses $y = (y_1, \ldots, y_n) \in \mathbf{R}^n$ and a set of covariates/design-variables $x = (x_1, \ldots, x_n) \in \mathbf{R}^d \times \mathbf{R}^n$. There are two classes of such models, in a regression model with *fixed design* we view the covariates x as fixed or user-defined quantities. In a model with *random design*, both x and y are observations of random variables. An example of the first could be that

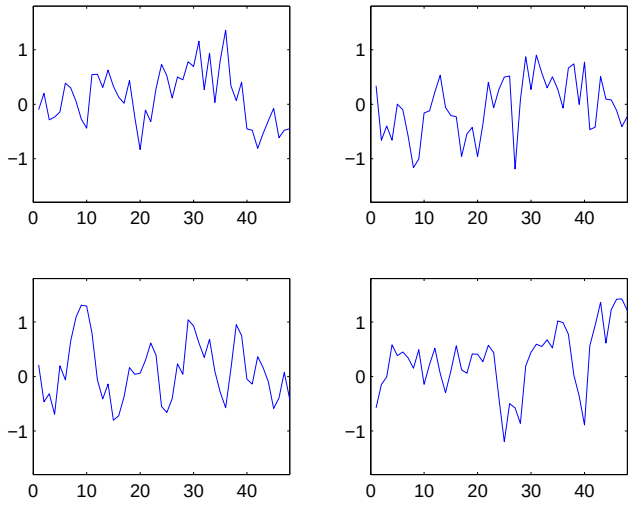

Figure 7.10: Four realisations of Y^* for the hormone data

we measure the daily temperature over a period of time, here y_i could be measured temperature on day i and $x_i = i$. An example of the second could be that (x_i, y_i) are measured weight an height of a randomly chosen individual, and we want to model weight as a function of height.

Fixed design linear regression

Consider data $y = (y_1, \ldots, y_n)$ generated from the model

$$Y_i = \alpha + \beta x_i + \epsilon_i, i = 1, \ldots, n, \tag{7.3}$$

where β and x_i might be a row and column vector respectively, the residuals ϵ_i are i.i.d. from F_0 and $E(\epsilon_i) = 0$. In the absence of further assumptions on the distribution of the residuals, this is a semi-parametric model with unknown parameter $\theta_0 = (\alpha_0, \beta_0, F_0)$. Moreover, P_0 is the distribution corresponding to distribution function

$$F(y) = \prod_{i=1}^{n} F_0(y_i - \alpha - \beta x_i).$$

How do we find an estimate $\hat{P}$ of this distribution? After all, we actually observe the residuals ϵ_i and hence can't estimate F_0 by the empirical distribution function directly. The solution is to start with the parametric part, i.e. we first find estimates $(\hat{\alpha}, \hat{\beta})$ of α and β, e.g. by least-squares

$$(\hat{\alpha}, \hat{\beta}) = \text{argmin}_{(\alpha,\beta)} \sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2. \tag{7.4}$$

Then we compute the *approximate residuals* $\hat{\epsilon}_i = y_i - \hat{\alpha} - \hat{\beta}x_i$. These will be approximately independent realisations from F_0 and hence we can estimate the distribution by

$$\hat{F}(u) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{\hat{\epsilon}_i \leq u\},$$

the empirical distribution of the approximate residuals. Sampling Y^* now proceeds as follows

1. Draw $i_1, i_2, \ldots, i_n$ independently from the uniform distribution on the set of integers $\{1, 2, \ldots, n\}$.

2. $y^* = (y_1^*, \ldots, y_n^*) = (\hat{\alpha} + \hat{\beta}x_1 + \hat{\epsilon}_{i_1}, \ldots, \hat{\alpha} + \hat{\beta}x_n + \hat{\epsilon}_{i_n})$ is now a draw from Y^* with distribution $\prod_{i=1}^{n} \hat{F}(y_i - \hat{\alpha} - \hat{\beta}x_i)$.

Example 7.9. We illustrate the above residual Bootstrap on a simulated data-set following the slightly more complicated model

$$Y_i = \alpha + \beta x_i + x_i\epsilon_i, i = 1, \ldots, n, \tag{7.5}$$

where the ϵ_i are independent from F_0, i.e. the size of the "measurement errors" is proportional to x_i. Data is plotted in Figure 7.11.

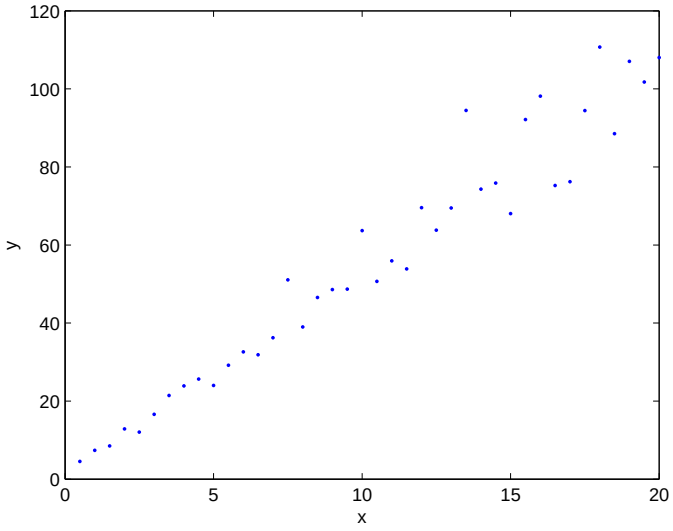

Figure 7.11: Data from the model in 7.5

In order to estimate F_0, we need to construct approximate residuals from this distribution. First we estimate α and β using least-squares (though a weighted version would have been more appropriate in this case). Matlab gives

```
polyfit(x,y,1)

ans =

    5.2794    1.2027
```

where the first is slope β and second intercept α. Residuals are now given by $\hat{\epsilon}_i = (y_i - \hat{\alpha} - \hat{\beta}x_i)/x_i$, in Matlab

```
ehat=(y-1.2027-5.2794*x)./x;
```

and Bootstrap proceeds by

```
for b=1:1000
 i=ceil(40*rand(1,40));
 estar=ehat(i);
 ystar=1.2027+5.2794*x+estar.*x;
 tstar(b,1:2)=polyfit(x,ystar,1);
end
```

In Figure 7.12 we have plotted 20 draws from the regression line.

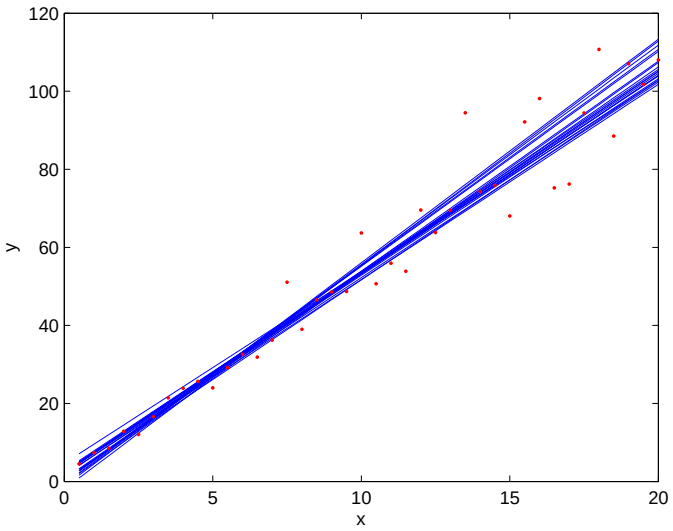

Figure 7.12: Data and regression lines estimated from each of 20 draws from Y^*.

Random design linear regression

In random design regression, (x_i, y_i), $i = 1, \ldots, n$, are independent draws from a bivariate distribution $F_0(x, y)$, with the property

$$E(Y|X = x) = \alpha + \beta x. \tag{7.6}$$

Bootstrap from this model is equivalent to that of Example 7.5, i.e. with $\hat{P}$ corresponding to the distribution $\prod_{i=1}^{n} \hat{F}(x_i, y_i)$, where $\hat{F}$ is the (bivariate) empirical distribution function..

Example 7.10 (Bodies and brains)**.** The left panel of Figure 7.13 shows average brain weight (g) against body weight (kg) for 62 species of mammals, in the right panel the same data is plotted on logarithmic scales, showing an approximate linear relation. Hence, with y denoting log-brain weight

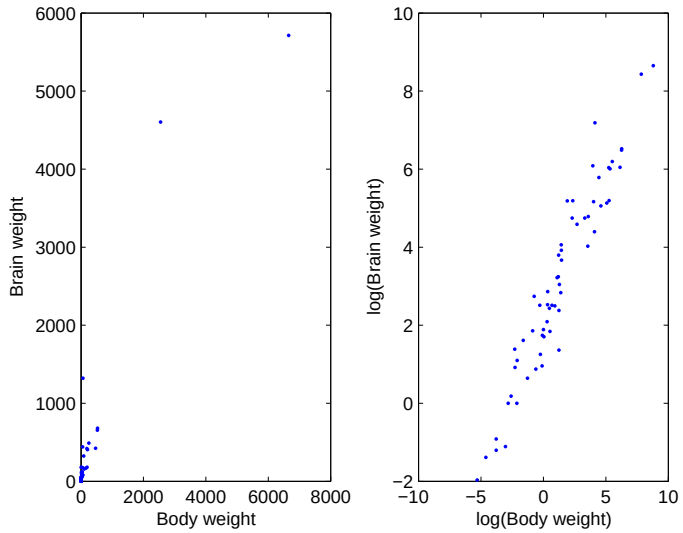

Figure 7.13: Average body and brain weights for 62 mammals.

and x log-body weight we might use the model in (7.6). Bootstrapping the regression lines proceeds as collows (c.f. Example 7.5)

```
x=log(mammals(:,1)); y=log(mammals(:,2));
for b=1:1000
  i=ceil(62*rand(1,62));
  ystar=y(i);
  xstar=x(i);
  tstar(b,1:2)=polyfit(xstar,ystar,1);
end
```

Figure 7.14 shows a few regression lines estimated from draws from (X^*, Y^*).

7.5.2 Time-series

Here we have another look at the lutenizing hormone data from Example 7.8, this time without assuming a Normal distribution. Given the AR(1)

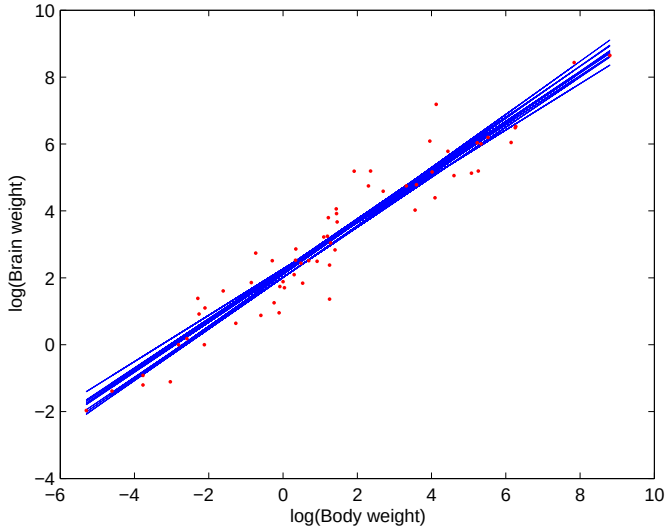

Figure 7.14: Bootstrap regression lines for the mammals data.

structure in (7.2) and estimates of α, μ we can, just as for the fixed design regression problem, construct approximate residuals

$$\hat{\epsilon}_i = (y_i - \hat{\mu}) - \hat{\alpha}(y_{i-1} - \hat{\mu}), i = 2, \ldots, n. \tag{7.7}$$

In Figure 7.15 we have made a Normal probability plot and a histogram of the approximate AR(1) residuals for the hormone data. It seems that the assumption of Normality made in Example 7.8 can be questioned, and hence we assume the residuals to be independent from F_0. To fully specify

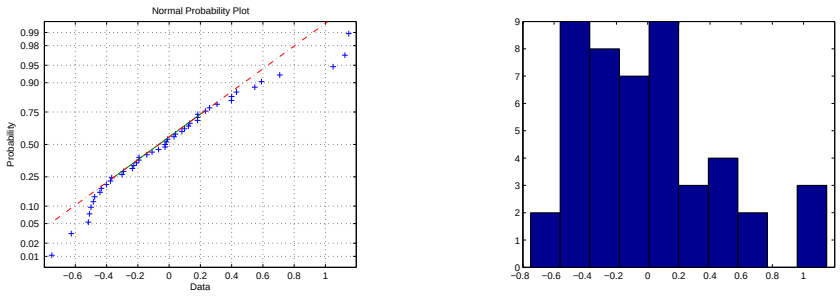

Figure 7.15: Left panel: Normal probability plot of approximate AR(1) residuals for hormone data. Right panel: Histogram of the same residuals.

the model, we assume the time-series is stationary, i.e. the observations y_i are marginally all from the same distribution G_0. This assumption is important,

since it allow us to draw a starting value y_1^* for the series. Hence, we proceed as follows given approximate residuals $\hat{\epsilon}_2, \ldots, \hat{\epsilon}_n$:

1. Draw i_1 from the uniform distribution on the set of indices $\{1, 2, \ldots, n\}$ and $i_2, \ldots, i_n$ independently from the uniform distribution on $\{2, \ldots, n\}$.
2. Set $y_1^* = y_{i_1}$ and $\epsilon^* = (\epsilon_2^*, \ldots, \epsilon_n^*) = (\hat{\epsilon}_{i_2}, \ldots, \hat{\epsilon}_{i_n})$.
3. Recursively compute $y_j^* = \hat{\alpha}(y_{j-1}^* - \hat{\mu}) + \epsilon_j^* + \hat{\mu}$ for $j = 2, \ldots, n$.

In Matlab, given the same parameter estimates as in Example 7.8, a realisation of Y^* is given by

```
ehat=(y(2:48)-mhat)-ahat*(y(1:47)-mhat);
i(1)=ceil(48*rand);
i(2:48)=ceil(47*rand(1,47));
estar=ehat(i(2:48));
ystar(1)=y(i(1));
for j=2:48
  ystar(j)=ahat*(ystar(j-1)-mhat)+estar(j)+mhat;
end
```

Figure 7.16 shows four realisations of Y^*.

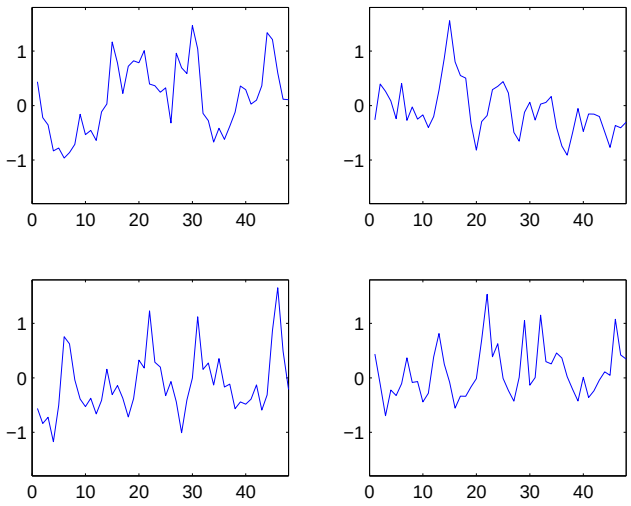

Figure 7.16: Realisations of Y^* for the hormone data using a semiparametric model.

We can also use the bootstrap to predict future observations of the series (even if this might not be too interesting for the current data-set). This simply amounts to drawing vectors $(y_{49}^*, \ldots, y_N^*)$ using the recursion and with the observed y_{48} as starting value. To predict the next five values we use

```
i=ceil(47*rand(1,5));
estar=ehat(i);
ystar(1)=ahat*(y(48)-mhat)+estar(j)+mhat;
for j=2:5
  ystar(j)=ahat*(ystar(j-1)-mhat)+estar(j)+mhat;
end
```

In Figure 7.17 we have plotted a few draws from this predictive distribution.

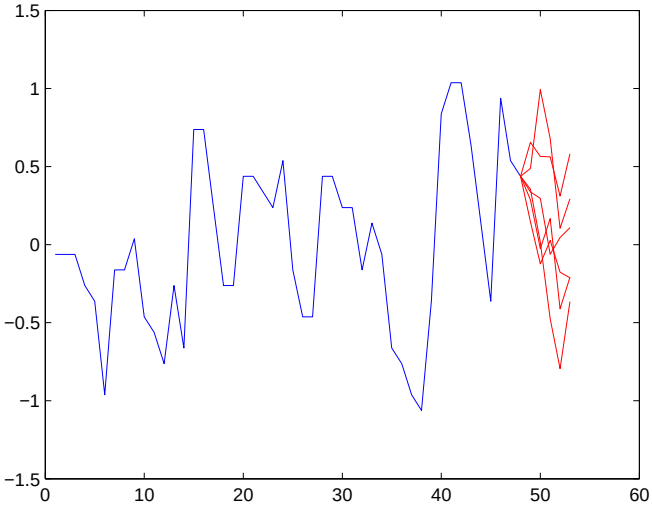

Figure 7.17: Bootstrap predicted values.

Chapter 8

Testing statistical hypothesises

A statistical hypothesis is a statement about the distributional properties of data. Given our statistical model $\mathcal{P} = \{P_\theta; \theta \in \Theta\}$, data y from $P_0 = P_{\theta_0}$, it can hence be stated as $H_0 : \theta_0 \in \Theta_0 \subseteq \Theta$. This is the *null-hypothesis*. One usually distinguishes between *simple* and *composite* hypothesises. A simple hypothesis specifies a single distribution for data (Θ_0 contains only one value) while a composite specifies more than one candidates (Θ_0 is a set of several values). For example, if $\mathcal{P}$ is the set of $N(\theta, 1)$ distributions, $H_0 : \theta = 0$ is simple while $H_0 : \theta > 0$ is composite. More examples of composite hypothesises are $H_0 : \theta = 0$ in the model of $N(\theta, \sigma^2)$ distributions (σ^2 remains unspecified by the hypothesis and $\Theta_0 = \{0\} \times \mathbf{R}^+$) and if the model contains all bivariate distributions $F(x, y)$, H_0 : "X and Y are independent" corresponds to the subset of distributions in $\mathcal{P}$ such that $F(x, y) = H(x)G(y)$ for two univariate distribution functions H and G. The hypothesis $H_A : \theta_0 \in \Theta \backslash \Theta_0$ is usually refered to as the *alternative hypothesis.*

A statistical *test* of a hypothesis H_0 specifies a *rejection region* $R \subseteq \mathcal{Y}$ and a *rule* stating that if data $y \in R$, H_0 should be rejected (and if $y \in \mathcal{Y} \backslash R$, H_0 is not rejected). The question as to whether a hypothesis can ever be *accepted* is a philosophical one; in this course we will only attempt to find evidence *against* hypothesises.

