Chapter III
Simulating random variables

3.1 Generating Random Variables

- Inference techniques used so far have been based on **simulation**
- We now consider how to simulate X from $f_X(x)$.
- In semester 1 used MCMC - but simpler methods are needed in order to do MCMC.
- Starting point: generate U from $U[0,1]$ distribution
- Then consider transformation $g(U)$ to obtain a random draw from $f_X(x)$.

How could we generate $U[0,1]$ r.v.s with coin tosses?

Sampling from $\boldsymbol{U(0,1)}$

Need to simulate independent random variables uniformly distributed on $[0,1]$.

Definition: A sequence of pseudo-random numbers $\{u_i\}$ is a deterministic sequence of numbers in $[0,1]$ having the same statistical properties as a similar sequence of random numbers. Ripley 1987.

The sequence $\{u_i\}$ is reproducible provided u_1 is known.

A good sequence would be “unpredictable to the uninitiated”.

Congruential generators (D.H. Lehmer, 1949)

The general form of a congruential generator is

$$\begin{aligned} N_i &= (aN_{i-1} + c) \bmod M, \\ U_i &= N_i/M, \text{ where integers } a, c \in [0, M-1] \end{aligned}$$

If $c = 0$, it is called a *multiplicative congruential generator* (otherwise, *mixed*).
These numbers are restricted to the M possible values

$$0, \quad \frac{1}{M}, \quad \frac{2}{M}, \quad \dots, \quad \frac{M-1}{M}.$$

Clearly, they are *rational* numbers, but if M is large they will practically cover the reals in $[0, 1]$.

N_1: the **seed**. Can be re-set so you can reproduce same set of uniform random numbers. In R, use `set.seed(i)`, where i an integer.

As soon as some N_i repeats, say, $N_i = N_{i+T}$, then the whole subsequence repeats, i.e. $N_{i+t} = N_{i+T+t}$, $t = 1, 2, \ldots$.
The least such T is called the *period*.
A good generator will have a long period.

The period cannot be longer than M and also depends on a and c.
Several useful Theorems exist concerning periods of congruential generators. For example, for $c > 0$, $T = M$ if and only if

1. c and M have no common factors (except 1),
2. $1 = a \pmod{p}$ for every prime number that divides M,
3. $1 = a \pmod{4}$ if 4 divides M.

Usually M is chosen to make the modulus operation efficient, and then a and c are chosen to make the period as long as possible. Ripley suggests $c = 0$ or $c = 1$ is usually a good choice.

The NAG Fortran Library G05CAF

$$M = 2^{59} \qquad a = 13^{13} \qquad c = 0$$

Another recommended one is

$$M = 2^{32} \qquad a = 69069 \qquad c = 1.$$

so that

$$N_i = (69069 N_{i-1} + 1) \text{mod } 2^{32}$$

and

$$U_i = 2^{-32} N_i$$

Lattice structure

Notice that for a congruential generator

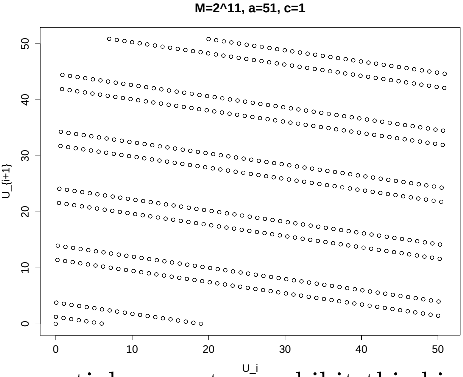
$$N_i - aN_{i-1} = c - bM,$$

where $b > 0$ is an integer. Therefore,

$$U_i - aU_{i-1} = \frac{c}{M} - b.$$

The LHS lies in $(-a, 1)$ since $U_i \in [0, 1)$.
Therefore, b can take at most $a + 1$ distinct values.

If we plot points (U_{i-1}, U_i), all the points will lie on at most $a + 1$ parallel lines.



All linear congruential generators exhibit this kind of lattice structure, not just for pairs (U_{i-1}, U_i), but also for triples (U_{i-2}, U_{i-1}, U_i), and in higher dimensions.

A good generator is expected to have *fine lattice structure*, that is, points $(U_{i-k+1}, \ldots, U_{i-1}, U_i) \in [0,1)^k$ must lie on many hyperplanes in $\mathbb{R}^k$ for all small k $(k \ll M)$.

RANDU - lattice structure

$M = 2^{31}$, $a = 2^{16} + 3 = 65539$, and $c = 0$.

Once very popular, RANDU has eventually been found out to be a rather poor generator.

RANDU - lattice structure II

Let $U_i = N_i/m$ then for this generator

$$U_{i+2} - 6U_{i+1} + 9U_i = k \text{ an integer.}$$

Since $0 \leq U_i < 1$

$$-6 < U_{i+2} - 6U_{i+1} + 9U_i < 10.$$

Therefore $k = -5, -4, \ldots, -1, 0, +1, \ldots, 9$.
Hence k can take on 15 integer values only, and subsequently (U_{i-2}, U_{i-1}, U_i) must lie on at most 15 parallel planes.

This is an example of *coarse lattice structure*, unsatisfactory coverage of $[0, 1)^3$.

Generation from non-$U(0,1)$

We have a sequence $U_1, U_2, U_3, \ldots$ of independent uniform random numbers in $[0,1]$.

We want $X_1, X_2, \ldots$ distributed independently and identically from some specified distribution.
The answer is to transform the $U_1, U_2, \ldots$ sequence into $X_1, X_2, \ldots$ sequence.
The idea is to find a function $g(U_1, U_2, U_3, \ldots)$ that has the required distribution.
There are always many ways of doing this. A good algorithm should be quick because millions of random numbers may be required.

3.2 The inversion method

Let X be any continuous random variable and define $Y = F_X(X)$, where F_X is the distribution function of X: $F_X(x) = P(X \leq x)$.
Claim: $Y \sim U[0, 1]$.
Proof $Y \in [0, 1]$ and the distribution function of Y is

$$\begin{aligned} F_Y(y) &= P(Y \leq y) = P\big(F_X(X) \leq y\big) \\ &= P\big(X \leq F_X^{-1}(y)\big) = F_X\big(F_X^{-1}(y)\big) = y \end{aligned}$$

which is the distribution function of a uniform random variable on $[0, 1]$.

So whatever the distribution of X, $Y = F_X(X)$ is uniformly distributed on $[0, 1]$. The inversion method turns this backwards. Let $U = F_X(X)$, then $X = F_X^{-1}(U)$.

- So to generate $X \sim F_X$ take a single uniform variable U, and set $X = F_X^{-1}(U)$.

Example: exponential distribution

Let $X \sim Exp(1/\lambda)$ (mean λ), i.e.

$$f(x) = \lambda^{-1}\mathrm{e}^{-x/\lambda} \quad (x \geq 0)$$

$$F(x) = \int_0^x \lambda^{-1}\mathrm{e}^{-z/\lambda}\,\mathrm{dz} = [-\mathrm{e}^{-\mathrm{z}/\lambda}]_0^{\mathrm{x}} = 1 - \mathrm{e}^{-\mathrm{x}/\lambda}.$$

Set $U = 1 - \mathrm{e}^{-X/\lambda}$ and solve for X

$$X = -\lambda \ln(1 - U).$$

Note that $1 - U$ is uniformly distributed on $[0, 1]$, so we might as well use

$$X = -\lambda \ln U.$$

Question: What are the limitations of the inversion method?

Discrete distributions

The inversion method works for discrete random variables in the following sense.
Let X be discretely distributed with possible values x_i having probabilities p_i. So

$$P(X = x_i) = p_i, \qquad \sum_{i=1}^{k} p_i = 1.$$

Then $F_X(x) = \sum_{x_i \leq x} p_i$ is a step function.
Inversion gives $X = x_i$ if $\sum_{x_j < x_i} p_j < U \leq \sum_{x_j \leq x_i} p_j$ which clearly gives the right probability values.

- Think of this as splitting $[0, 1]$ into intervals of length p_i. The interval in which U falls is the value of X.

Question: What problems might we face using this method?
Eg Consider a Poisson(100) distribution.

Discrete distributions - example

Let $X \sim \text{Bin}(4, 0.3)$. The probabilities are

$$P(X = 0) = .2401, \quad P(X = 1) = .4116, \quad P(X = 2) = .2646$$
$$P(X = 3) = .0756, \quad P(X = 4) = .0081.$$

$$\begin{aligned} \text{The algorithm says } X &= 0 \quad \text{if} \quad 0 \le U \le .2401, \\ X &= 1 \quad \text{if} \quad .2401 < U \le .6517, \\ X &= 2 \quad \text{if} \quad .6517 < U \le .9163, \\ X &= 3 \quad \text{if} \quad .9163 < U \le .9919, \\ X &= 4 \quad \text{if} \quad .9919 < U \le 1. \end{aligned}$$

Carrying out the binomial algorithm means the following. Let $U \sim U(0, 1)$.