The *significance level* of a test with rejection region R is defined as

$$\alpha = \sup_{\theta \in \Theta_0} P_\theta(Y \in R) = \sup_{\theta \in \Theta_0} \int_R f_\theta(y)\, dy, \tag{8.1}$$

that is, the probability of rejecting a true null-hypothesis maximized over all probability distributions satisfying the null-hypothesis. Typically α is a fixed small value, e.g. $\alpha = 0.05$.

8.1 Testing simple hypothesises

In the case of a simple hypothesis, the distribution of data under the null-hypothesis is completely specified and we write

$$\alpha = P(Y \in R|H_0) \tag{8.2}$$

for the level (8.1) of the test. To define a rejection region R, a common approach is to first define a test-statistic $t : \mathcal{Y} \mapsto \mathbf{R}$. $t(y)$ provides a real-valued summary of observed data y, and it is usually designed in such a way that "large" values of $t(y)$ will constitute evidence against H_0. Hence, a natural rejection region is of the form $R = \{y; t(y) > r\}$, that rejects H_0 for large values of $t(y)$. The question remains how to choose r (i.e. what is "large") in order to to achive a pre-specified level α. To achieve this, we introduce the P-value as the statistic $p : \mathcal{Y} \mapsto [0, 1]$ defined by

$$p(y) = P(t(Y) \geq t(y)|H_0), \tag{8.3}$$

i.e. the probability of, under H_0, observing a larger value of the test-statistic than the one we actually observed. The below simple lemma allow us to determine the distribution of the random variable $p(Y)$:

Lemma 8.1.1. *If $X \in \mathbf{R}$ is a random variable with a continuous distribution function F, then the random variable $Z = 1 - F(X)$ has an uniform distribution on $[0, 1]$.*

Hence, if F_{T_0} is the distribution function of $t(Y)$ under the null-hypothesis, then $p(Y) = 1-F_{T_0}(t(Y))$, which has an uniform distribution under the null-hypothesis. Note also that the result only applies to continuous distribution functions in general.

This result provides a convenient way of constructing rejection regions; we set $R = \{y; p(y) \leq \alpha\}$. Since $p(Y)$ is uniformly distributed under H_0, $P(Y \in R|H_0) = \alpha$ as required by a level-α test. Hence, to test a hypothesis, we simply compute $p(y)$ and reject the hypothesis if this value is less than α. The mechanic "rejection" involved hypothesis testing has often been critisised, an alternative approach favoured by many is to report the P-value rather than the decision, since the P-value itself is a measure of the accuracy (or rather the lack thereof) of H_0. The scientist community is then left to interpret this evidence provided by the statistician.

What is not so simple here is to compute $p(y)$, since this involves the distribution F_{T_0} which is not analytically available in general. However, if we can simulate new samples from the null-hypothesis, $p(y)$ can easily be approximated by Monte-Carlo integration.

Algorithm 8.1 (Monte-Carlo test of a simple hypothesis)**.**

1. Draw N samples $y^{(1)}, \ldots, y^{(N)}$, from the distribution specified by H_0.
2. Compute $t^{(i)} = t(y^{(i)})$ for $i = 1, \ldots, N$.
3. Compute $\hat{p}(y) = N^{-1} \sum_{i=1}^{N} \mathbf{1}\{t^{(i)} \geq t(y)\}$.
4. Reject H_0 if $\hat{p} \leq \alpha$.

Traditionally, P-values have been approximated through asymptotic expansions of the distribution of $t(Y)$. One advantage of Monte-Carlo based tests is that we have almost complete freedom in choosing our test-statistic (without having to proove an asymptotic result).

Example 8.1 (Normal mean)**.** If we have observed $y_1, \ldots, y_n$ independently from $N(\mu_0, \sigma_0^2)$, where σ_0^2 is known, we may want to test $H_0 : \mu_0 = 0$. A natural test-statistic here is $t(y) = |\bar{y}|$ and testing proceed by

1. Draw $y^{(1)}, \ldots, y^{(N)}$, N vectors of n independent $N(0, \sigma^2)$ variables.
2. Compute $t^{(i)} = \bar{y}^{(i)}$, $i = 1, \ldots, N$.
3. Compute $\hat{p}(y) = N^{-1} \sum_{i=1}^{N} \mathbf{1}\{t^{(i)} \geq t(y)\}$.
4. Reject H_0 if $\hat{p} \leq \alpha$.

Randomized tests and the problem with discretely distributed test-statistics

When the distribution of $t(Y)$ is discrete, the above procedure does not guarantee a test that rejects a true null-hypothesis with probability α. Consider, for example the (slightly pathological) case where we have one observation from a Bernoulli(p) distribution, and want to test $H_0 : p = 1/2$. In this case, no matter how we define the rejection region R, it will contain $Y \sim$ Bernoulli(1/2) with probability 0, 1/2 or 1. Hence, for example, we can't define a level $\alpha = 0.05$ test. We *can* construct a level 0.05 test by a randomization argument however. By augmenting the sample space, i.e. defining $\mathcal{Y}' = \mathcal{Y} \times \mathbf{R}$, where the "new" data-set is $y' = (y, z) \in \mathcal{Y}'$ and z is a draw from, say, $N(0, \sigma^2)$, the test-statistic $t'(y') = t(y) + z$ will now have a continuous distribution and the procedure applies again. Since the result of such a procedure depends on z, which we just drew using a random number generator independently of data, randomized tests are not very popular in practise but rather employed as technical devices in theory.

Example 8.2 (Flipping a coin)**.** To test whethter a coin was biased, 100 independent coin-flips were performed out of which 59 turned out to be

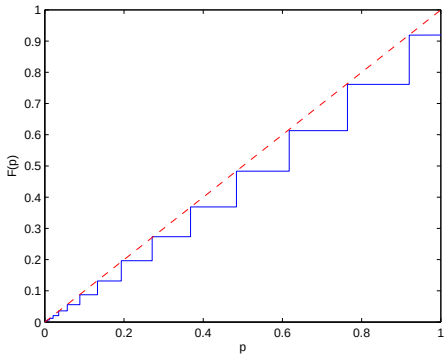

Figure 8.1: The null-distribution function of $p(Y)$ for the coin-flipping example (solid line).

"tails". If the coin was unbiased, $y = 59$ could be regarded as a draw from a Bin(100,1/2) distribution, hence we choose this as out null-hypothesis. A reasonable choice of test-statistic is $t(y) = |y/100 - 1/2|$, since this can be expected large if the true probability of "tails" deviated from 1/2. To estimate the P-value, we perform 1000 draws from Bin(100,1/2), $y^{(1)}, \ldots, y^{(1000)}$, and computed $t(y^{(1)}), \ldots, t(y^{(1000)})$. In a particular application, 75 of the $t(y^{(i)})$ were larger than the observed $t(59)$. This gives an approximate P-value of 75/1000 which is larger than 0.05. Hence, we can not reject the null-hypothesis at this level. In Matlab:

```
t=abs(59/100-0.5);
Y=binornd(100,.5,1,1000);
tY=abs(Y/100-0.5);
p=mean(tY>=t)

p =

    0.0750
```

For this simple example, we can calculate the P-value exactly as $p(59) = P(Y \leq 41) + P(Y \geq 59) = 0.088\ldots$. In Figure 8.1 we have plotted the distribution function of $p(Y)$. As you can see, it is not exactly uniform (the dashed line is the uniform), but fairly close to uniform in its left tail. The true level of the test is $\alpha \approx 0.035$.

8.2 Testing composite hypothesises

Unfortunately, most null-hypothesises of interest are composite, and for a composite hypothesis the P-value (8.3) is no longer well-defined since it might take different values in different parts of Θ_0. There are three basic strategies for dealing with this problem:

1. Pivotal tests; here we find a *pivotal* test-statistic t such that $t(Y)$ has the same distribution for all values of $\theta \in \Theta_0$.
2. Conditional tests; here we convert the composite hypothesis into a simple by conditioning on a *sufficient statistic*.
3. Bootstrap tests; here we replace the composite hypothesis Θ_0 by the simple $\{\hat{\theta}\}$, $\hat{\theta} \in \Theta_0$.

8.2.1 Pivot tests

A *pivotal statistic* is a statistic t such that the distribution of $t(Y)$, $Y \sim P_\theta$, is the same for all values $\theta \in \Theta_0$. Hence, in Step 2 of Algorithm 8.1, we can simulate from any such P_θ.

Example 8.3 (Testing for an equal mean, Normal case)**.** Assume we have two independent samples, $y_1, \ldots, y_n$ from $N(\mu_y, 1)$ and $x_1, \ldots, x_m$ from $N(\mu_x, 1)$ and we want to test $H_0 : \mu_y = \mu_x$. Here the null-hypothesis contains one parameter that remains unspecified ($\mu_y = \mu_x$). Since we can write $y_i = u_i + \mu_y$ and $x_i = v_i + \mu_x$, for independent draws u_i and v_i from $N(0, 1)$ the statistic t given by

$$t(x, y) = |\frac{1}{n}\sum_{i=1}^{n} y_i - \frac{1}{m}\sum_{i=1}^{m} x_i| = |\frac{1}{n}\sum_{i=1}^{n} u_i - \frac{1}{m}\sum_{i=1}^{m} v_i + (\mu_y - \mu_x)|, \quad (8.4)$$

is independent of μ_y and μ_x under the null-hypothesis. Hence, a test might proceed as follows

1. Draw N samples $y^{(1)}, \ldots, y^{(N)}$, each being a vector of n independent $N(37, 1)$ draws.
2. Draw N samples $x^{(1)}, \ldots, x^{(N)}$, each being a vector of m independent $N(37, 1)$ draws.
3. Compute $t^{(i)} = t(x^{(i)}, y^{(i)})$, $i = 1, \ldots, N$ with t as defined in (8.4).
4. Compute $\hat{p} = N^{-1} \sum_{i=1}^{N} \mathbf{1}\{t^{(i)} \geq t(x, y)\}$.
5. Reject H_0 if $\hat{p} \leq \alpha$.

Of course, the value "37" can be replaced by any value in **R** as long as it is the same in Steps 1 and 2.

Tests based on ranks

Example 8.4 (Ph-levels cont.)**.** Consider the data of Example 7.6, where it was relevant to test for an increased Ph-level. If we again denote by F_0 the distribution of historical measurements, G_0 the distribution of the current and assume all measurements independent, a natural null-hypothesis is that of $H_0 : F_0 = G_0$. This is a complex hypothesis since F_0 and G_0 are unknown. We represent data as $y = (y_1, \ldots, y_{273})$ where $y_1, \ldots, y_{124}$ are

historical and $y_{125}, \ldots, y_{273}$ current measurements. To find a pivot, we let r_i be the *rank* of y_i, i.e. if the components of y are sorted with respect to size, then y_i comes in r_i:th place. What is important to note here is that if the null-hypothesis is true, the vector $(r_1, \ldots, r_{273})$ can be seen as a draw from the uniform distribution on the set of all permutations of the vector $(1, 2, \ldots, 273)$. Hence, any statistic based on r alone is pivotal. Since there was some prior suspicion that the Ph-level had increased, this would correspond to the current measurements having larger ranks than what could be expected from the null-distribution. Hence, we choose as a test-statistic the average difference in ranks

$$t(y) = \frac{1}{149}\sum_{i=125}^{273} r_i - \frac{1}{124}\sum_{i=1}^{124} r_i.$$

In Matlab the test is implemented as follows

```
[dummy,r]=sort(y);
t=mean(r(125:273))-mean(r(1:124));
for i=1:1000
  R=randperm(273);
  T(i)=mean(R(125:273))-mean(R(1:124));
end
phat=mean(T>=t)

phat =

    0.0770
```

and this test could not find any hard evidence of an increase.

Pivotal tests of the above form are (naturally) called rank-based tests. They are generally applicable when the null-hypothesis is *exchangable*, i.e. when $(Y_{i_1}, \ldots, Y_{i_n})$ has (under the null hypothesis) the same distribution for all permutations $(i_1, \ldots, i_n)$ of the vector $(1, \ldots, n)$. They are of somewhat limited efficiency in many situations since there is a loss of information in the transformation from y to r.

Example 8.5 (Hormone levels cont.)**.** In Example 7.8 we assumed a time-series model for the measurements of lutenizing hormone levels. The fact that consequtive measurements are really dependent of eachother may be questioned. Can we reject the null-hypothesis H_0 :" Measurements are independent from F_0"? If there is (positive) serial dependence, the ranks of consequtive observations will be closer to eachother than under the null-hypothesis. Hence, we choose as a test statistic

$$t(y) = \frac{1}{\sum_{i=2}^{48} |r_i - r_{i-1}|}.$$

In Matlab

```
[dummy,r]=sort(hormone);
t=1/sum(abs(r(2:48)-r(1:47)));
for i=1:1000
  R=randperm(48);
  T(i)=1/sum(abs(R(2:48)-R(1:47)));
end
phat=mean(T>=t)
phat =

    0.0670
```

and no significant deviation from the null-hypothesis was found with this test either.

8.2.2 Conditional tests

Consider the usual setup with data $y \in \mathcal{Y}$ a realisation of $Y \sim P_{\theta_0}$. A *sufficient statistic* $s(y)$ for θ is a statistic that summarises all information available on θ in the sample.

Definition 8.1. A statistic $s(\cdot)$ is said to be sufficient for a parameter θ if the distribution of $Y|s(Y) = s(y)$ does not depend on θ (for any y).

Obviously, $s(y) = y$ is always a sufficient statistic. The following theorem provides a convenient way to check whether a particular statistic is sufficient:

Theorem 8.1. *A statistic t is sufficient for θ if and only if the likelihood factorises as*

$$f_\theta(y) = h(y)k(t(y), \theta), \theta \in \Theta, \quad y \in \mathcal{Y}, \tag{8.5}$$

i.e. into a part that does not depend on θ and a part that only depends on y through $t(y)$.

Example 8.6. Assume $y_1, \ldots, y_n$ are independent draws from $N(\theta, 1)$. Then

$$\begin{aligned} f_\theta(y) &= C \exp(-\sum_{i=1}^{n}(y_i - \theta)^2/2) \\ &= C \exp(-\sum_{i=1}^{n} y_i^2/2) \exp(-n\theta^2/2 + \theta \sum_{i=1}^{n} y_i). \end{aligned}$$

Hence $t(y) = \sum_{i=1}^{n} y_i$ is a sufficient statistic for θ. This means that there is no loss of information if we forget the individual values and just keep their sum.

A conditional test is performed conditionally on the observed value of $s(y)$. Hence, the P-value (8.3) is replaced by the *conditional* P-value defined by

$$p(y) = P(t(Y) \geq t(y)|s(Y) = s(y), H_0). \tag{8.6}$$

Example 8.7 (A call for help?)**.** Assume that person D has agreed with person E that in the need of assistance, he should continuously transmit a binary message ...0101010101010101... (i.e. alternating zeros and ones). When D is not transmitting, E recieves background noise $\ldots y_i y_{i+1} y_{i+2} \ldots$ where the y_i are independent and equal to zero with probability p and one with probability $1-p$. One day, E recieved the sequence 000101100101010100010101. Was this just background noise or a call for help where some signals got distorted?

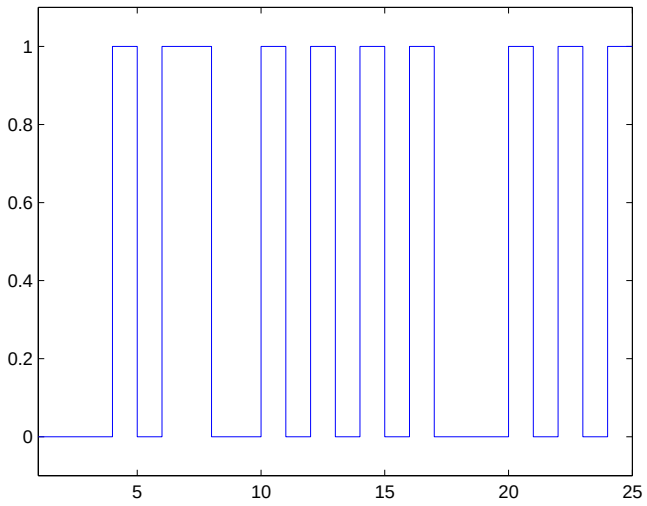

Figure 8.2: Recorded message in Example 8.7.

Here we want to test the hypothesis H_0 :"message is background noise" and a possible test-statistic is

$$t(y) = max(|y - I_1|, |y - I_2|)$$

where I_1 and I_2 are alternating sequences of zeros and ones starting with a zero and a one respectively. If we knew p, this would be a straightforward Monte-Carlo test. However, $s(y) = \sum y_i$, the total number of ones, is a sufficient statistic for p (check this). Conditionally on the observed $s(y) = 10$, $Y|s(y) = 10$ is just a random arrangement of 10 ones and 14 zeros in a vector. Hence we proceed as follows in Matlab

```
y=[0 0 0 1 0 1 1 0 0 1 0 1 0 1 0 1 0 0 0 1 0 1 0 1];
I1=repmat([0 1],1,12);
I2=repmat([1 0],1,12);
t=max(sum(abs(y-I1)),sum(abs(y-I2)));
for i=1:10000
  Y=y(randperm(24));
  T(i)=max(sum(abs(Y-I1)),sum(abs(Y-I2)));
```

```
end
phat=mean(T>=t)

phat =

    0.0018
```

Looks like a call for help.

Example 8.8 (Pines)**.** Figure 8.3 describes the location of pines in a square forest, data is contained in the file `pines.mat`. It is of interest to determine if they are randomly scattered or tend to cluster in groups. Hence, we might want to test H_0:"locations form a Poisson process with rate λ". Simulating from H_0 involves first generating random variable N with a Poisson(λA) distribution (where A is the area of the forest) and then N points uniformly distributed over the forest. Unfortunately we can't simulate such an N since we do not know λ. Hence, we condition on the value of N we actually observed: $N = 118$.

We let

$$H_0 : \textit{"the locations are randomly scattered"} \tag{8.7}$$

and as a test statistic we choose

$$t = \sum_{i=1}^{n} \tau_i, \tag{8.8}$$

where τ_i is the distance between pine i and its nearest neighbour. We would expect t to be small if pines tend to be clustered and large if they tend to avoid eachother. Since there are 118 pines in the data-set, simulating from H_0 simply amounts to simulating 118 points uniformly on $[0, 100] \times [0, 100]$. t is calculated by the following matlab function

```
function t=tn2(x);
n=length(x);
for i=1:n
    d=sqrt((x(i,1)-x(:,1)).^2+(x(i,2)-x(:,2)).^2);
    tau(i)=min(d(d>0));
end
t=sum(tau);
```

and analysis by simulating 1000 (WARNING: 1000 values might take some time on a slow computer, use less) values from $t(Y)|H_0, N = 118$ is performed by

```
for i=1:1000
  T(i)=tn2(rand(118,2)*100);
end
```

A histogram of the simulated test statistics is given in Figure 8.4. Evaluating (8.8) for the observed data gave the value 536, which gives no confidence in rejecting the hypothesis that pines are randomly scattered. Note again that (8.8) was rather arbitrarily chosen as a test statistic and perhaps with another choice we would have produced a stronger test that could reject H_0.

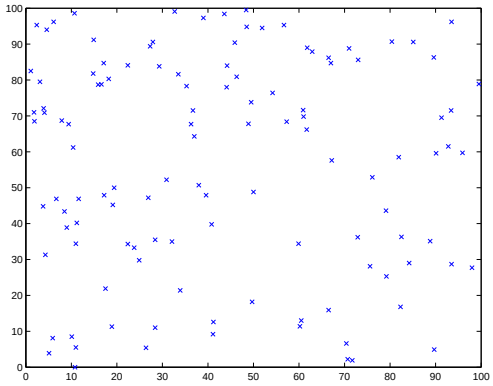

Figure 8.3: Locations of pines.

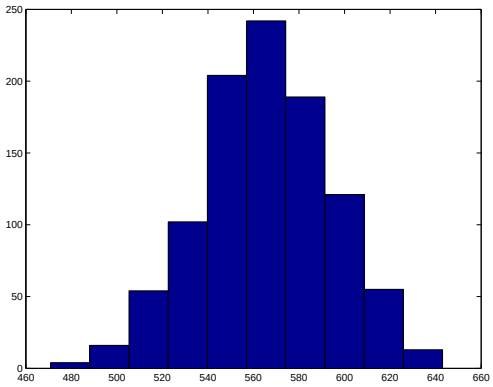

Figure 8.4: Histogram of simulated test statistic t^*.

Permutation tests

We have already mentioned that the full sample, $s(y) = (y_1, \ldots, y_n)$, is a sufficient statistic for any parameter θ. It is not very useful in testing though, since the distribution of $t(Y)|Y = y$ is a point mass at $t(y)$. However, the related statistic $s(y) = (y_{(1)}, \ldots, y_{(n)})$, the ordered sample is often useful.

Definition 8.2 (Exchangable random variables)**.** A vector $Y = (Y_1, \ldots, Y_n)$ of random variables is said to be *exchangable* if $(Y_{i_1}, \ldots, Y_{i_n})$ has the same distribution for all permutations $i = (i_1, \ldots, i_n)$ of the index set $(1, \ldots, n)$.

Obviously independent and identically distributed random variables are exchangable, but the exchangability concept is slightly more general. For example, if $(Z_1, \ldots, Z_n)$ are i.i.d. and X a random variable independent of the Z_i, then $(Y_1, \ldots, Y_n) = (Z_1 + X, \ldots, Z_n + X)$ are exchangable but not independent. Examples of non-exchangable vectors are cases where (some of) the Y_i have different distributions and vectors with serial dependence like time-series models.

What is important here is that if $Y = (Y_1, \ldots, Y_n)$ with distribution P_θ are exchangeble, then $s(Y) = (Y_{(1)}, \ldots, Y_{(n)})$ is a sufficient statistic for θ. This is clear since with f_θ, the density corresponding to P_θ, by exchangability

$$f_\theta(y_1, \ldots, y_n) = f_\theta(y_{(1)}, \ldots, y_{(n)}) = k(s(y), \theta).$$

Moreover, the distribution of $(Y_1, \ldots, Y_n)|(Y_{(1)}, \ldots, Y_{(n)}) = (y_{(1)}, \ldots, y_{(n)})$ is the uniform distribution over the set of all permutations of the vector $(y_1, \ldots, y_n)$. Based on these reults, a *permutation test* proceeds as follows

Algorithm 8.2 (Monte-Carlo permutation test)**.**

1. Draw N random permutations $y^{(1)}, \ldots, y^{(N)}$, of the vector y.
2. Compute $t^{(i)} = t(y^{(i)})$ for $i = 1, \ldots, N$.
3. Compute $\hat{p}(y) = N^{-1} \sum_{i=1}^{N} \mathbf{1}\{t^{(i)} \geq t(y)\}$.
4. Reject H_0 if $\hat{p} \leq \alpha$.

Permutation tests can be very efficient for testing an exchangable null-hypothesis against a non-exchangable alternative.

Example 8.9 (Ph-levels cont.)**.** Returning to the Ph-levels again, we assume this time that historical data are independent observations from some unknown distribution $F(x)$ and current data from $G(y) = F(y - \theta)$. The null-hypothesis is $H_0 : \theta = 0$. Under this hypothesis the full vector of current and historical measurements $y = (y_1, \ldots, y_{273})$ is an observation of a vector of i.i.d. and hence also exchangable random variables.

A natural test-statistic is the difference in means

$$t = t(y) = \frac{1}{149}\sum_{i=125}^{273} y_i - \frac{1}{124}\sum_{i=1}^{124} y_i,$$

and we want to reject H_0 when this is "large". In Matlab

```
y=[ph1;ph2];
t=mean(ph2)-mean(ph1);
for i=1:1000
  Y=y(randperm(273));
  T(i)=mean(Y(125:273))-mean(Y(1:124));
end
p=mean(T>=t)
ans = 0.0240
```

Since the P-value is less than 0.05 we can reject the null-hypothesis at this level.

Example 8.10 (Law school data cont.)**.** Returning to the law schools scores from Example 7.5, of interest is again to test for independence between the two tests and hence we define H_0:$F_0(x, y) = H(x)G(y)$, where data are assumed to be drawn from F_0.

This time, a sufficient statistic under the null-hypothesis is the two ordered samples, $s(x, y) = ((y_{(1)}, \ldots, y_{(15)}), (x_{(1)}, \ldots, x_{(15)})$. And drawing from $(X, Y)|s(X, Y) = s(x, y)$ proceeds by combining the two randomly permuted vectors in pairs. As test-statistic we choose the absolute value of the empirical correlation of the pairs.

```
t=abs(corr(law));
for i=1:1000
  LAW=law([randperm(15)',randperm(15)']);
  T(i)=abs(corr(LAW));
end
p=mean(tn>=t)

p = 0.0010
```

based on this value, we can reject the null-hypothesis of independence with large confidence.

8.2.3 Bootstrap tests

Tests based on confidence intervals

With the knowledge of how to produce bootstrap confidence intervals, we immediately have the tools for testing hypothesis of e.g. the form $H_0 : \theta = \theta_0$ or $H_0 : \theta > \theta_0$ for fixed values θ_0, using the relation between construction of confidence intervals and hypothesis testing. For example using the analysis in Example 7.5 we can immediately reject the hypothesis that the two law-scores are uncorrelated, since the bootstrap confidence interval did not cover

0. Similarily, a quick look at the histogram in Figure 7.7 suggests that 0 is not contained in a percentile confidence interval of the median differences, and we can reject the hypothesis that there has been no increase in Ph-level with some confidence.

The above procedure generalises traditional tests based on the normal distribution to cases where there are evidence of non-normality.

Sampling from an approximate null-distribution

As before, the Bootstrap idea is to estimate P_0 by $\hat{P}$ and then proceed as if the latter was the correct distribution. Hence, we want to find a Monte-Carlo approximation of the P-value

$$p(y) = P(t(Y^*) > t(y)|H_0), \text{ where } Y^* \sim \hat{P}. \tag{8.9}$$

In contrast to the previous sections, this is only a statistical approximation of a P-value but if H_0 is true and $\hat{P}$ is a good estimate it should provide sensible results. An important difference from the Bootstrap algorithms discussed earlier is that here we need to insist that $\hat{P}$ satisfies H_0.

Example 8.11 (Testing for a zero mean)**.** Assume we have observations $y = (y_1, \ldots, y_n)$ independent from F_0 and we want to test if they have a zero mean, $H_0 : E(Y_i) = 0$. Here F_0 remains unspecified by the null-hypothesis and can be estimated by the empirical distribution function $\hat{F}$. However, if $Z^* \sim \hat{F}$, then $E(Z^*) = \bar{y}$ which is not necessarily zero. Hence we choose to approximate P_0 by the distribution function $F_n(u) = \prod_{i=1}^n \hat{F}(u_i - \bar{y})$ instead, which has zero-mean marginals. With $t(y) = |\bar{y}|$ as test-statistic, we proceed as follows:

1. Repeat for $j = 1, \ldots, N$

   $j.1$ Draw $i_1, \ldots, i_n$ independently from the uniform distribution on $\{1, \ldots, n\}$.

   $j.2$ Set $y^{*j} = (y_{i_1} - \bar{y}, \ldots, y_{i_n} - \bar{y})$.

   $j.3$ Compute $t^{*j} = t(y^{*j})$.

2. Compute $\hat{p} = N^{-1} \sum_{j=1}^N \mathbf{1}\{t^{*j} \geq t(y)\}$.

3. Reject H_0 if $\hat{p} \leq \alpha$.

The procedure seems similar to that of a permutation test, the main difference being that we draw new observations with replacement in the Bootstrap version and without replacement in the permutation test. However, a permutation test would not be appropriate here since the model under the alternative hypothesis is also exchangable. Moreover, the test-statistic $t(y) = t(|\bar{y}|)$ is invariant under permutations of the vector y.

Example 8.12 (Testing goodness of fit)**.** Consider Example 7.3 with the failure-times for airconditioning equipment. In Figure 7.2 we plotted the

empirical distribution function together with the fitted Exponential distribution. Of course we would not expect the functions to agree fully, but we might worry that the discrepancy is larger than what could be assumed from an Exponential distribution. This could be formulated in terms of the hypothesis H_0 :"Data are independent draws from an Exponential distribution" and as a test-statistic we might choose

$$t(y) = \sum_{i=1}^{n} (i/n - (1 - \exp(-y_{(i)}/\bar{y}))^2, \tag{8.10}$$

which is the sum of the squared differences between the empirical distribution and the fitted Exponential distribution function evaluated at the sampled points. In the Boostrap approximation, we sample from $Y^* \sim$Exp(106.4), the fitted Exponential (which obviously is a member of Θ_0). In Matlab

```
ysort=sort(y);
m=mean(y);
u=(1:12)/12;
t=sum((u-(1-exp(-ysort/m))).^2);
for i=1:1000
  ystar=sort(exprnd(106.4,1,12));
  m=mean(ystar);
  T(i)=sum((u-(1-exp(-ystar/m))).^2);
end
phat=mean(T>=t)
phat =

    0.2110
```

hence, the Exponential seems a safe assumption.

Example 8.13 (Comparing models)**.** In the above example, the alternative hypothesis contained all distributions that are not Exponential. Sometimes you have two candidate models and want to choose the "best" by testing them against eachother. For the air-conditioning example we might test H_0 :"Data are independent draws from an Exponential distribution" against H_A :"Data are independent from a log-normal distribution". The difference in procedure lies mainly in the choice of test-statistic, here a natural choice is the difference of log-likelihood

$$t(y) = \log(L_y^A(\hat{\theta}_A(y))) - \log(L_y^0(\hat{\theta}_0(y))),$$

where L_y^A,$\hat{\theta}_A(y)$ and L_y^0,$\hat{\theta}_0(y)$ are the likelihoods and maximum-likelihood estimates under the null and alternative hypothesises respectively. In Matlab

```
tA=lognfit(y);
t0=mean(y);
t=sum(log((lognpdf(y,tA(1),tA(2)))))-sum(log(exppdf(y,t0)));
for i=1:1000
  ystar=exprnd(106.4,1,12);
```

```
  tA=lognfit(ystar);
  t0=mean(ystar);
  T(i)=sum(log((lognpdf(ystar,tA(1),tA(2))))))...
         -sum(log(exppdf(ystar,t0)));
end
phat=mean(T>=t)

phat =

    0.4600
```

suggesting little evidence for or against either (if the Exponential had a superior fit we would expect a P-value close to 1).

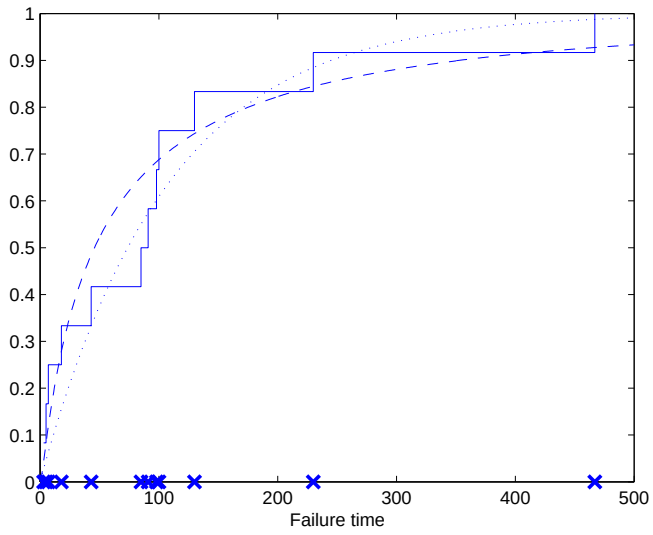

Figure 8.5: Empirical distribution function (solid) fitted Exponential (dotted) and fitted log-Normal (dashed) for airconditioning data (×)

Chapter 9

Missing data models

In this chapter we will be concerned with Maximum-Likelihood estimation of parameters in models involving stochastic variables that are not directly observed. This includes cases where part of data were actually lost for some reason, models where some of the unknowns are better modelled as random variables than fixed parameters (often refered to as latent/hidden variable models) and cases where random variables are introduced into the model for purely computational reasons.

The general setup is that of an *observed data model*: $y \in \mathcal{Y}$ is an observation of $Y \sim P_{\theta_0}$, $P_{\theta_0} \in \mathcal{P} = \{P_\theta; \theta \in \Theta\}$, just as in Chapter 6. The difference is that in this chapter we will consider cases where the likelihood $L_y(\theta) = f_\theta(y)$ is not explicitly known or difficult to maximise.

Example 9.1 (Truncated data)**.** Assume you have observed $y \in \mathbf{R}^n$, that are independent observations from $Y_i = \lceil Z_i \rceil$, where Z_i has density f_{θ_0} for $i = 1, \ldots, n$. Here $Z = (Z_1, \ldots, Z_n)$ can be viewed as "missing" data. The likelihood in this example is

$$L_y(\theta) = \prod_{i=1}^{n} \int_{y_i - 1}^{y_i} f_\theta(z_i)\, dz_i, \tag{9.1}$$

which can be difficult to compute or maximize. The idea here and in the following is that deriving the likelihood would have been much easier *had we observed the missing data* z.

The *augmented/complete data* is $(y, Z) \in \mathcal{Y} \times \mathcal{Z}$, where marginally y is an observation of $Y \sim P_{\theta_0}$ and $(Y, Z) \sim P_{\theta_0}^{\text{aug}}$. Correspondingly, we may define the augmented/complete data model as $P_{\theta_0}^{\text{aug}} \in \{P_\theta^{\text{aug}}; \theta \in \Theta\}$ and the augmented likelihood

$$L_{y,Z}(\theta) = f_\theta(y, Z), \tag{9.2}$$

where $f_\theta(y, Z)$ is the density corresponding to P_θ^{aug} evaluated at observed data y and the random variable Z. Note that $L_{y,Z}(\theta)$ is a random variable.

Our quantity of interest is now the Maximum-Likelihood estimator based on observed data, i.e.

$$\hat{\theta} = \text{argmax}_\theta L_y(\theta) = \text{argmax}_\theta \int f_\theta(y, z)\, dz. \tag{9.3}$$

We will in the next chapter discuss the *EM-algorithm*, a numerical scheme for finding $\hat{\theta}$ defined in (9.3) while avoiding explicit evaluation of the integral on the right-hand side.

In Example 9.1, the augmented data had a clear interpretation as being "lost" in the observation process. An example where hidden random variables are part of the modelling is hidden Markov models:

Example 9.2 (Noisy binary data)**.** Assume a signal $Z_1, \ldots, Z_n$ consisting of zeros and ones evolves according to a Markov chain with transition matrix $M = (m_{ij})$, $(i, j) \in \{0, 1\} \times \{0, 1\}$ i.e. $P(Z_k = j | Z_{k-1} = i) = m_{ij}$, but we can only observe a noisy version $y_1, \ldots, y_n$ where $Y_k = Z_k + \epsilon_k$ and the ϵ_k are independent $N(0, \sigma^2)$. This is an example of a hidden Markov model (HMM), a sample data set is shown in Figure 9.1.

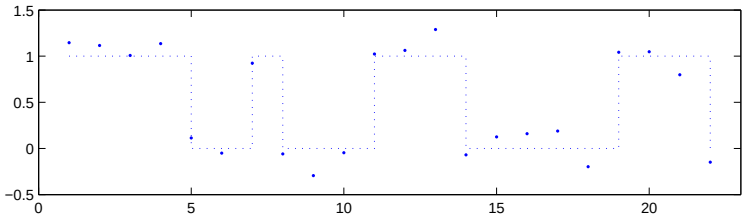

Figure 9.1: Observations (dots) from a hidden Markov model, the dotted line is the (unobserved) signal.

A similar example, but where the hidden variables are artificially introduced, is given by mixture models:

Example 9.3. For a certain species of insects it is difficult to distinguish sex of the animals. However, based on previous experiments, the weight of males are known to be distributed according to a density f_0 and of females according to f_1. The weight of a 1000 randomly chosen insects in a large population is recorded, and of interest is the proportion θ of males in the population. Here, the sample can be viewed as independent draws from the mixture density

$$\theta f_0(y) + (1 - \theta) f_1(y). \tag{9.4}$$

The likelihood is

$$L_y(\theta) = \prod_{i=1}^{n} (\theta f_0(y_i) + (1 - \theta) f_1(y_i)), \tag{9.5}$$

and requires numerical maximation. However, we may introduce latent indicator variables Z_i, where $Z_i = 0$ if y_i is measured on a male and $Z_i = 1$ if it is a female and hence $P(Z_i = 0) = \theta$, we get the augmented likelihood

$$L_{y,z}(\theta) = \prod_{i=1}^{n} f_{z_i}(y_i)\theta^{1-z_i}(1-\theta)^{z_i}, \tag{9.6}$$

which is often easier to handle numerically.