1. Test $U \le .2401$. If true, return $X = 0$.
2. If false, test $U \le .6517$. If true, return $X = 1$.
3. If false, test $U \le .9163$. If true, return $X = 2$.
4. If false, test $U \le .9919$. If true, return $X = 3$.
5. If false, return $X = 4$.

Discrete distributions - example

Consider the speed of this. The expected number of steps (which roughly equates to speed) is

$$1 \times .2401 + 2 \times .4116 + 3 \times .2646 + 4 \times .0756 + 4 \times .0081$$

$$= 1 + E(X) - 0.0081 = 2.1919$$

To speed things up we can rearrange the order so that the later steps are less likely.

1. Test $U \leq .4116$. If true return $X = 1$.
2. If false, test $U \leq .6762$. If true return $X = 2$.
3. If false, test $U \leq .9163$. If true return $X = 0$.
4. and 5. as before.

Expected number of steps:
$1 \times .4116 + 2 \times .2646 + 3 \times .2401 + 4 \times (0.0956 + 0.0081) = 1.9959$.
Approximate 10% speed increase.

3.3 Other Transformations

(a) If $U \sim U(0,1)$ set $V = (b-a)U + a$ then $V \sim U(a,b)$ where $a < b$.

(b) If Y_i are iid exponential with parameter λ then

$$X = \sum_{i=1}^{n} Y_i = -\frac{1}{\lambda} \sum_{i=1}^{n} \log U_i = -\frac{1}{\lambda} \log \left(\prod_{i=1}^{n} U_i \right)$$

has a $Ga(n, \lambda)$ distribution.

(c) If $X_1 \sim Ga(p,1)$, $X_2 \sim Ga(q,1)$, X_1 and X_2 independent then $Y = X_1/(X_1 + X_2) \sim Be(p,q)$.

(d) Composition: if

$$f = \sum_{i=1}^{r} p_i f_i$$

where $\sum p_i = 1$ and each f_i is a density, then we can sample from f by first sampling I from the discrete distribution $p = \{p_1, \ldots, p_r\}$ and then taking a sample from f_I.

The Box-Müller algorithm for the normal distribution

We cannot generate a normal random variable by inversion, because F_X is not known in closed form (nor its inverse).

The Box–Müller method (1958). Let $U_1, U_2 \sim U[0,1]$. Calculate

$$X_1 = \sqrt{-2\ln U_1}\cos(2\pi U_2),$$
$$X_2 = \sqrt{-2\ln U_1}\sin(2\pi U_2).$$

Then X_1 and X_2 are independent $N(0,1)$ variables.

The method is not particularly fast, but is easy to program and quite memorable.

3.4 Rejection Algorithm

Fundamental Theorem of Simulation:
Simulating

$$X \sim f(x)$$

is equivalent to simulating

$$(X, U) \sim U\{(x, u) : 0 < u < f(x)\}.$$

Note that $f(x, u) = \mathbb{I}_{0<u<f(x)}$ so that

$$\int f(x, u)\mathrm{du} = \int_0^{\mathrm{f(x)}} \mathrm{du} = \mathrm{f(x)}$$

as required.

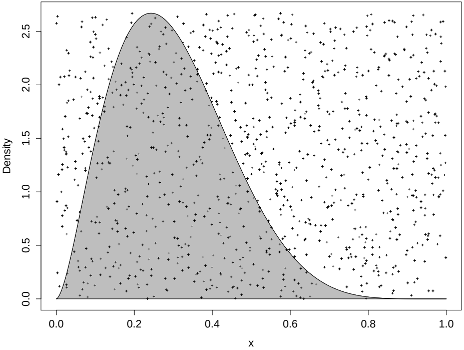
Hence, f is the marginal density of the joint distribution $(X, U) \sim U\{(x, u) : 0 < u < f(x)\}$.

Rejection Algorithm Explained

The problem with this result is that simulating uniformly from the set

$$\{(x,u) : 0 < u < f(x)\}$$

may not be possible. A solution is to simulate the pair (X,U) in a bigger set, where simulation is easier, and then take the pair if the constraint is satisfied.



Rejection: Uniform bounding box

Suppose that $f(x)$ is zero outside the interval $[a, b]$ (so that $\int_a^b f(x)\mathrm{dx} = 1$) and that f is bounded above by m.

- Simulate the pair $(Y, U) \sim U[a, b] \times [0, m]$ ($Y \sim U[a, b],\ U \sim U[0, m]$ independently).
- Accept the pair if the constraint $0 < U < f(Y)$ is satisfied.