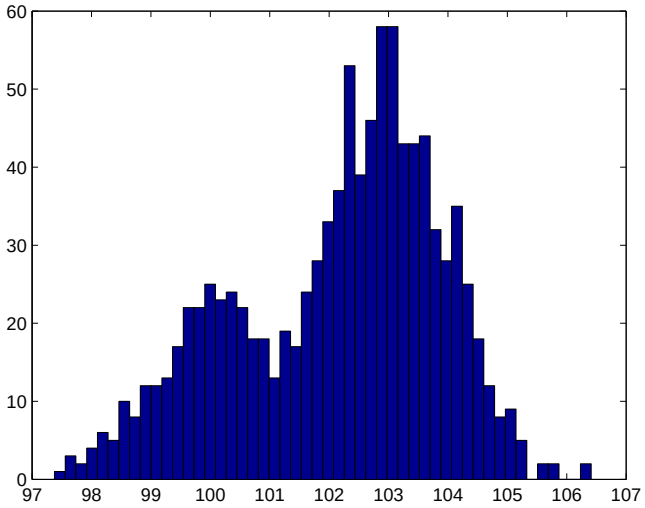

Figure 9.2: Histogram of data from the mixture example.

The EM-algorithm is an iterative algorithm for approximating $\hat{\theta}$ in (9.2). Given a current approximation $\theta^{(i)}$, we define the function

$$\theta \mapsto Q(\theta, \theta^{(i)}) = \int \log(L_{y,z}(\theta)) f_{\theta^{(i)}}(z|y)\, dz = E(\log(L_{y,Z^{(i)}}(\theta))), \tag{9.7}$$

where $Z^{(i)}$ is a random variable with density $f_{\theta^{(i)}}(z|y)$, the density of missing data conditionally on observed data and that the unknown parameter equals $\theta^{(i)}$.

Algorithm 9.1 (The EM-algorithm)**.**

1. Choose a starting value $\theta^{(0)}$.
2. Repeat for $i = 1, 2, \ldots$ until convergence.
   i Compute $\theta^{(i)} = \text{argmax}_\theta Q(\theta, \theta^{(i-1)})$.

Note that if

$$L_{y,z}(\theta) = h(\theta, y, z)g(y, z)$$

we can replace $Q(\theta, \theta^{(i)})$ by

$$\tilde{Q}(\theta, \theta^{(i)}) = E(\log(h(\theta, y, Z^{(i)})))$$

in the algorithm without changing the location of the maxima.

The name of the algorithm comes from the fact that each iteration involves taking an *expectation* when computing $Q(\theta, \theta^{(i-1)})$ (the E-step) and performing a *maximisation* (the M-step) when maximising the same. For the algorithm to be of any use, it is of course essential that these two steps are easier to perform than a direct evaluation of 9.3.

Theorem 9.1. *If $\theta^{(i)} = argmax_\theta Q(\theta, \theta^{(i-1)})$, then $L_y(\theta^{(i)}) \geq L_y(\theta^{(i-1)})$ with equality holding if and only if $Q(\theta^{(i+1)}, \theta^{(i)}) = Q(\theta^{(i)}, \theta^{(i)})$. I.e. the (observed data) likelihood is increased in each step of the EM-algorithm.*

If in addition $Q(\theta, \theta')$ is continuous in both variables, the algorithm will converge to a *stationary point* $\tilde{\theta}$, i.e. such that $\tilde{\theta} = \text{argmax}_\theta Q(\theta, \tilde{\theta})$. Under further technical conditions, a stationary point will correspond to a *local* maxima of $L_y(\theta)$, but there is no guarantee that this will be the global maxima.

Example 9.4 (Truncated data cont.)**.** Assume we have n independent observations of $Y = \lceil Z \rceil$, where we assume $Z \sim \text{Exp}(\theta_0)$. To apply the EM-algorithm, we first need to define our augmented data-set which here is naturally given by $(y_1, Z_1), \ldots, (y_n, Z_n)$, where Z_i is the value of y_i *before* truncation. Next we need to derive the augmented likelihood $L_{y,z}(\theta)$ and the distribution of $Z|y$: First note that the conditional distribution of $Y_i|Z_i = z_i$ is just a point-mass at $\lceil z_i \rceil$. Hence

$$L_{y,z}(\theta) = f_\theta(y, z) = f_\theta(y|z)f_\theta(z) = \prod_{i=1}^{n} \mathbf{1}\{y_i = \lceil z_i \rceil\} \exp(-z_i/\theta)/\theta, \quad (9.8)$$

where of course the factor $f_\theta(y|z)$ does not depend on θ and can be dropped. Further,

$$z_i \mapsto f_\theta(z_i|y_i) \propto f_\theta(z_i, y_i) = \mathbf{1}\{y_i = \lceil z_i \rceil\} \exp(-z_i/\theta)/\theta \quad (9.9)$$

an $\text{Exp}(\theta)$ random variable truncated to $[y_i, y_i + 1]$. Hence,

$$Q(\theta, \theta^{(i)}) = \sum_{j=1}^{n} (E(-Z_j^{(i)})/\theta - \log(\theta))$$

which is maximized by $\theta^{(i+1)} = n^{-1} \sum_{j=1}^{n} E(-Z_j^{(i)})$. Finally

$$\begin{aligned} E(Z_j^{(i)}) &= \frac{\int_{y_i}^{y_i+1} z \exp(-z/\theta^{(i)})/\theta^{(i)}\, dz}{\int_{y_i}^{y_i+1} \exp(-z/\theta^{(i)})/\theta^{(i)}\, dz} \\ &= \frac{\exp(1/\theta^{(i)})(y_j + \theta^{(i)}) - y_j - 1 - \theta^{(i)}}{\exp(1/\theta^{(i)}) - 1}, \end{aligned}$$

which specifies the recursion. In Matlab, with a simulated data-set:

```
y=ceil(exprnd(3,1,100));
t(1)=1;
for i=1:20
t(i+1)=mean((exp(1/t(i))*(y+t(i))-y-1-t(i)))/(exp(1/t(i))-1);
end
```

In Figure 9.3 we have plotted the 20 iterations of the algorithm and convergence seems almost immediate to $\hat{\theta} = 3.7\ldots$. It is perhaps questionable as to whether using the EM-algorithm here was simpler than direct maximisation and the example should be viewed as an illustration only.

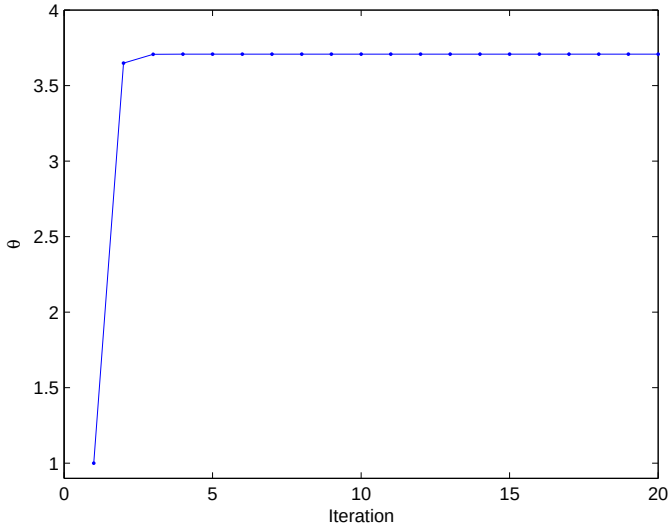

Figure 9.3: Iterations $\theta^{(i)}$ for the truncated data example.

Example 9.5. Assume a radioactive material is known to emit Po(100) particles in unit time. A measurement equipment records each particle with probability θ_0. If the recorded value was $y = 84$, what is the maximum-likelihood estimate of θ_0?

Here we failed to observe Z, the total number of particles emitted. Conditionally on Z, y is an observation of a $\mathrm{Bin}(\theta_0, Z)$ variable. The augmented likelihood is

$$\begin{aligned} L_{y,z}(\theta) = f_\theta(y,z) &= \binom{z}{y} \theta^y (1-\theta)^{z-y} \frac{100^z \exp(-100)}{z!} \\ &\propto \theta^y (1-\theta)^{z-y} \end{aligned}$$

and

$$z \mapsto f_\theta(z|y) \propto \frac{z!}{(z-y)!}(1-\theta)^{z-y}\frac{100^z \exp(-100)}{z!}$$
$$\propto \frac{(1-\theta)^{z-y}100^z}{(z-y)!} \propto \frac{(100(1-\theta))^{z-y}}{(z-y)!},$$

which we recognise as the probability mass function of $y + W$, where $W \sim \text{Po}(100(1-\theta))$. Hence,

$$\tilde{Q}(\theta, \theta^{(i)}) = y\log(\theta) + \log(1-\theta)E(Z^{(i)} - y)$$
$$= y\log(\theta) + 100\log(1-\theta)(1-\theta^{(i)}),$$

which is maximized by

$$\theta^{(i+1)} = y/(y + 100(1-\theta^{(i)})).$$

In Matlab

```
y=84;
t(1)=0.5;
for i=1:50
  t(i+1)=y/(y+100*(1-t(i)));
end
```

which converges to 0.84 as expected.

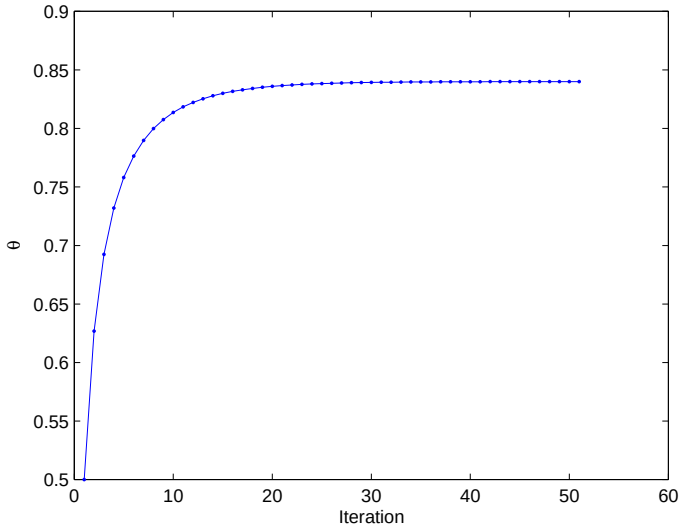

Figure 9.4: Iterations $\theta^{(i)}$ for the Poisson-Binomial example.

150	170	190	220
8064*	1764	408	408
8064*	2772	408	408
8064*	3444	1344	504
8064*	3542	1344	504
8064*	3780	1440	504
8064*	4860	1680*	528*
8064*	5196	1680*	528*
8064*	5448*	1680*	528*
8064*	5448*	1680*	528*
8064*	5448*	1680*	528*

Table 9.1: Censored regression data

Example 9.6 (Censored regression)**.** Here we consider a regression problem involving censored data. Consider the data in Table 9.1. These data represent failure times (hours) of motorettes at four different temperatures (Celsius). The asterisks denote a censored observation, so that for example, 8064* means the item lasted *at least* 8064 hours.

Physical considerations suggest the following model relating the logarithm (base 10) of lifetime, $y_i = \log_{10}(t_i)$ and $x_i = 1000/(\text{temperature} + 273.2)$:

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i \tag{9.10}$$

where $\epsilon_i \sim N(0, \sigma^2)$ and the x_i are fixed. On this scale a plot of y_i against x_i is given in Figure 9.5 in which censored data are plotted as open circles. The transformed data are stored in `motorette.mat`, where the first 17 values are uncensored and the rest censored.

Now, in this situation, if the data were uncensored we would have a simple regression problem. Because of the censoring we use the EM algorithm to first estimate where the censored values actually are (the E–step) and then fit the model to the augmented data set (the M–step). Re–ordering the data so that the first m values are uncensored, and denoting by Z_i the augmented data values, we have the augmented log-likelihood:

$$\log(L_{y,Z}(\theta)) = -n\log\sigma - (\sum_{i=1}^{m}(y_i - \beta_0 - \beta_1 x_i)^2 + \sum_{i=m+1}^{n}(Z_i - \beta_0 - \beta_1 x_i)^2)/(2\sigma^2). \tag{9.11}$$

The E–step requires us to find the expectations of Z_i given the censoring time c_i. Unconditionally on the censoring, the distribution of each Z_i is normal, so the conditioning is a normal probability, conditioned on $Z_i > c_i$. It is fairly straightforward to show that with $\theta = (\beta_0, \beta_1, \sigma^2)$,

$$M_i(\theta) = E(Z_i | Z_i > c_i) = \mu_i + \sigma H((c_i - \mu_i)/\sigma)$$

and

$$V_i(\theta) = E(Z_i^2 | Z_i > c_i) = \mu_i^2 + \sigma^2 + \sigma(c_i + \mu_i)H((c_i - \mu_i)/\sigma),$$

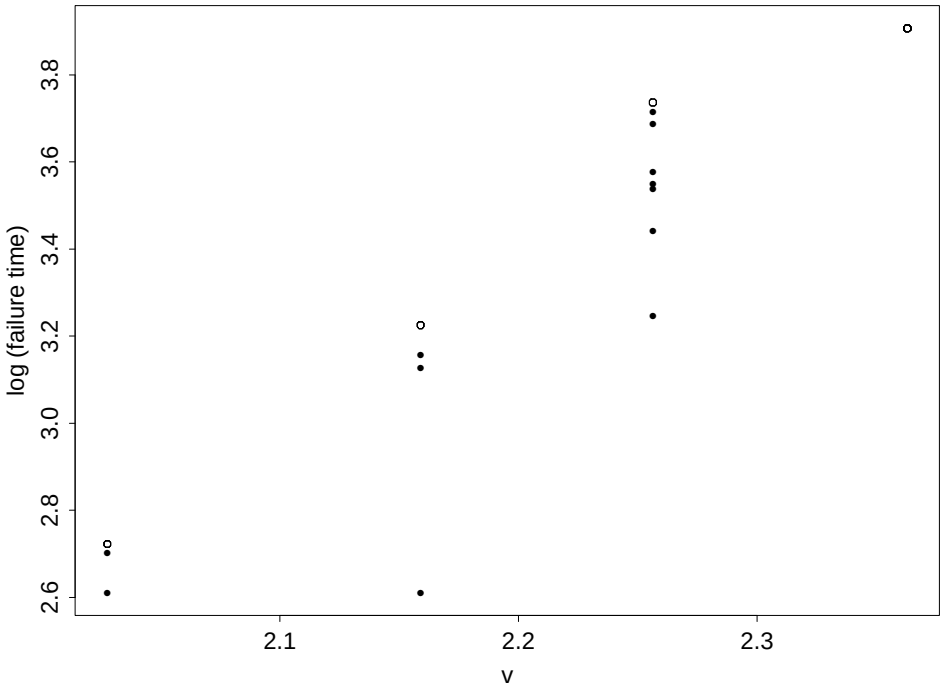

Figure 9.5: Censored regression data

where $\mu_i = \beta_0 + \beta_1 x_i$ and $H(x) = \phi(x)/(1-\Phi(x))$. Thus

$$\begin{aligned} Q(\theta, \theta^{(i)}) &= -n \log \sigma - \sum_{j=1}^{m} (y_j - \beta_0 - \beta_1 x_j)^2/(2\sigma^2) \\ &\quad - \sum_{j=m+1}^{n} E(Z_j^{(i)} - \beta_0 - \beta_1 x_j)^2/(2\sigma^2) \\ &= -n \log \sigma - \sum_{j=1}^{m} (y_j - \beta_0 - \beta_1 x_j)^2/(2\sigma^2) \\ &\quad - \sum_{j=m+1}^{n} (V_j(\theta^{(i)}) - 2(\beta_0 + \beta_1 x_i) M_i(\theta^{(i)}) + (\beta_0 + \beta_1 x_j)^2)/(2\sigma^2). \end{aligned}$$

Maximising this with respect to (β_0, β_1) proceeds as for an ordinary regression problem with data

$$(y_1, x_1), \ldots, (y_m, x_m), (M_{m+1}(\theta^{(i)}), x_{m+1}), \ldots, (M_n(\theta^{(i)}), x_n),$$

while the solution for σ^2 equals

$$\begin{aligned} \sigma^{(i+1)2} &= \frac{1}{n} \sum_{j=1}^{m} (y_j - \mu_j^{(i+1)})^2 \\ &\quad + \frac{\sigma^{(i)2}}{n} \sum_{j=m+1}^{n} (1 + ((c_j - \mu_j^{(i)})/\sigma^{(i)}) H((c_j - \mu_j^{(i)})/\sigma^{(i)}). \end{aligned}$$

In Matlab, with starting values fitted through the uncensored observations,

```
b=polyfit(x(1:17),y(1:17),1);
mu=b(1,2)+b(1,1).*x;
s=std(y(1:17)-mu(1:17));
d=(y(18:40)-mu(18:40))/s(1);
for i=1:50
    M(1:23)=mu(18:40)+s(i)*normpdf(d)./(1-normcdf(d));
    b(i+1,1:2)=polyfit(x,[y(1:17) M],1);
    mu=b(i+1,2)+b(i+1,1).*x;
    d=(y(18:40)-mu(18:40))/s(i);
    s1=(y(1:17)-b(i+1,2)-b(i+1,1)*x(1:17)).^2;
    s2=s(i)^2*(1+d.*normpdf(d)./(1-normcdf(d)));
    s(i+1)=sqrt(mean([s1 s2]));
end
```

The sequence of linear fits is shown in Figure 9.7. Note that the initial fit (bottom line) was made without taking censored data into account, so the difference between the first and final fits gives an indication of the necessity of including the censored observations.

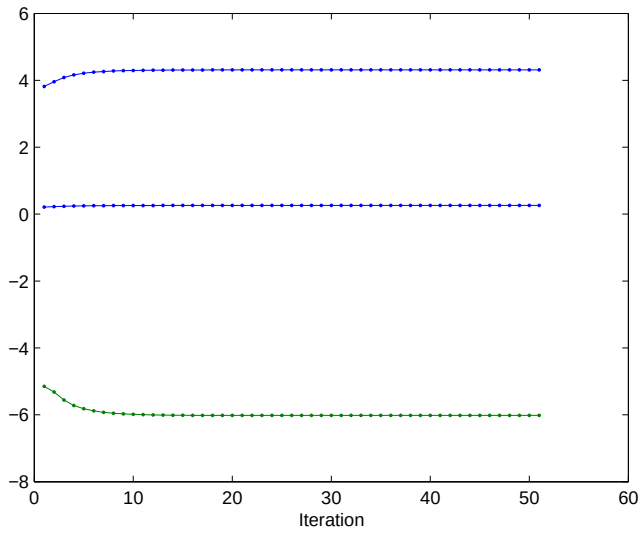

Figure 9.6: Convergence of parameter estimates for censored regression example, from top to bottom β_1, σ, β_0.

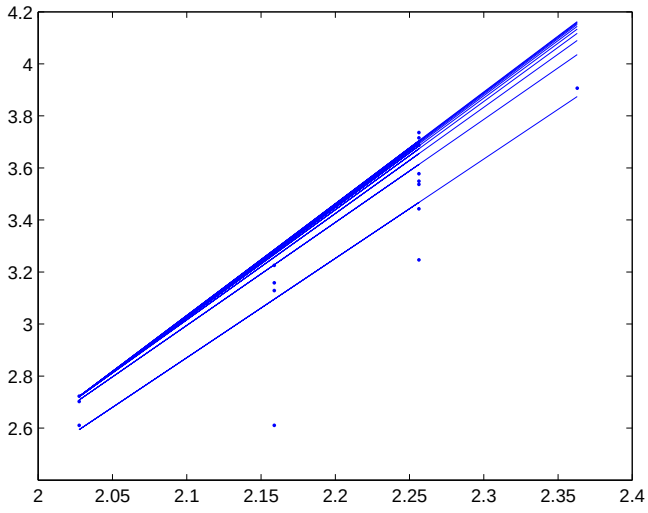

Figure 9.7: EM iterates to regression fit of censored data

Part III

Applications to Bayesian inference

Chapter 10

Bayesian statistics

During the past 30 years, interest in the Bayesian approach to statistics has increased dramatically, much due to the popularisation of MCMC methods which has made previously intractable calculations straightforward with a fast computer. The fundamental difference between the Bayesian and classical approach is that instead of viewing parameters as unknown and fixed quantities, the Bayesian view them as unobserved random variables. Hence, to a Bayesian statistician, a statement like $P(\theta > 0)$ makes sense and will give an answer in $[0, 1]$ while a classical/frequentist would argue that the probability is either zero or one (and we are not likely to deduce the answer) since there are no random variables involved. The view that the parameter is a random variable should not be interpreted as "there is no fixed true value θ_0" but rather as a way to model the uncertainty involved in this unknown value.

10.1 Bayesian statistical models

Remember that a classical statistical model was defined by the set $\mathcal{P}$ of possible distributions P_θ of data. A Bayesian statistical model is defined by the *joint* distribution of data and parameters, i.e. the density $f(y, \theta) = f(y|\theta)f(\theta)$ of (Y, Θ) (here and further on Θ is a random variable and not a parameter space).

Roughly, a Bayesian analysis proceeds through the following four steps:

1. Specification of a likelihood $f(y|\theta)$, the distribution of $Y|\Theta = \theta$, data conditionally on the parameters.
2. Determination of a *prior distribution* $f(\theta)$ of the unknown parameters Θ.
3. Observing data y and computing the *posterior distribution* $f(\theta|y)$, the distribution of $\Theta|Y = y$, parameters conditionally on observed data.
4. Drawing inferences from this posterior distribution.

Essentially new here (in comparison with the classical approach) is the concept of a *prior distribution* and that inference is based on the *posterior*

distribution $f(\theta|y)$ rather than the likelihood $f(y|\theta)$. Another difference is that the classical assumption of independent and identically distributed observation transfers to the assumption of exchangable observations. In a Bayesian model, the distribution of the data vector $Y = (Y_1, \ldots, Y_n)$ depends on the value of the random variable Θ, hence the components of the data-vector will in general not be independent (though they are often assumed independent *conditionally* on Θ).

10.1.1 The prior distribution

The prior distribution $f(\theta)$ summarises the *apriori* knowledge of Θ i.e. *the knowledge of* Θ *obtained before observing data.* For example, if you "know" that $2 \leq \Theta \leq 5$ it may be appropriate to let $f(\theta)$ be an uniform distribution on this interval. Prior considerations tend to vary from person to person, and this lead to subjective choices of prior distribution. The main objection to Bayesian inference is that such subjective choices of prior will necessarily affect the final conclusion of the analysis. Historically, Bayesians have argued that prior information is always available and that the subjective view is advantageous. Nowadays, much research in Bayesian statistics is instead focused on finding formal rules for "objective" prior elicitation.

Classical statisticians on the other hand tend to argue; Bayesians are subjective, I am not a Bayesian, hence my analysis is objective. Though classical choices of estimators and test-statistics are of course subjective in a similar manner and both approaches make the subjective choice of a likelihood function. We will leave this discussion to the philosophers for now and just mention that as more data is collected, the prior distribution will have less influence on the conclusion.

10.1.2 The posterior distribution

The posterior distribution $f(\theta|y)$ summarises the information about Θ provided by data and prior considerations. It is given by *Bayes' Theorem*, which is essentially a trivial consequence of the definition of conditional probabilities:

$$\theta \mapsto f(\theta|y) = \frac{f(y|\theta)f(\theta)}{f(y)} \propto f(y|\theta)f(\theta), \tag{10.1}$$

i.e. the posterior density is proportional to the product of the likelihood and the prior. Based on this density, any question of interest regarding Θ can be derived through basic manipulations derived from probability theory only. No asymptotic or "statistical" approximations are necessary. However, for complex models, *numerical* approximations are often necessary and this is where MCMC comes into the picture. By constructing a Markov-Chain $\Theta^{(i)}$, $i = 1, \ldots$, with $f(\theta|y)$ as stationary distribution, we can answer questions about $f(\theta|y)$ by Monte-Carlo integration.

Example 10.1. If $Y|\Theta = \theta \sim \text{Bin}(n, \theta)$ and, apriori, Θ is standard uniform, then

$$f(\theta|y) \propto \theta^y(1-\theta)^{n-y}, \quad \theta \in [0,1], \tag{10.2}$$

which we recognise as (proportional to) the density of a Beta$(y+1, n-y+1)$ distribution. Of course we do not need to calculate the normalising constant $f(y) = \int_0^1 \theta^y (1-\theta)^{n-y}\, d\theta$ to come to this conclusion. In Chapter 8 we considered a problem where 59 out of 100 coin-flips turned tails up. If Θ is the probability of tails and with the above prior specification, the posterior distribution of Θ is Beta$(60, 42)$. The posterior probability $P(\Theta = 1/2|Y = 59)$ is then equal to 0, since the Beta-distribution is a continuous distribution. But this is simply a consequence of the precise mathematical nature of the statement; we do not really believe there is a coin that lands tails up with probability *exactly* equal to $1/2$ (or a coinflipper that could flip it in an exactly fair manner for that matter). More interesting is the statement $P(\Theta > 1/2|Y = 59) = \int_{1/2}^1 f(\theta|y = 59)\, d\theta = 0.9636\ldots$, which suggests a rather strong posterior evidence that the coin is biased.

In the coin-flipping example it seems reasonable to assume apriori that Θ is close to $1/2$ (it seems hard to construct a coin that always/never ended up tails). Hence, we may choose a Beta(α, α) prior for some $\alpha > 1$ (the standard uniform corresponds to $\alpha = 1$). This gives posterior

$$f(\theta|y) \propto \theta^y (1-\theta)^{n-y} \theta^{\alpha-1} (1-\theta)^{\alpha-1} \tag{10.3}$$

$$= \theta^{y+\alpha-1} (1-\theta)^{n-y+\alpha-1} \quad \theta \in [0,1], \tag{10.4}$$

i.e. a Beta$(y+\alpha, n-y+\alpha)$ posterior distribution. In Figure 10.1 we have plotted the posterior distributions for this example together with the priors for α equal to 1, 2, 10 and 50. Note that in this example we consider α a fixed parameter that index the prior distribution. Such a parameter is often refered to as a *hyperparameter*. Note that the mean of the Beta$(y+\alpha, n-y+\alpha)$ distribution is $(y+\alpha)/(n+2\alpha)$ and its variance will be $O(n^{-1})$, thus for sufficiently large n, the posterior distribution will be concentrated around the Maximum-Likelihood estimator y/n regardless of the choice of prior hyperparameter α. This is generally the case; Bayesian and classical conclusions tend to agree for large samples in the sense that the Maximum-likelihhod estimator $t(Y)$ will have a distribution close to $f(\theta|y)$ for large samples. For small samples you have to choose whether you want to rely on the classical statisticians large-sample approximations or the Bayesian statisticians prior distribution.

Multi-parameter models

Classical statistical approaches often fail to produce reasonable answers when a model contains too many unknown parameters, this is not a problem (in principle) with a Bayesian model. Note that since parameters are treated as unobserved random variables, there is no formal distinction between parameters and missing/latent data. Hence, if you are only interested in one of the model-parameters, i.e. Θ_1, where $\Theta = (\Theta_1, \ldots, \Theta_d)$, its posterior distribution is given by integrating the other parameters out of the problem, i.e.

$$f(\theta_1|y) = \int f(\theta|y)\, d\theta_2 \cdots d\theta_d \propto \int f(y|\theta) f(\theta)\, d\theta_2 \cdots d\theta_d. \tag{10.5}$$

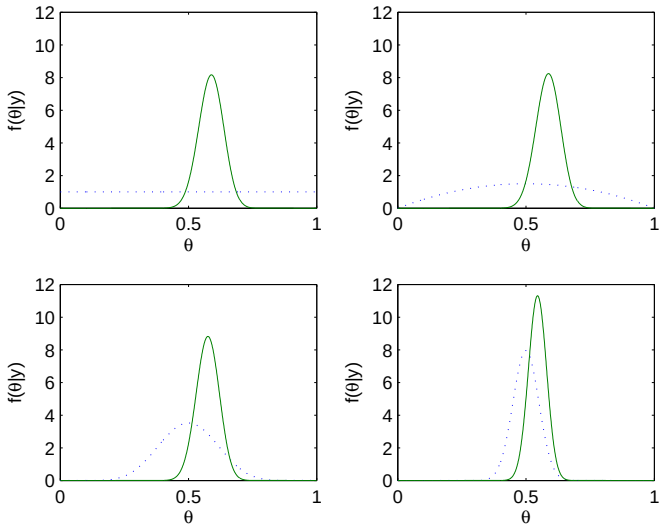

Figure 10.1: Prior (dotted) and posterior (solid) distributions for α equal to 1, 2, 10 and 50

When the above integration is not feasible, a Monte-Carlo approach is to simulate vectors $\theta^{(i)} = (\theta_1^{(i)}, \ldots, \theta_d^{(i)})$ from the posterior $f(\theta|y)$ (which is easy to derive up to proportionality using (10.1)), the first coordinates, $\theta_1^{(i)}$, are now a sample from the marginal $f(\theta_1|y)$ and can be used to approximate related quantities through Monte-Carlo integration.

Prediction

Suppose you have data y, an observation of $Y = (Y_1, \ldots, Y_n)$, and want to say something about a "future observation" Y_{n+1}. For example, you might be interested in the probability that Y_{n+1} lies in some set A given the information contained in y, $P(Y_{n+1}|Y = y)$. A solution based on classical statistics (using the plug-in principle) would be to compute $P(Y_{n+1} \in A|\Theta = \hat{\theta})$ for an estimate $\hat{\theta}$ of θ_0. This can be a reasonable approximation, but it fails to take into account the uncertainty of the estimate $\hat{\theta}$ (we are effectively conditioning on an event that we know is only approximately true). Hence, for example prediction intervals based on classical statistics tends to be too short.

The Bayesian solution however follows directly from probabilistic principles (rather than more ad-hoc statistical principles like the plug-in principle) since given a Bayesian model

$$P(Y_{n+1} \in A|y) = \int_A \int f(y_{n+1}, \theta|y)\, d\theta\, dy_{n+1}, \tag{10.6}$$

where $f(y_{n+1}, \theta|y)$ is the density of $(Y_{n+1}, \Theta)|Y = y$. Hence, a Bayesian *integrates out* instead of *plugging in*. This is a very attractive feature of Bayesian statistics; once a model is defined answers follow directly from probability theory only and "statistical approximations" are unneccesary. The "ad-hockery" lies instead in the model-formulation and especially so in the choice of prior.

Example 10.2. Lets continue Example 10.1 and predict the result Y_2 of 10 independent new flips of the coin. We have observed $y_1 = 59$, of $Y_1|\Theta = \theta \sim \text{Bin}(100, \theta)$ and we choose a uniform prior for Θ. First note that since Y_1 and Y_2 are conditionally independent given $\Theta = \theta$,

$$f(\theta|y_1, y_2) \propto f(y_1, y_2|\theta)f(\theta) = f(y_2|\theta)(f(y_1|\theta)f(\theta)) \propto f(y_2|\theta)f(\theta|y_1).$$

Hence, deriving the posterior $f(\theta|y_1, y_2)$ with prior $f(\theta)$ is equivalent to deriving the posterior $f(y_2|\theta)$ *with* $f(\theta|y_1)$ *as prior.*

Now we were interested in

$$\begin{aligned} f(y_2|y_1) &= \int f(y_2, \theta|y_1)\, d\theta = \int \frac{f(y_1, y_2|\theta)f(\theta)}{f(y_1)}\, d\theta \\ &\propto \int f(y_1|\theta)f(y_2|\theta)f(\theta)\, d\theta \\ &\propto \frac{1}{(10-y_2)!y_2!}\int_0^1 \theta^{y_1+y_2}(1-\theta)^{110-y_1-y_2}\, d\theta \\ &\propto \frac{(y_1+y_2)!(110-y_2-y_1)!}{(10-y_2)!y_2!}, \quad y_2 \in \{0, 1, \ldots, 10\}. \end{aligned}$$

We have plotted this distribution in Figure 10.2.

10.2 Choosing the prior

10.2.1 Conjugate priors

In Example 10.1 we saw that when using a Beta-distribution as a prior for Θ, the posterior distribution was also Beta but with different parameters. This property of the model, that the prior and posterior distributions belong to the same family of distributions is an example of *conjugacy*, and a prior with this property (i.e. a Beta-prior in Example 10.1) is refered to as a *conjugate prior*.

Example 10.3. Assume y is an observation of $Y|\Theta = \theta \sim \text{Po}(\theta)$ and we choose a Gamma(α, β) prior for Θ, where (α, β) are fixed hyperparameters, then

$$\begin{aligned} f(\theta|y) &\propto f(y|\theta)f(\theta) \propto \theta^y \exp(-\theta)\theta^{\alpha-1}\exp(-\theta\beta) \\ &= \theta^{y+\alpha-1}\exp(-\theta(1+\beta)), \end{aligned}$$

i.e. the posterior is Gamma$(y+\alpha, 1+\beta)$ and hence the Gamma-prior is conjugate for a Poisson likelihood just as the Beta was conjugate for a Binomial

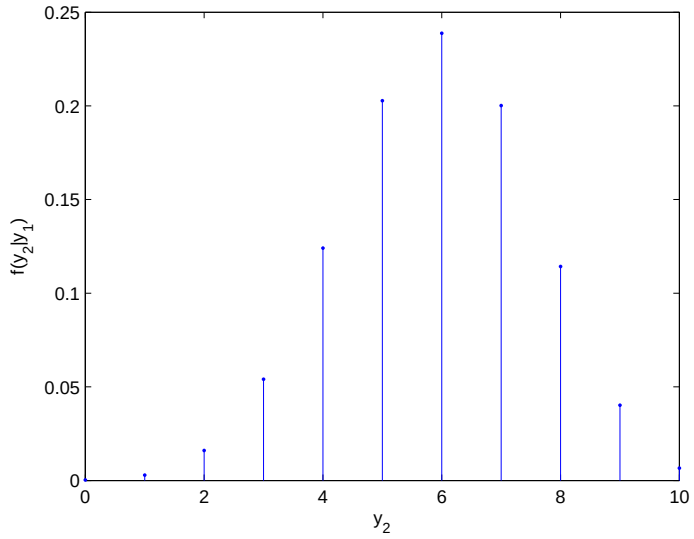

Figure 10.2: Predictive distribution of $Y_2|Y_1 = 59$ in Example 10.2.

likelihood. Note that we use Gamma(α, β) to represent the distribution with density proportional to $y^{\alpha-1}\exp(-y\beta)$ and that this distribution is sometimes denoted Gamma$(\alpha, 1/\beta)$ (The latter is for example used by Matlab!).

Conjugate priors are often applied in practise, but they are difficult to derive in more complex models. Moreover, their motivation is mainly based on mathematical convenience and the concept does not give much guidance on how to choose the hyperparameters. Table 10.2.1 lists conjugate priors and their posteriors for some common models, you may want to check these results.

10.2.2 Improper priors and representing ignorance

Consider the conjugate $N(m, s^2)$ prior for Θ in the $N(\theta, \sigma^2)$ likelihood (Table 10.2.1). The posterior is here

$$N(\frac{m/s^2 + n\bar{y}/\sigma^2}{1/s^2 + n/\sigma^2}, \frac{1}{1/s^2 + n/\sigma^2}) \to N(\bar{y}, \sigma^2/n) \text{ as } s \to \infty.$$

The limiting *prior* distribution (i.e. $N(m, \infty)$) can be interpreted as $f(\theta) \propto 1$, for $\theta \in \mathbf{R}$, but this is not a probability density since it is not integrable. A prior if this type, i.e. that fails to integrate to one, is called an *improper prior*. Improper priors are allowed in Bayesian analysis *as long as the posterior is proper*, i.e. as long as the right-rand side of $f(\theta|y) \propto f(y|\theta)f(\theta)$ is integrable. Choosing a prior that is uniformly distributed over the whole real line is a common strategy in cases where there exist no reliable prior information.

Likelihood	*Prior*	*Posterior*
$\text{Bin}(n,\theta)$	$\text{Beta}(\alpha,\beta)$	$\text{Beta}(\alpha+y,\beta+n-y)$
$\text{Ge}(\theta)$	$\text{Beta}(\alpha,\beta)$	$\text{Beta}(\alpha+n,\beta+\sum_{i=1}^n y_i-n)$
$\text{NegBin}(n,\theta)$	$\text{Beta}(\alpha,\beta)$	$\text{Beta}(\alpha+n,\beta+y-n)$
$\text{Gamma}(k,\theta)$	$\text{Gamma}(\alpha,\beta)$	$\text{Gamma}(\alpha+nk,\beta+\sum_{i=1}^n y_i)$
$\text{Po}(\theta)$	$\text{Gamma}(\alpha,\beta)$	$\text{Gamma}(\alpha+\sum_{i=1}^n y_i,\beta+n)$
$N(\mu,\theta^{-1})$	$\text{Gamma}(\alpha,\beta)$	$\text{Gamma}(\alpha+\frac{n}{2},\beta+\frac{\sum_{i=1}^n (y_i-\mu)^2}{2})$
$N(\theta,\sigma^2)$	$N(m,s^2)$	$N(\frac{m/s^2+n\bar{y}/\sigma^2}{1/s^2+n/\sigma^2},\frac{1}{1/s^2+n/\sigma^2})$

Table 10.1: Conjugate priors for Θ for some common likelihoods. All parameters except θ are assumed fixed and data $(y_1,\ldots,y_n)$ are conditionally independent given $\Theta=\theta$.