This results in the correct distribution for the accepted Y value, call it X.

$$\begin{aligned} \mathbb{P}(X \leq x) &= \mathbb{P}(Y \leq x | U < f(Y)) \\ &= \frac{\int_a^x \int_0^{f(y)} \mathrm{dudy}}{\int_a^b \int_0^{f(y)} \mathrm{dudy}} \\ &= \int_a^x f(y)\mathrm{dy}. \end{aligned}$$

Note: we can use the rejection algorithm even if we only know f upto a normalising constant (as is often the case in Bayesian statistics - see chapter 4).

Example: Sampling from a beta distribution

Consider sampling from $X \sim \text{Beta}(\alpha, \beta)$ for $\alpha, \beta > 1$ which has pdf

$$f(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1} \quad 0 < x < 1.$$

We note

$$f(x) \propto f_1(x) = x^{\alpha-1}(1-x)^{\beta-1} \quad 0 < x < 1$$

and that $M = \sup_{0<x<1} x^{\alpha-1}(1-x)^{\beta-1}$ occurs at $x = \dfrac{\alpha-1}{\alpha+\beta-2}$ (mode) and hence

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

The rejection algorithm is

1. Generate $Y \sim U(0,1)$ and $U \sim U(0,M)$.
2. If $U \leq f_1(Y) = Y^{\alpha-1}(1-Y)^{\beta-1}$ then let $X = Y$ (accept) else go to 1 (reject).

Generalising the Rejection Idea

If the support of f is not finite, then bounding it within a rectangle will not work. Instead of using a box to bound the density $f(x)$ (ie requiring $f(x) < m$ for some constant m) we can use a function $m(x)$ such that $f(x) \leq m(x)$ for all x.

Suppose the larger bounding set is

$$\mathcal{L} = \{(y, u) : 0 < u < m(y)\}$$

then all we require is that simulation of a uniform from $\mathcal{L}$ is feasible. Note

- The closer m is to f the more efficient our algorithm.
- Because $m(x) \geq f(x)$, m cannot be a probability density. We write

$$m(x) = Mg(x) \text{ where } \int m(x)\mathrm{dx} = \int \mathrm{Mg(x)dx} = \mathrm{M}$$

for some density g.

Generalising the Rejection Idea II

This suggests a more general implementation of the fundamental theorem:

Corollary: Let $X \sim f(x)$ and let $g(x)$ be a density function that satisfies $f(x) \leq Mg(x)$ for some constant $M \geq 1$. Then, to simulate $X \sim f$, it is sufficient to generate

$$Y \sim g \qquad \text{and} \qquad U|Y = y \sim U(0, Mg(y))$$

and set $X = Y$ if $U \leq f(Y)$.

Proof:

$$\begin{aligned}
\mathbb{P}(X \in A) &= \mathbb{P}(Y \in A | U \leq f(Y)) \\
&= \frac{\int_A \int_0^{f(y)} \frac{\mathrm{du}}{Mg(y)} g(y)\mathrm{dy}}{\int \int_0^{f(y)} \frac{\mathrm{du}}{Mg(y)} g(y)\mathrm{dy}} \\
&= \int_A f(y)\mathrm{dy}
\end{aligned}$$

The Rejection Algorithm

The rejection algorithm is usually stated in a slightly modified form:

Rejection Algorithm

If g is such that f/g is bounded, so there exists M such that $Mg(x) \geq f(x)$ for all x then

1. Generate Y from density g, and U from $U(0,1)$.
2. If $U \leq f(Y)/Mg(Y)$ set $X = Y$. Otherwise, return to step 1.

produces simulations from f

We keep sampling new Y and U until the condition is satisfied.

Exercise: Convince yourself that these two descriptions of the rejection algorithm are the same.

Example: Sampling from a beta distribution revisited

Use rejection to sample from $X \sim \text{Beta}(\alpha, \beta)$. Let $g(y) = \alpha y^{\alpha-1}$, $0 < y < 1$, then

$$\frac{f_1(x)}{g(x)} = \frac{(1-x)^{\beta-1}}{\alpha} \quad \text{is bounded if and only if } \beta \geq 1$$

Then $M = \sup_x \left\{ \frac{f_1(x)}{g(x)} \right\} = \frac{1}{\alpha}$ occurs at $x = 0$.

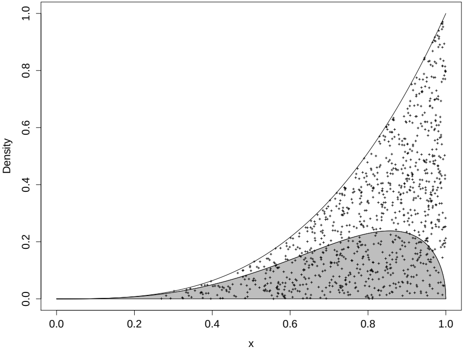
1. Simulate Y with pdf $g(y) = \alpha y^{\alpha-1}$, $0 < y < 1$ and $U \sim U(0,1)$.
2. If $U \leq \frac{f_1(Y)}{Mg(Y)} = \frac{(1-Y)^{\beta-1}}{\left(\frac{1}{\alpha}\right)\alpha} = (1-Y)^{\beta-1}$ then set $X = Y$ else go to 1.

How to simulate Y with pdf $g(y) = \alpha y^{\alpha-1}$?

- We note that the cdf of Y is $G(y) = y^{\alpha}$, $0 < y < 1$.
- Therefore we can use inversion. Let $Z \sim U(0,1)$ then solve $Z = G(Y) = Y^{\alpha}$ and so $Y = Z^{\frac{1}{\alpha}}$.

Full algorithm is:

1. Generate $U \sim U(0,1)$ and $Z \sim U(0,1)$. Let $Y = Z^{\frac{1}{\alpha}}$.
2. If $U \leq (1-Y)^{\beta-1}$ then set $X = Y$ else go to 1.



Efficiency of the rejection method

Each time we generate a (Y, U) pair,

$$\text{Prob(Reject)} = P\big(U \geq f(Y)/Mg(Y)\big) = 1 - \frac{1}{M}, \quad \text{Prob(Accept)} = \frac{1}{M}.$$

The number of tries until we accept Y is a geometric random variable with expectation M.

Note that M here must be calculated with the normalised density f, i.e., $M = \sup \frac{f(x)}{g(x)}$.
If we used an unnormalised density $f_1(x)$, where $\int f_1(x) \mathrm{d}x = c$, so that $f(x) = \frac{1}{c} f_1(x)$, then if we used

$$M = \sup \frac{f_1(x)}{g(x)}$$

the acceptance rate is

$$\mathbb{P}(\text{Accept}) = \frac{c}{M}$$

For maximum efficiency, we want M as small as possible, i.e. $\sup f(x)/g(x)$ as small as possible. This means finding a g that

(a) we can sample from efficiently, and

(b) mimics f as closely as possible.

There are many good generators based on rejection from a well-chosen g.

Rejection Example III

Let θ have von Mises distribution with pdf

$$f(\theta) = \frac{\exp(k\cos\theta)}{2\pi I(k)} \quad 0 < \theta < 2\pi \quad (k \geq 0)$$

where $I(k)$ is the normalising constant.

Let $f_1(\theta) = \frac{1}{2\pi}\exp(k\cos\theta)$, $0 < \theta < 2\pi$.

$Y \sim U(0, 2\pi)$ so that $g(y) = \frac{1}{2\pi}$, $0 < y < 2\pi$.

Then

$$M = \sup_\theta \left\{ \frac{f_1(\theta)}{g(\theta)} \right\} = \sup_\theta \{\exp(k\cos\theta)\} = \exp k.$$

Let $U \sim U(0,1)$.

If

$$U \leq \frac{f_1(Y)}{Mg(Y)} = \frac{\exp(k\cos Y)}{2\pi \cdot \frac{1}{2\pi} \cdot \exp k} = \exp\big(k(\cos Y - 1)\big)$$

we accept $\theta = Y$ otherwise reject.

Truncated distributions

Suppose we wish to sample X from the following distribution:

$$f_X(x) \propto \begin{cases} g_X(x) & \text{for } x \in A \\ 0 & \text{otherwise} \end{cases}$$

where $g_X(x)$ is a known density that we can sample from, e.g. $g_X(x)$ is the $N(0,1)$ density, and $A = [0, \infty)$.

$$f_X(x) = \begin{cases} k\, g_X(x) & \text{for } x \in A \\ 0 & \text{otherwise} \end{cases}$$

where k is a normalising constant, given by

$$k^{-1} = \int_A g_X(x) dx$$

$$f_X(x) \propto \begin{cases} g_X(x) & \text{for } x \in A \\ 0 & \text{otherwise} \end{cases}$$

Consider using rejection method to sample X from $f_X(x)$. We sample Y from the full (non-truncated) density $g_X(x)$.

$$\frac{f_X(x)}{g_X(x)} = \begin{cases} k & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases}$$

So $M = \sup_x \frac{f_X(x)}{g_X(x)} = k$.
Rejection algorithm: sample u from $U[0,1]$ and y from $g_Y(y)$, and accept $X = y$ if $u \leq \frac{f_X(y)}{M\, g_Y(y)}$.
But since

$$\frac{f_X(x)}{M\, g_X(x)} = \begin{cases} \frac{f_X(x)}{k\, g_X(x)} = 1 & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases}$$

we will always have $u \leq \frac{f_X(y)}{M\, g_Y(y)}$ if $y \in A$, and $u \geq \frac{f_X(y)}{M\, g_Y(y)}$ if $y \notin A$.
So we don't need to sample u. Can just do

1. generate y from $g_Y(y)$
2. if $y \in A$, accept $X = y$
3. otherwise, return to step 1.