Indeed, the uniform distribution does not "favour" any particular value and can be seen as a way to represent ignorance.

Now, if you are ignorant about a parameter Θ you are likely to be equally ignorant about $\Theta'=h(\Theta)$ for some function h. But if Θ has a uniform prior, $f_\Theta(\theta)=1$, then (with $g=h^{-1}$) this corresponds to using a prior $f_{\Theta'}(\theta')=f_\Theta(g(\theta'))|g'(\theta')|=|g'(\theta')|$ on $\Theta'=h(\Theta)$ (c.f. the theorem on transformation of random variables, Theorem 3.1). Hence, ignorance represented by the uniform prior distribution does not translate across scales. For example, if you put a uniform prior on $\mathbf{R}^+$ on the parameter $\Theta=\sigma$ in a Normal likelihood, this corresponds to using a prior proportional to $\sqrt{\theta'}$ for the parameter $\Theta'=\sigma^2$. Uniformity is also not very natural for parameters that do not represent location. For example if Θ is a scale parameter, i.e. the likelihood is of the form $f(y|\theta)=g(y/\theta)/\theta$, then we would like to put more weight on the interval $\theta\in[0,1]$ than the interval $\theta\in[10,11]$, since even if the intervals are of the same length choosing between $\theta=0.1$ and $\theta=0.9$ is going to affect the conclusion dramatically while choosing between $\theta=10.1$ and $\theta=10.9$ is not.

Jeffrey's prior

A prior that does translate across scale is *Jeffrey's prior*. Recall the concept of Fisher information,

$$I_\Theta(\theta)=-E\left(\frac{d^2\log(f(Y|\theta))}{d\theta^2}\right)=E\left(\frac{d\log f(Y|\theta)}{d\theta}\right). \tag{10.7}$$

The Jeffrey's prior is then defined as

$$f(\theta)=|I_\Theta(\theta)|^{1/2}. \tag{10.8}$$

This means that if $\Theta'=h(\Theta)$ for a one-to-one transformation h with inverse g, then

$$f_{\Theta'}(\theta')=|I_\Theta(g(\theta'))|^{1/2}|g'(\theta')|=|I_{\Theta'}(\theta')|^{1/2} \tag{10.9}$$

since

$$I_{\Theta'}(\theta') = -E\left(\frac{d^2\log(f_{Y|\Theta'}(Y|\theta')}{d\theta'^2}\right) = -E\left(\frac{d^2\log(f_{Y|\Theta}(Y|g(\theta'))}{d\theta'^2}\right)$$
$$= I_\Theta(\theta')g'(\theta')^2,$$

which also explains the square root.

Example 10.4. If we have observations $y_1, \ldots, y_n$ from $Y|\Theta = \theta \sim N(\theta, \sigma^2)$, σ^2 known, then

$$I(\theta) = -E\left(\frac{d^2(\sum_{i=1}^n -(Y_i-\theta)^2/(2\sigma^2)}{d\theta^2}\right) = \frac{n}{\sigma^2}, \tag{10.10}$$

which does not depend on θ, i.e. for a Normal location parameter the Jeffrey's prior is uniform on the real line.

Example 10.5. If y is an obs from $Y|\Theta = \theta \sim \text{Bin}(n, \theta)$, then

$$I(\theta) = -E\left(\frac{d^2(Y\log(\theta) + (n-Y)\log(1-\theta))}{d\theta^2}\right) = E\left(\frac{Y}{\theta^2} + \frac{n-Y}{(1-\theta)^2}\right)$$
$$= n\left(\frac{1}{\theta} + \frac{1}{(1-\theta)}\right) = \frac{n}{\theta(1-\theta)}.$$

Hence, the Jeffrey's prior for the Binomial likelihood is proportional to $\theta^{-1/2}(1-\theta)^{-1/2}$, i.e. a Beta$(1/2, 1/2)$-prior (an not a standard uniform as we may have expected).

Jeffrey's priors can be extended to multiparameter models (though Jeffrey himself was emphasising the use in single-parameter models), but we will not pursue this further here.

10.3 Hierarchical models

Consider a Bayesian statistical model where the parameter Θ can be partitioned into blocks $\Theta = (\Theta_1, \ldots, \Theta_d)$ where the components Θ_i may be vector valued. A *hierarchical prior* is a prior that factorises as

$$f(\theta) = f(\theta_1)f(\theta_2|\theta_1)\cdots f(\theta_d|\theta_{d-1}), \tag{10.11}$$

i.e. according to a Markov structure where Θ_i is independent of Θ_j, $j < i-1$, conditionally on $\Theta_{i-1} = \theta_{i-1}$. In a similar fashion, a *hierarchical model* is a model where the joint density $f(y, \theta)$ factorises as

$$f(y, \theta) = f(\theta_1)f(\theta_2|\theta_1)\cdots f(\theta_d|\theta_{d-1})f(y|\theta_d). \tag{10.12}$$

A model of this form is often represented graphically as

$$\Theta_1 \longrightarrow \Theta_2 \longrightarrow \cdots \longrightarrow \Theta_d \longrightarrow Y \tag{10.13}$$

emphasising that data is generated sequentially starting with Θ_1.

Hierarchical models form a convenient framework in many cases where observations can be assumed exchangable within subsets of the sample.

Example 10.6. Assume we are interested in the proportion of domestic animals in a population that carry a certain virus. In order to estimate this proportion, n randomly chosen herds are examined giving observations $y_1, \ldots, y_n$ where y_i is the number of infected animals in herd i. A simple model here would be to assume y_i is an observation of $Y_i|\Theta = \theta \sim \text{Bin}(n_i, \theta)$ where n_i is the number of animals in herd i. But this is clearly inappropriate if e.g. the virus is contagious, since it is based on the assumption that animals in a particular herd are infected independently of eachother.

More appropriate is to assume that animals are exchangable within but not across herds, which leads to the likelihood model $Y_i|P_i = p_i \sim \text{Bin}(n_i, p_i)$. If we in addition assume the herds to be exchangable, the model is completed by assuming $P_i|(A, B) = (a, b)$, $i = 1, \ldots, n$, to be independent Beta(a, b) conditionally on $(A, B) = (a, b)$ and finally A and B to be apriori independent Gamma(α_A, β_A) and Gamma(α_B, β_B) for hyperparameters $(\alpha_A, \beta_A, \alpha_B, \beta_B)$. This defines a hierarchical model with $\Theta_1 = (A, B)$ and $\Theta_2 = (P_1, \ldots, P_n)$.

In this model, the parameters Θ_1 then describes the spread of the disease at the full population-level and Θ_2 the spread at the herd-level (which is likely to be less interesting than the former).

A graphical description for $n = 3$ would be

$$\begin{array}{ccccc} & & & & P_1 \longrightarrow Y_1 \\ & & \nearrow & & \\ \Theta_1 & \longrightarrow & P_2 & \longrightarrow & Y_2 \\ & & \searrow & & \\ & & & & P_3 \longrightarrow Y_3 \end{array}$$

Example 10.7 (HMM)**.** If $\Theta_2 = (Z_1, \ldots, Z_n)$ where $\Theta_2|\Theta_1 = \theta_1$ is a stationary Markov chain with transition matrix Θ_1 and our observations can be viewed as draws from $Y = (Y_1, \ldots, Y_n)$, $Y_i = Z_i + \epsilon_i$ where the ϵ_i:s are conditionally (on $\Theta_3 = \theta_3$) independent $N(0, \theta_3)$ and finally Θ_3 is apriori independent of everything else. The joint distribution can now be factorised as

$$f(y, \theta) = f(\theta_1) f(\theta_2|\theta_1) f(\theta_3) f(y|\theta_2, \theta_3).$$

This can be described by the graph

$$\begin{array}{ccccc} \Theta_1 & \longrightarrow & \Theta_2 & \rightarrow & Y \\ & & & \nearrow & \\ & & \Theta_3 & & \end{array}$$

Chapter 11

Bayesian computation

While the posterior distribution of the full parameter vector $\Theta = (\Theta_1, \ldots, \Theta_d)$ is usually easy to derive (up to proportionality) using Bayes' theorem, it is difficult to interpret (being a function of several variables $\theta_1, \ldots \theta_d$). Hence, being able to compute marginals, i.e.

$$f(\theta_i|y) = \int f(\theta_1, \ldots, \theta_d|y)\, d\theta_1 \cdots d\theta_{i-1}\, d\theta_{i+1} \cdots d\theta_d, \tag{11.1}$$

probabilities

$$P(\Theta \in A|Y = y) = \int_A f(\theta_1, \ldots, \theta_d|y)\, d\theta_1 \cdots d\theta_d, \tag{11.2}$$

or predicitve distributions

$$f(y_{n+1}|y) = \int f(y_{n+1}, \theta_1, \ldots, \theta_d|y)\, d\theta_1 \cdots d\theta_d, \tag{11.3}$$

is crucial in order to make conclusions in a Bayesian analysis.

Historically, the above integrals have only been analytically available for rather simple models. It is with the advent of efficient MCMC simulation techniques that Bayesian statistical analysis have been popularised and its full potential realised. Indeed, with the aid of a large sample $\theta^{(1)}, \ldots, \theta^{(N)}$ from $f(\theta|y)$, approximating the above integrals is straightforward with Monte-Carlo methods since the integrands are easily derived up to proportionality using Bayes' theorem.

11.1 Using the Metropolis-Hastings algorithm

If $\Theta \in \mathbf{R}^d$, and d is not too large, a simple choice is to sample from $f(\theta|y)$ using the Metropolis-Hastings algorithm. Here we can fully appreciate the fact that the normalising constant $f(y) = \int f(y, \theta)\, d\theta$ need not be known.

11.1.1 Predicting rain

Figure 11.1 displays annual maximum daily rainfall values recorded at the Maiquetia international airport in Venezuela for the years 1951-1998. In December 1999 a daily precipitation of 410 mm caused devastation and an estimated 30000 deaths.

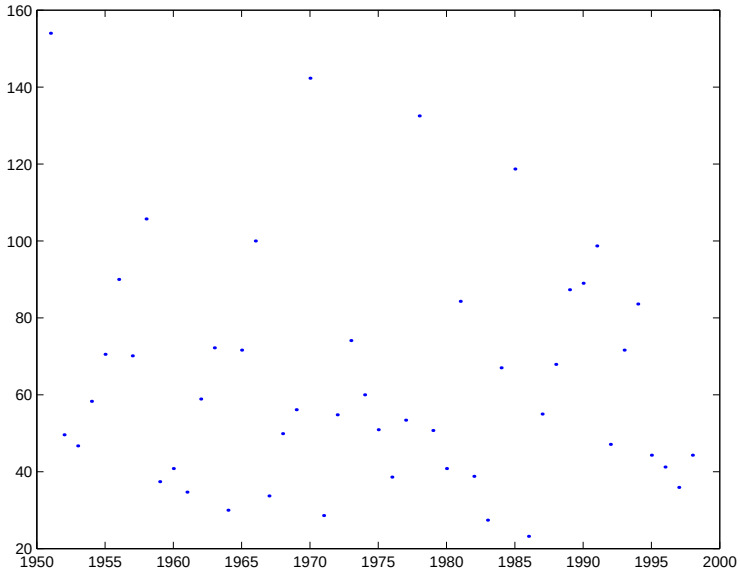

Figure 11.1: Annual maximum daily rainfall.

A popular model for extreme-values (based on asymptotic arguments, see the course MAS231/FMS155) is the Gumbel distribution with density function

$$f(x|\theta) = \frac{1}{\theta_2}\exp(\frac{x-\theta_1}{\theta_2})\exp\{-\exp(\frac{x-\theta_1}{\theta_2})\}, \theta_2 > 0. \tag{11.4}$$

It is not really appropriate for this example, since it may give negative values, but we disregard from this fact since we only aim to illustrate the compuatations involved. Our goal is to predict future rainfall. We will assume the data to be independent and Gumbel(θ) conditionally on $\Theta = \theta$. Further, we assume Θ_1 and Θ_2 to be apriori independent with improper prior $f(\theta_1, \theta_2) = \theta_2^{-1}$, $\theta_2 > 0$. This gives a posterior distribution

$$\begin{aligned} f(\theta|y) &\propto (\prod_{i=1}^{n} f(y_i|\theta)) f(\theta) \\ &\propto (\prod_{i=1}^{n} \frac{1}{\theta_2}\exp(\frac{y_i-\theta_1}{\theta_2})\exp\{-\exp(\frac{y_i-\theta_1}{\theta_2})\})/\theta_2. \end{aligned}$$

To compute the predictive density, we need to solve the integral

$$\int f(y_{n+1}, \theta|y)\, d\theta = \int \frac{f(y_{n+1}, y|\theta)f(\theta)}{f(y)}\, d\theta \tag{11.5}$$

$$\propto \int f(y_{n+1}|\theta)f(y|\theta)f(\theta)\, d\theta, \tag{11.6}$$

where $f(y_{n+1}|\theta)$ is the Gumbel density.

We start by sampling from $f(\theta|y)$ using a Gaussian random walk MH-algorithm, in Matlab

```
acc=0;
n=length(y);
th=50*ones(2,10000);
for i=1:9999;
  thprop=th(:,i)+2*mvnrnd([0 0],[32 -6;-6 16])';
  yi=(y-th(1,i))/th(2,i);
  yp=(y-thprop(1))/thprop(2);
  a=min(exp(sum(yp-yi)-sum(exp(yp)-exp(yi)))*...
                (th(2,i)/thprop(2))^(n+1)*(thprop(2)>0),1);
  if rand<a
     th(1:2,i+1)=thprop;
     acc=acc+1;
  else
     th(1:2,i+1)=th(1:2,i);
  end
end
```

where we have tuned the proposal covariance matrix as to approximately accept 25% of the proposed values. The sample is shown in Figure 11.2 and trace-plots in Figure 11.3. Convergence seems immediate but we remove the first 100 iterations as a burn-in. Figure 11.4 shows autocorrelation plots of the draws.

Now we want to predict the maximum for a future year. Since we already have 10000 draws from $\Theta|y$ and Y_{n+1} is independent of Y conditionally of Θ, we can draw produce a draw from $f(y_{n+1}|y)$ by, for each sampled value $\theta^{(i)}$ from $\Theta|y$, draw $y_{n+1}^{(i)}$ from the Gumbel($\theta^{(i)}$) distribution. The values $y_{n+1}^{(1)}, \ldots, y_{n+1}^{(10000)}$ now form a draw from the required distribution (we are using the conditional method for drawing random variables, Algorithm 3.3, here). The Gumbel distribution has distribution function

$$F(x|\theta) = 1 - \exp\{-\exp(\frac{x-\theta_1}{\theta_2})\}, \tag{11.7}$$

with inverse

$$F^{-1}(u|\theta) = \theta_1 + \log(-\log(1-u))\theta_2,$$

and is hence easy to draw from using the inversion method. 10000 predictive draws are now given by

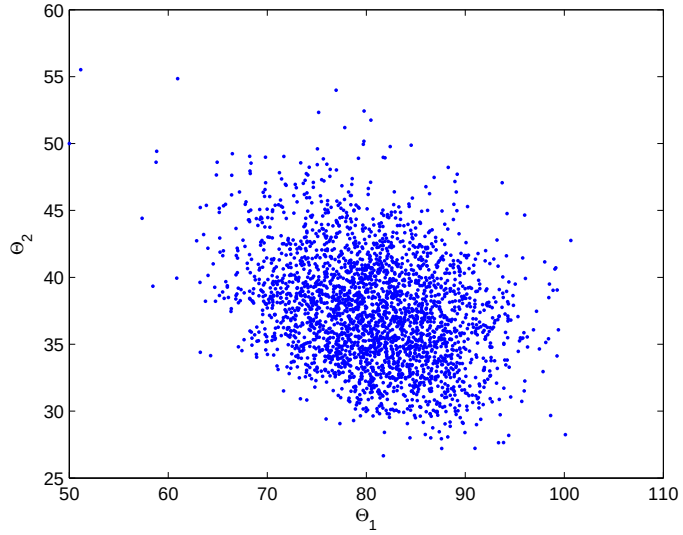

Figure 11.2: Posterior draws from $\Theta|y$.

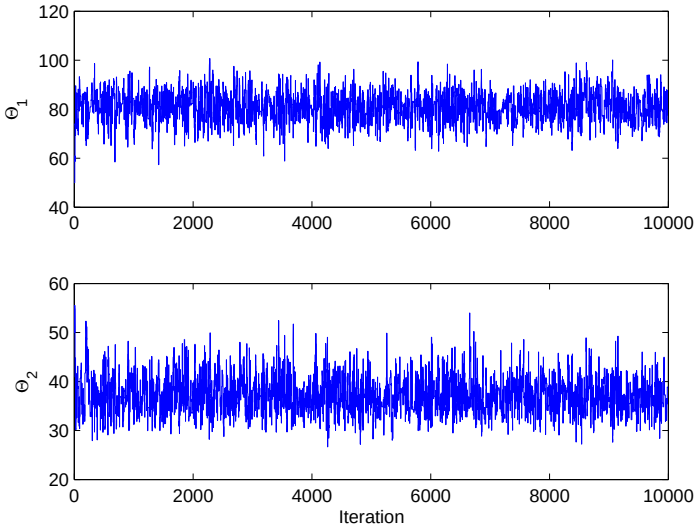

Figure 11.3: Posterior draws from $\Theta|y$.

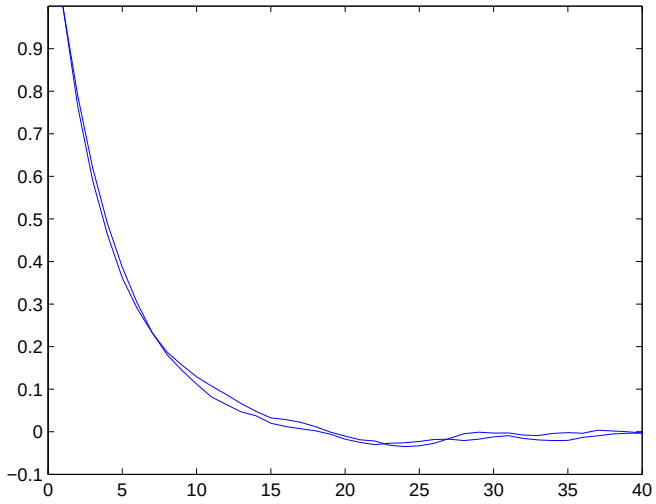

Figure 11.4: Autocorrelation plots for draws from $\Theta|y$.