As usual, acceptance probability will be high if M is small, i.e $\int_A g_Y(y)\, dy$ is near 1. So if the truncated region is large, rejection sampling will be inefficient.

3.5 Multivariate generators

Now suppose we want to generate a random vector $\boldsymbol{X} = (X_1, \ldots, X_p)$ from density $f(\boldsymbol{x})$. We can note the following simple points.

1. If the elements of $\boldsymbol{X}$ are to be independent, i.e.

   $$f(\boldsymbol{x}) = f_1(x_1) f_2(x_2) \ldots f_p(x_p),$$

   then we can separately generate X_1 from f_1, X_2 from $f_2, \ldots, X_p$ from f_p using different uniforms.
2. Inversion no longer works as the theorem can't be generalised.
3. Rejection *does* work. If we can generate from $g(\boldsymbol{x})$ (and g may be a product of independent components) and find $M \geq \sup_{\boldsymbol{x}} \dfrac{f(\boldsymbol{x})}{g(\boldsymbol{x})}$ and otherwise reject.

Sequential methods

We can obviously write

$$f(\boldsymbol{x}) = f_1(x_1) f_2(x_2|x_1) f_3(x_3|x_1, x_2) \ldots .$$

So we can first generate X_1 from f_1. Then for that given value of X_1, generate X_2 from f_2, and so on.

Example

Suppose we wish to sample $\{x_1, x_2\}$ from the density function

$$f(\theta, \phi) \propto x_2^{-\frac{1}{2}} x_2^{-(\alpha+1)} \mathrm{e}^{-\frac{2\beta+\lambda(x_1-\mu)^2}{2x_2}}$$

Firstly, consider the marginal distribution of x_1

$$f(x_1|x_2) \propto \mathrm{e}^{-\frac{\lambda(x_1-\mu)^2}{2x_2}}$$

as we can ignore factors not depending on x_1.

Thus we can recognise that

$$f(x_1|x_2) \sim N(\mu, \frac{x_2}{\lambda})$$

Next consider the marginal of x_2

$$\begin{aligned} f(x_2) &\propto \int f(x_1, x_2) \mathrm{dx}_1 \\ &\propto x_2^{-\frac{1}{2}} x_2^{-(\alpha+1)} \mathrm{e}^{-\frac{\beta}{x_2}} \left(\frac{x_2}{\lambda}\right)^{\frac{1}{2}} \\ &\propto x_2^{-(\alpha+1)} \mathrm{e}^{-\frac{\beta}{x_2}} \end{aligned}$$

where the term on the right in rd is the missing constant from the $N(\mu, \frac{x_2}{\lambda})$ distribution.

We can recognise this as an inverse gamma distribution $x_2 \sim \Gamma^{-1}(\alpha, \beta)$.

So to simulate random variables from f we can first simulate x_2 from an inverse-Gamma distribution (e.g. by rejection sampling) and then simulate $x_1 \sim N(\mu, \frac{x_2}{\lambda})$ using, e.g., Box-Muller.

Multivariate normal distributions

How can we generate $\mathbf{X}$ from a $N(\mathbf{m}, V)$ distribution, for some non-diagonal matrix V?

We know how to generate iid $N(0, 1)$ rvs from the Box-Muller algorithm, so perhaps we can take a sequence of independent standard normal random variables $Z_1, Z_2, \ldots$ and transform these in some way?

One technique involves the use of the **Cholesky square root** of the matrix V. For any (symmetric, square) positive definite matrix V, we can find a square root U (called the Cholesky decomposition), such that $U^T U = V$.

To find the Cholesky square root of a matrix V in R, type `chol(V)`.