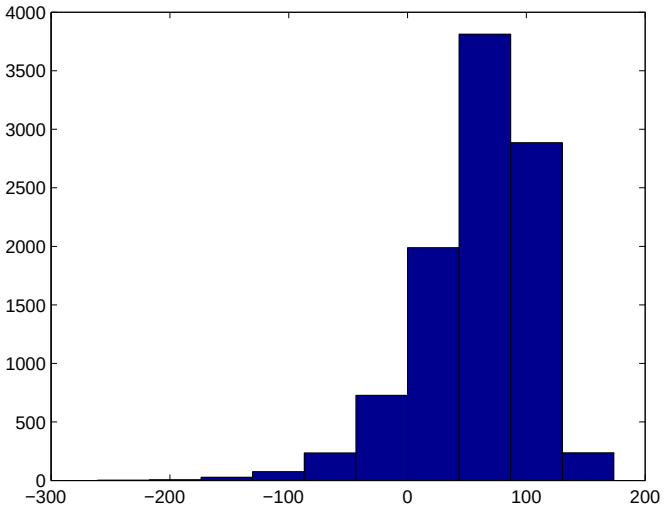

Figure 11.5: Histogram of predicted observations.

```
ypred=th(1,:)+log(-log(rand(1,10000))).*th(2,:);
```

A histogram is given in Figure 11.5. The largest of 10000 predicted values was 173.8, suggesting that the observed value of 410mm was indeed an extreme event. An alternative way of approximating the predicitve density is based on the observation that

$$f(y_{n+1}|y) = \int f(y_{n+1},\theta|y)\,d\theta = \int f(y_{n+1}|\theta,y)f(\theta|y)\,d\theta \tag{11.8}$$

$$= \int f(y_{n+1}|\theta)f(\theta|y)\,d\theta = E(f(y_{n+1}|\Theta)|y), \tag{11.9}$$

hence we can approximate $P(Y_{n+1} > y_{n+1}|y)$ by

$$\hat{P}(Y_{n+1} > y_{n+1}|y) = \frac{1}{10000}\sum_{i=1}^{10000}(1 - F(y_{n+1}|\theta^{(i)}))$$

where F is the Gumbel distribution function in (11.7). Figure 11.6 shows a plot of this function.

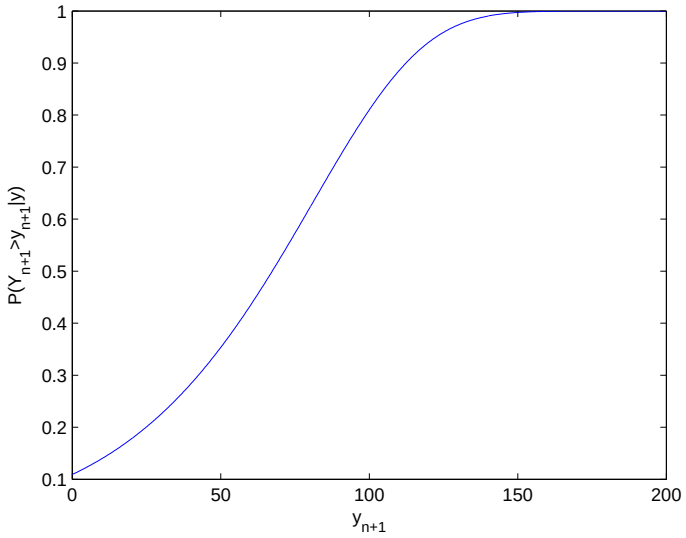

Figure 11.6: Predictive distribution $\hat{P}(Y_{n+1} > y_{n+1}|y)$ for rainfall data.

A more appropriate analysis of this example can be found in

`http://web.maths.unsw.edu.au/~scott/papers/paper_hydrology.pdf`

11.2 Using the Gibbs-sampler

Recall the concept of conjugate priors that led to simple posteriors within the same family of distribution. For complex statistical models, conjugate

priors are rarely available. However, the related property of *conditional conjugacy* often is. If the parameter is $\Theta = (\Theta_1, \ldots, \Theta_d)$, then a prior $f(\theta)$ is said to be *conditionally conjugate* for Θ_i if $f(\theta_i|\theta_{-i}, y)$ is in the same family of distributions as $f(\theta_i|\theta_{-i})$ for all y (here we write θ_{-i} for $(\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_d)$, i.e. the parameter vector with θ_i removed).

While it is not too important that $f(\theta_i|\theta_{-i}, y)$ belongs to the same class of distributions as the conditional prior, the fact that it is often of simple form (when the full posterior $f(\theta|y)$) is not) greatly simplifies computation since it implies there will be a Gibbs-sampler naturally available.

11.2.1 A poisson change-point problem

To illustrate the use of Gibbs sampling in Bayesian statistics, we will look at a series relating to the number of British coal mining disasters per year, over the period 1851 – 1962. A plot of these data is given in Figure 11.7, and the time series itself is stored in the file `coal.mat`.

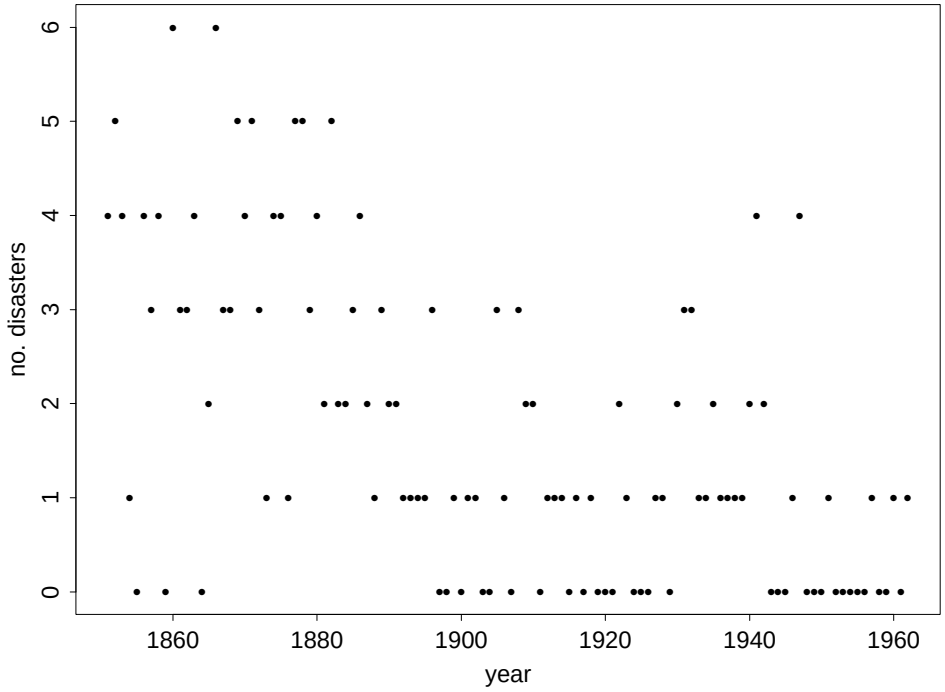

Figure 11.7: Time series of counts of coal mine disasters

From this plot it does seem to be the case that there has been a reduction in the rate of disasters over the period. We will assume there has been an abrupt change at year Θ_1 (corresponding to e.g. the introduction of a new legislation on security measurements) for which we assume an uniform prior

on $\{1,\ldots,1962-1850=n\}$. Further, we assume the number of disasters Y_i in year i follow a $\text{Po}(\Theta_2)$ distribution for $i=1,\ldots,\Theta_1$ and a $\text{Po}(\Theta_3)$ distribution for $i=\Theta_1+1,\ldots,n$. Finally we choose a hierarchical prior for (Θ_2,Θ_3) by assuming them to be independent $\text{Exp}(\Theta_4)$ where Θ_4 is apriori $\text{Exp}(0.1)$ independent of everything else.

We want to update the 4 parameters using a Gibbs-sampler, and hence compute the conditionals

$$\begin{aligned}
f(\theta_1|\theta_2,\theta_3,\theta_4,y) &\propto f(y|\theta_1,\ldots,\theta_4)f(\theta_1,\ldots,\theta_4)\\
&\propto \prod_{i=1}^{\theta_1}\frac{\theta_2^{y_i}\exp(-\theta_2)}{y_i!}\prod_{i=\theta_1+1}^{n}\frac{\theta_3^{y_i}\exp(-\theta_3)}{y_i!}\\
&\propto \theta_2^{\sum_{i=1}^{\theta_1}y_i}\exp(-\theta_1\theta_2)\theta_3^{\sum_{i=\theta_1+1}^{n}y_i}\exp(-(n-\theta_1)\theta_3)\\
&= L(\theta_1), \theta=1,\ldots,n.
\end{aligned}$$

This is a discrete distribution with

$$P(\Theta_1=i|\theta_2,\theta_3,\theta_4,y)=\frac{L(i)}{\sum_{j=1}^{n}L(j)}, i=1,\ldots,n. \tag{11.10}$$

Next

$$\begin{aligned}
f(\theta_2|\theta_1,\theta_3,\theta_4,y) &\propto f(y|\theta_1,\ldots,\theta_4)f(\theta_1,\ldots,\theta_4)\\
&\propto \theta_2^{\sum_{i=1}^{\theta_1}y_i}\exp(-\theta_1\theta_2)f(\theta_2|\theta_4)\\
&\propto \theta_2^{\sum_{i=1}^{\theta_1}y_i}\exp(-\theta_1\theta_2)\exp(-\theta_2\theta_4)
\end{aligned}$$

which we recognise as a $\text{Gamma}(1+\sum_{i=1}^{\theta_1}y_i,\theta_1+\theta_4)$ distribution. In a similar fashion we find that $f(\theta_3|\theta_1,\theta_2,\theta_4,y)$ is a $\text{Gamma}(1+\sum_{i=\theta_1+1}^{n}y_i,(n-\theta_1)+\theta_4)$ distribution. Finally,

$$\begin{aligned}
f(\theta_4|\theta_1,\theta_2,\theta_3) &\propto f(y|\theta_1,\ldots,\theta_4)f(\theta_1,\ldots,\theta_4)\\
&\propto f(\theta_2,\theta_3,\theta_4)=f(\theta_4)f(\theta_2|\theta_4)f(\theta_3|\theta_4)\\
&\propto \exp(-0.1\theta_4)\theta_4\exp(-\theta_2\theta_4)\theta_4\exp(-\theta_3\theta_4)\\
&= \theta_4^2\exp(-(0.1+\theta_2+\theta_3)\theta_4)
\end{aligned}$$

i.e. $\text{Gamma}(3,0.1+\theta_2+\theta_3)$. Hence, while the posterior distributions of $\Theta_2,\Theta_3,\Theta_4$ are *not* Gamma-distributed (you may want to check this), their conditional distributions are. This is an example of conditional conjugacy which makes Gibbs-sampling easy to implement.

A Gibbs-sampler where Θ_1 is updated using a Random-Walk MH-algorithm (adding a uniform random variable on $\{-5,\ldots,5\}$) is set up in Matlab as follows:

```
n=length(y);
th1=50*ones(1,10000);
```

```
th2=ones(1,10000);th3=th2;th4=th3;
for i=1:10000
    % Draw theta1 with RW-MH
    Li=getL(th1(i),th2(i),th3(i),y);
    th1prop=th1(i)+floor(11*rand)-5;
    Lp=getL(th1prop,th2(i),th3(i),y);
    if (rand<Lp/Li)
        th1(i+1)=th1prop;
    else
        th1(i+1)=th1(i);
    end
    th2(i+1)=gamrnd(1+sum(y(1:th1(i+1))),1/(th1(i+1)+th4(i)));
    th3(i+1)=gamrnd(1+sum(y(th1(i+1)+1:n)),1/(n-th1(i+1)+th4(i)));
    th4(i+1)=gamrnd(3,1/(0.1+th2(i+1)+th3(i+1)));
end
```

where `getL` computes $L(i)$:

```
function L=getL(t1,t2,t3,y);
n=length(y);
y1=sum(y(1:t1));
y2=sum(y(t1+1:n));
L=exp(log(t2)*y1+log(t3)*y2-t1*t2-(n-t1)*t3);
```

Sequences of draws are shown in Figures 11.8 and 11.9. Convergence seems immediate and we decide not to remove any burn-in. In Figure 11.10 we have plotted estimated autocorrelation functions for the four sequences. The draws from Θ_1 show a stronger correlation due to the inefficient Metropolis-Hastings update.

We can now make various posterior inferences for this problem. In Figure 11.11 we have plotted a histogram over the draws from $\Theta_1|y$. It seems certain that there was a change somewhere between 1886 and 1898, with 1851+41=1892 as the most likely candidate with $P(\Theta_1 = 41) \approx 0.23$ (2316 of the 10000 drawn values equal 41).

In Figure 11.12 we have plotted 20 draws from the posterior distribution of the intensity-of-disaster function, i.e. the function that equals Θ_2 until year $1851 + \Theta_1$ and then Θ_3.

The probability that there was a reduced rate of accidents is $P(\Theta_2 > \Theta_3|y)$ which roughly equals 1, since $\theta_2^{(i)}$ was greater than $\theta_3^{(i)}$ for all of the sampled pairs $(\theta_2^{(i)}, \theta_3^{(i)})$.

11.2.2 A regression model

Figure 11.13 shows a plot of hardness against density of Australian timbers. An appropriate model for this data could be a simple linear regression where we let density x be fixed and model hardness Y as

$$Y = \Theta_1 + \Theta_2 x + \epsilon, \text{ where } \epsilon \sim N(0, 1/\Theta_3), \tag{11.11}$$

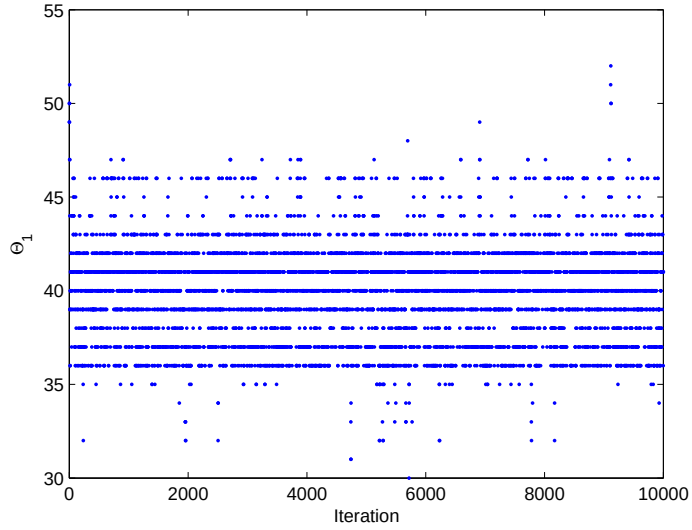

Figure 11.8: Draws from $\Theta_1|y$.

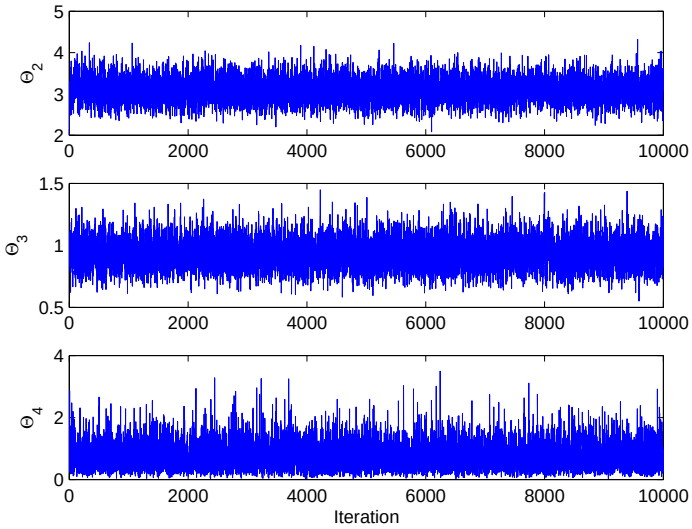

Figure 11.9: Draws from $\Theta_i|y$ for $i = 2, 3, 4$ (from top to bottom)

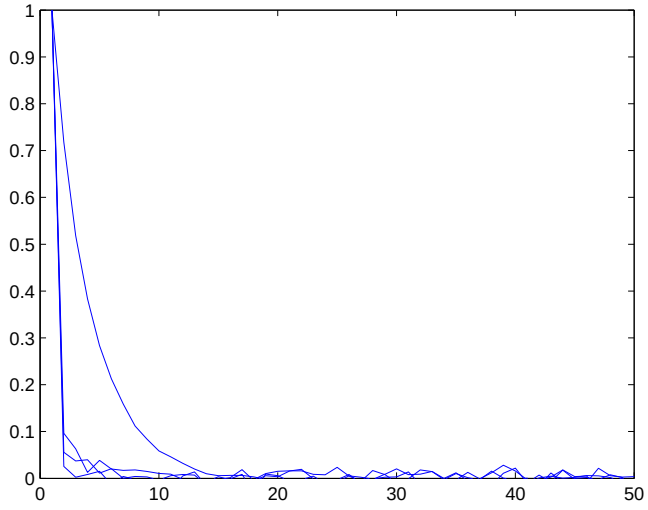

Figure 11.10: Autocorrelation plot for the draws from Θ_i, $i = 1, \ldots, 4$.

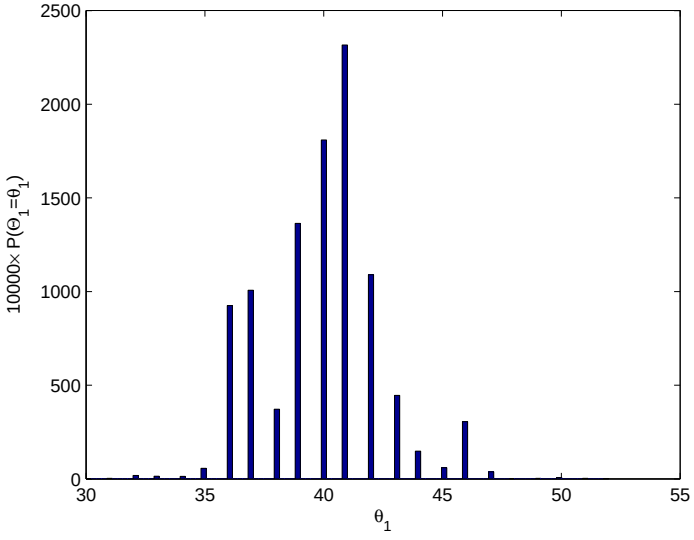

Figure 11.11: Histogram of sample from $\Theta_1|y$

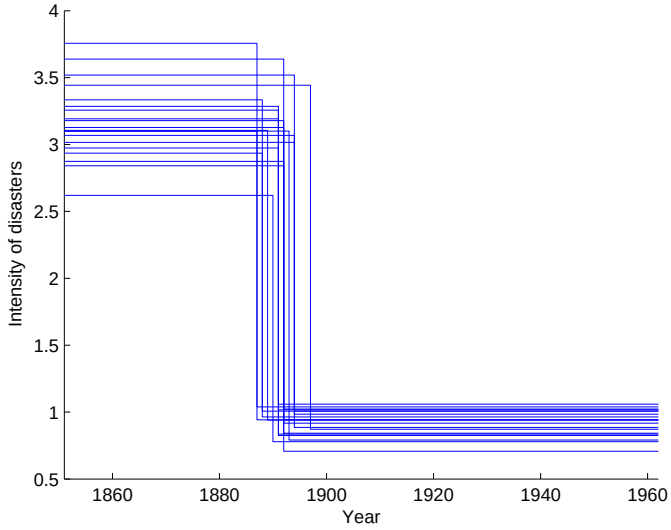

Figure 11.12: Draws from the posterior distribution of the intensity function.

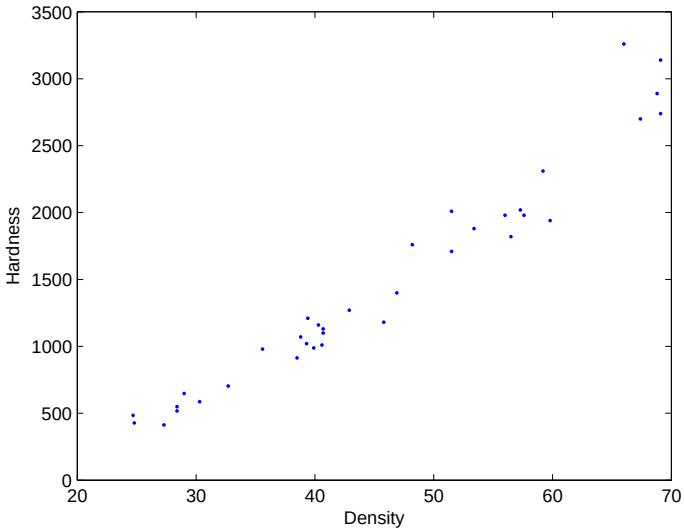

Figure 11.13: Herdness versus density for Australian timbers.

where we have choosen to work with *precision* Θ_3 rather than variance. We complete the model by choosing an improper prior $f(\theta_1, \theta_2, \theta_3) = 1$. The posterior distribution is given by

$$\begin{aligned} f(\theta|y) &\propto f(y|\theta)f(\theta) \\ &\propto \theta_3^{(n/2)} \exp(-\sum_{i=1}^{n}(y_i - \theta_1 - \theta_2 x_i)^2\theta_3/2). \end{aligned}$$

Which is of non-standard form. However

$$\begin{aligned} f(\theta_1|\theta_2, \theta_3, y) &\propto \exp(-\sum_{i=1}^{n}(y_i - \theta_1 - \theta_2 x_i)^2\theta_3/2) \\ &\propto \exp(-n\theta_1^2/2\theta_3 - \sum_{i=1}^{n}\theta_1(y_i - \theta_2 x_i)\theta_3) \\ &\propto \exp(-n * \theta_3(\theta_1 - \bar{y} + \theta_2\bar{x})^2/2, \end{aligned}$$

i.e. an $N(\bar{y}-\theta_2\bar{x}, (n\theta_3)^{-1})$ distribution. Similarily we find that $f(\theta_2|\theta_1, \theta_3, y)$ is $N((s_{xy} - \theta_0 n\bar{x})/s_{xx}, (s_{xx}\theta_3)^{-1})$ where $s_{xx} = \sum x_i^2$ and $s_xy = \sum x_i y_i$. Finally,

$$f(\theta_3|\theta_1, \theta_2, y) \propto \theta_3^{(n/2)} \exp(-\sum_{i=1}^{n}(y_i - \theta_1 - \theta_2 x_i)^2\theta_3/2),$$

which we recognise as a Gamma$(n/2+1, \sum_{i=1}^{n}(y_i - \theta_1 - \theta_2 x_i)^2/2)$ distribution.

Hence, we can implement a Gibbs-sampler in Matlab as follows:

```
n=length(y);
sxx=sum(x.^2);
sxy=sum(x.*y);
sx=sum(x);
sy=sum(y);
th=ones(3,10000);
for i=1:10000
  th(1,i+1)=normrnd((sy-th(2,i)*sx)/n,sqrt(1/(n*th(3,i))));
  th(2,i+1)=normrnd((sxy-th(1,i+1)*sx)/sxx,sqrt(1/(sxx*th(3,i))));
  th(3,i+1)=gamrnd(n/2+1,1/(sum(y-th(1,i+1)-th(2,i+1)*x)/2));
end
```

Traceplots are shown in Figure 11.14, convergence seems fast but a plot of Θ_1 against Θ_2 in Figure 11.15 shows they are strongly dependent apriori (the spread-out points in the right of the figure constitute the burn-in). The reason for the strong correlation is that the design points x are not centered around the origin; if $\bar{x}$ is large Θ_2 is constrained by the fact that the regression line has to have intercept Θ_1 and also pass through the point-cloud defined by data. On the other hand, if $\bar{x} = 0$ we see that the conditional distribution of $\Theta_1|\Theta_2, \Theta_3, y$ will not depend on Θ_2 (and similarily for

$\Theta_2|\Theta_1, \Theta_3, y$). Hence, if we reparametrise the model as

$$Y_i = \Theta_1^* + \Theta_2 z_i + \epsilon_i,$$

with $z_i = x_i - \bar{x}$ performance of the Gibbs-sampler will be improved upon (the improvement is not huge here since $\bar{x}$ is fairly small to begin with, try adding a factor 1000 to the original design points and see what happens then). Traceplots of the reparametrised model is shown in Figure 11.16, in Figure 11.17 we have made autocorrelation-plots of the parameters under the different parametrisations.

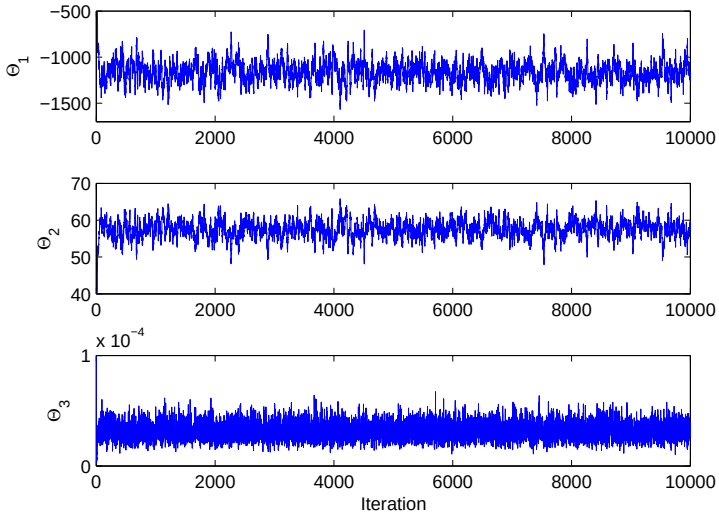

Figure 11.14: Draws from $\Theta|y$ in the regression example.

We may now be interested in predicting the hardness (which could be difficult to measure) of a timber with density 65, i.e. (in the new parametrisation) $Y_{n+1} = \Theta_1^* + \Theta_2(65 - \bar{x}) + \epsilon_{n+1}$. The predictive density is given by $f(y_{n+1}|y) = E(f(y_{n+1}|\Theta)|y)$, where $f(y_{n+1}|\Theta)$ is the $N(\Theta_1^* + \Theta_2(65 - \bar{x}), 1/\Theta_3)$ density. In Matlab

```
ypred=linspace(1700,3400,1000);
for i=1:1000
  fy(i)=mean(normpdf(ypred(i),th(1,:)+th(2,:)*19.3,1./sqrt(th(3,:))));
end
```

The results is plotted in Figure 11.18.

11.2.3 Missing data models

In a Bayesian framework there is no formal difference between missing data and unknown parameters. The algorithm corresponding to the EM-algorithm is here the Gibbs-sampler that iterates draws from missing data

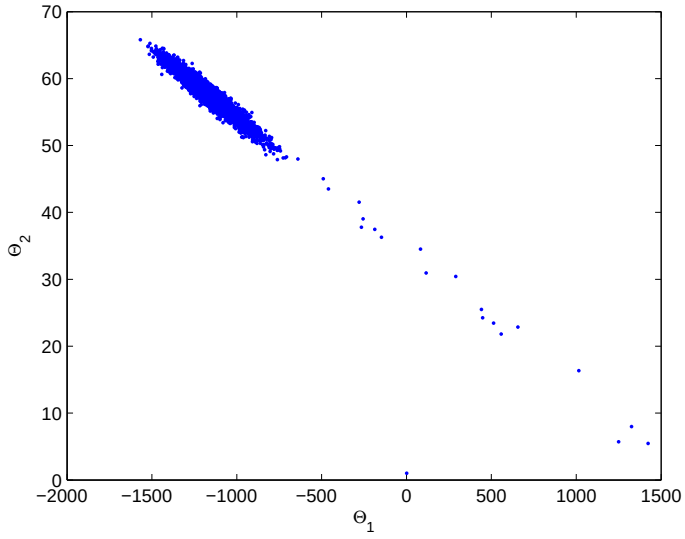

Figure 11.15: Draws from $(\Theta_1, \Theta_2)|y$ in the regression example.

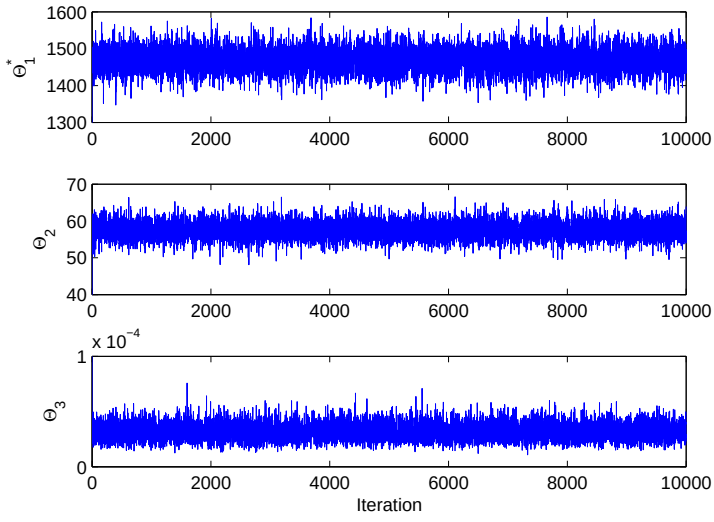

Figure 11.16: Draws from $\Theta|y$ in the regression example after reparametrisation.

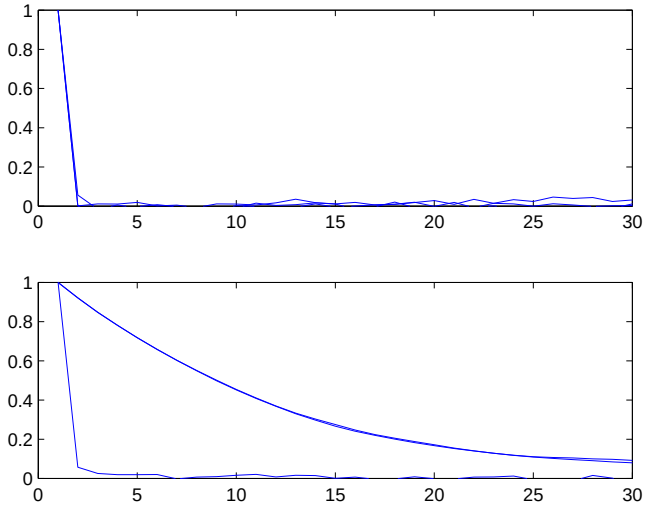

Figure 11.17: Autocorrelation plots of parameters, after reparametrisation (top) and before (bottom).

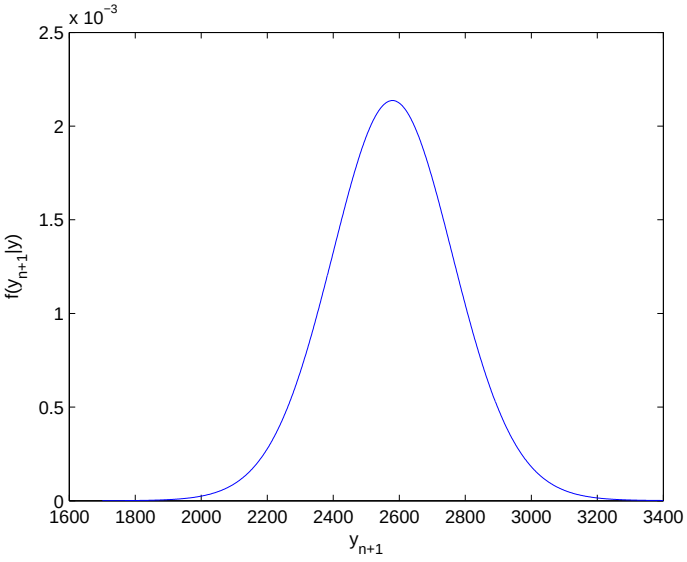

Figure 11.18: Predictive density for hardness of a timber with density 65

Z and unknown parameters Θ_1. This special case of the Gibbs-sampler is often refered to as the *Data augmentation algorithm.*

Figure 11.19 shows a histogram of service-times for customers arriving at one of two service stations. However, it is not known what service station is used. The service-time for station 1 is Exponential with mean $1/\Theta_1$ and for station 2 Exponential with mean $1/\Theta_2$. If a customer uses station 1 with probability Θ_3, the observations are draws from the mixture density

$$f(y_i|\Theta) = \Theta_3 \exp(-y_i\Theta_1)\Theta_1 + (1-\Theta_3)\exp(-y_i\Theta_2)\Theta_2. \tag{11.12}$$

Service station 1 is known to be faster, hence we choose a flat prior $f(\theta_1, \theta_2) = 1$ on $0 < \theta_2 < \theta_1$. We could sample from the posterior using a Metropolis-Hastings algorithm, but choose instead to augment the model with indicator variables $I_1, \ldots, I_n$ where $I_i = 1$ if the i:th customer used service station 1 and $I_i = 2$ is he used station 2. Hence, apriori, $P(I_i = 1) = \Theta_3$ and the model is completed by choosing a standard uniform prior on Θ_3.

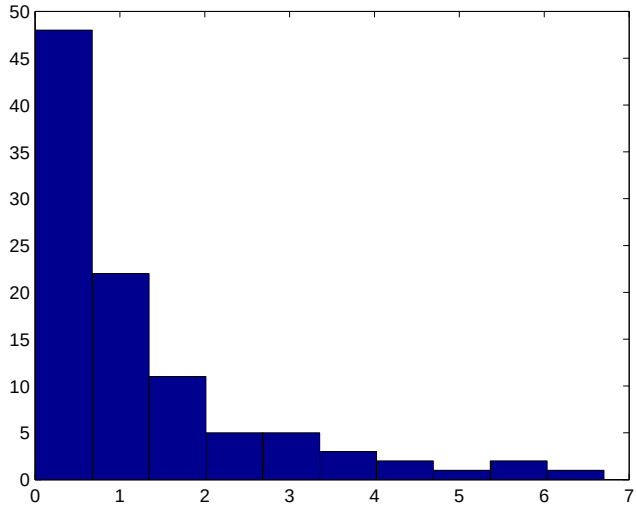

Figure 11.19: Histogram of service times

To derive the conditional distributions, we note that

$$\begin{aligned} f(\theta_1|\theta_2,\theta_3,I,y) &\propto f(y|\theta,I)f(\theta_1|\theta_2) \\ &\propto \prod_{i=1}^{n} \exp(-y_i\theta_{I_i})\theta_{I_i}\mathbf{1}\{\theta_1 > \theta_2\} \\ &\propto \theta_1^{\sum_{i=1}^{n}(2-I_i)} \exp(-\theta_1 \sum_{i=1}^{n} y_i(2-I_i))\mathbf{1}\{\theta_1 > \theta_2\} \end{aligned}$$

i.e. a Gamma$(\sum_{i=1}^{n}(2-I_i)+1, \sum_{i=1}^{n} y_i(2-I_i))$ distribution truncated on $\theta_1 > \theta_2$. Similarily, $f(\theta_2|\theta_1,\theta_3,I,y)$ will be Gamma$(\sum_{i=1}^{n}(I_i-1)+1, \sum_{i=1}^{n} y_i(I_i-$

1)) truncated on the same interval. We also have that

$$\begin{aligned} f(\theta_3|\theta_1,\theta_2,I,y) &= f(y|\theta,I)f(I|\theta_3)f(\theta_3) \\ &\propto \theta_3^{\sum_{i=1}^n(2-I_i))}(1-\theta_3)^{\sum_{i=1}^n(I_i-1)} \end{aligned}$$

since the likelihood does not depend on θ_3, and this we recognise as a Beta$(\sum_{i=1}^n(2-I_i)+1, \sum_{i=1}^n(I_i-1)+1)$ distribution. Finally,

$$P(I_i = j|I_{-i},\theta,y) \propto \exp(-y_i\theta_j)\theta_j\theta_3^{2-j}(1-\theta_3)^{j-1}$$

which implies that

$$P(I_i = j|I_{-i},\theta,y) = \frac{\exp(-y_i\theta_j)\theta_j\theta_3^{2-j}(1-\theta_3)^{j-1}}{\exp(-y_i\theta_1)\theta_1\theta_3 + \exp(-y_i\theta_2)\theta_2(1-\theta_3)},$$

conditionally independently for $i = 1,\ldots,n$. I will present a full Matlab analysis of this example on the course web-page.

October 2005, 2nd printing August 2006

Mathematical Statistics
Centre for Mathematical Sciences
Lund University
Box 118, SE-221 00 Lund, Sweden

http://www.maths.lth.se/