Multivariate normal distributions II

$$\text{Set } \mathbf{Z} = \begin{pmatrix} Z_1 \\ \vdots \\ Z_n \end{pmatrix} \quad \text{where } Z_i \sim N(0,1) \quad \text{and} \quad n = \dim \mathbf{X}.$$

Consider

$$\mathbf{Y} = \mathbf{m} + U^T \mathbf{Z}.$$

Then $\mathbf{Y}$ must have a multivariate normally distribution (why?), and

$$\begin{aligned} \mathbb{E}(\mathbf{m} + U^T \mathbf{Z}) &= \mathbf{m}, \\ \mathbb{V}\mathrm{ar}(\mathbf{m} + U^T \mathbf{Z}) &= U^T I_n U = V, \end{aligned}$$

(with I_n the $n \times n$ identity matrix $= \mathbb{V}\mathrm{ar}\mathbf{Z}$).
Hence to generate $\mathbf{X}$, we generate independent standard normal random variables $\mathbf{Z}$, and then transform them by $\mathbf{m} + U^T \mathbf{Z}$ to obtain $\mathbf{X}$.

3.6 Importance sampling

In order to estimate an integral of the form $\int h(x)f(x)\mathrm{dx}$ we find that it is sometimes better to generate values not from the distribution $f(x)$, but instead from some other distribution $g(x)$ and to then account for this by using a weighting. This is the idea behind importance sampling.

To introduce the idea we consider a simple example.

Example of Monte Carlo/Importance Sampling

Let X be Cauchy $f(x) = \dfrac{1}{\pi(1+x^2)}$, $-\infty < x < \infty$.

Let $\theta = P(X > 2) = I = \displaystyle\int_2^\infty \frac{1}{\pi(1+x^2)}\,\mathrm{dx}$ $(= 0.1476)$.

Use Monte Carlo Methods to estimate θ.

(i) Generate n Cauchy variates, $X_1, \ldots, X_n$.
Let Y_1 be the number that are greater than 2,
$Y_1 = \sum \mathbb{I}_{X_i > 2}$. Then $Y_1 \sim B(n, \theta)$ so that

$$E(Y_1) = n\theta, \quad V(Y_1) = n\theta(1-\theta)$$

$$\hat{\theta}_1 = \frac{Y_1}{n}$$

$$E(\hat{\theta}_1) = \frac{E(Y_1)}{n} = \frac{n\theta}{n} = \theta$$

and

$$V(\hat{\theta}_1) = \frac{V(Y_1)}{n^2} = \frac{n\theta(1-\theta)}{n^2} = \frac{\theta(1-\theta)}{n} = \frac{0.126}{n}.$$

Example of Monte Carlo/Importance Sampling - II

(ii) Note that $\theta = \frac{1}{2}P(|X| > 2)$ - we want to use this to reduce the variance of our estimator $\hat{\theta}$.
Generate n Cauchy variates.
Let Y_2 be the number that are greater than 2 in modulus then $Y_2 \sim B(n, 2\theta)$
and $\hat{\theta}_2 = \frac{1}{2}\frac{Y_2}{n}$

$$\Longrightarrow E(\hat{\theta}_2) = \tfrac{1}{2}\frac{E(Y_2)}{n} = \tfrac{1}{2} \cdot \frac{n2\theta}{n} = \theta$$

and

$$V(\hat{\theta}_2) = \frac{V(Y_2)}{2^2n^2} = \frac{n2\theta(1-2\theta)}{2^2n^2} = \frac{\theta(1-2\theta)}{2n} = \frac{0.052}{n}.$$

Example of Monte Carlo/Importance Sampling - III

(iii) The relative inefficiency of these methods is due to generation of values outside the domain of interest $[2, \infty)$. Alternatively note we can write

$$\theta = \frac{1}{2} - \int_0^2 \frac{1}{\pi(1+x^2)} \,\mathrm{dx}.$$

This integral can be considered the expectation of $h(X) = \frac{2}{\pi(1+x^2)}$ where $X \sim U[0,2]$ as the density of $U[0,2]$ is $g(x) = 1/2$.
An alternative method of evaluation of θ is therefore

$$\hat{\theta}_3 = \frac{1}{2} - \frac{1}{n} \sum_{i=1}^{n} h(U_i)$$

where $U_i \sim U[0,2]$.

Example of Monte Carlo/Importance Sampling - IV

We can see that

$$\mathbb{E}(\hat{\theta}_3) = \frac{1}{2} - \frac{1}{n}\sum_{i=1}^{n}\int_0^2 \frac{2}{\pi(1+x^2)}\mathrm{dx} = \frac{1}{2} - \mathbb{P}(0 < \mathrm{X} < 2)$$

where $X \sim$ Cauchy, so that it too is an unbiased estimator.

The variance of $\hat{\theta}_3$ is $\mathbb{V}\mathrm{ar}(h(U))/n$ and we can see that

$$\begin{aligned}\mathbb{E}h(U) &= \int_0^2 h(x)\frac{1}{2}\mathrm{dx} = 0.5 - 0.1475 = 0.3525\\ \mathbb{E}h(U)^2 &= \int_0^2 h(x)^2\frac{1}{2}\mathrm{dx} = \int_0^2 \frac{2}{\pi^2(1+\mathrm{x}^2)^2}\mathrm{dx}\\ &= \frac{1}{\pi^2}\left[\frac{x}{x^2+1} + \tan^{-1}(x)\right]_0^2 = 0.1527\end{aligned}$$

Hence $\mathbb{V}\mathrm{ar}(h(x)) = 0.1527 - 0.3525^2 = 0.02851$ and thus

$$\mathbb{V}\mathrm{ar}(\hat{\theta}_3) = \frac{0.02851}{n}$$

Example of Monte Carlo/Importance Sampling - V

(iv) Finally, note that another possibility is to note that if $y = \frac{1}{x}$

$$\theta = \int_{+2}^{\infty} \frac{1}{\pi(1+x^2)}\ \mathrm{dx} = \int_0^{\frac{1}{2}} \frac{\mathrm{y}^{-2}\ \mathrm{dy}}{\pi(1+\mathrm{y}^{-2})} = \int_0^{\frac{1}{2}} \mathrm{h(y)\ dy}.$$

This can be seen as the expectation of $h(X) = \frac{X^{-2}}{2\pi(1+X^{-2})}$ where $X \sim U[0, \frac{1}{2}]$. We can estimate this as

$$\hat{\theta}_4 = \frac{1}{n}\sum_{i=1}^{n} h(U_i)$$

where $U_1, \dots, U_n \sim U[0, 1/2]$.

Again, we have $\mathbb{E}\hat{\theta}_4 = \theta$ and now

$\mathbb{E}h(U)^2 = \int_0^{1/2} h(x)^2 {\cdot} 2\mathrm{dx} = \frac{1}{4\pi^2}\left[\frac{\mathrm{x}}{\mathrm{x}^2+1} + \tan^{-1}(\mathrm{x})\right]_0^{1/2} = 0.02188$

$$\text{Hence } \mathbb{V}\mathrm{ar}(\hat{\theta}_4) = \frac{0.02188 - 0.1476^2}{n} = \frac{0.0000955}{n}$$

Summary of Example

We found 4 unbiased estimators of θ, each with a different variance.

$$\mathbb{V}\mathrm{ar}(\hat{\theta}_1) = \frac{0.126}{n} \qquad \mathbb{V}\mathrm{ar}(\hat{\theta}_2) = \frac{0.052}{n}$$

$$\mathbb{V}\mathrm{ar}(\hat{\theta}_3) = \frac{0.02851}{n} \qquad \mathbb{V}\mathrm{ar}(\hat{\theta}_4) = \frac{0.0000955}{n}$$

The best estimator is the one with the smallest variance, namely $\hat{\theta}_4$.

Compared with $\hat{\theta}_1$, the evaluation of $\hat{\theta}_4$ requires $\sqrt{(0.126/0.0000955)} \approx 36$ times fewer simulations to achieve the same precision.

By carefully considering our simulation method we can hope to get more accurate estimates.

Estimate $\hat{\theta}_2$ and $\hat{\theta}_4$ are both types of importance sampling.

Importance Sampling

Consider calculating the integral

$$I = \mathbb{E}_f h(X) = \int h(\boldsymbol{x}) f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}.$$

Importance sampling

Let $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ be independently and identically distributed random variables with common density $g(\boldsymbol{x})$.

Define $w(\boldsymbol{x}) = f(\boldsymbol{x})/g(\boldsymbol{x})$, so that

$$\mathbb{E}_g\{h(\boldsymbol{X}_i)w(\boldsymbol{X}_i)\} = \int h(\boldsymbol{x})w(\boldsymbol{x})g(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \int \mathrm{h}(\boldsymbol{x})\mathrm{f}(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \mathrm{I}.$$

Therefore

$$\hat{I} = \frac{1}{n} \sum_{i=1}^{n} w(\boldsymbol{X}_i) h(\boldsymbol{X}_i) \tag{1}$$

is an unbiased estimator of I.

Some comments:

- $g(\boldsymbol{x})$ is called the importance function, and $w(\boldsymbol{X}_i)$ are called the importance weights.
- The sum (1) will converge for the same reasons the Monte Carlo sum does.
- Notice that this sum is valid for any choice of the distribution g, as long as $\text{supp}(f) \subseteq \text{supp}(g)$.
- This is a very general representation that expresses the fact that a given integral is not intrinsically associated with a given distribution.
- Because very little restriction is put on the choice g, we can choose a distribution which is easy to sample from, and one which gives nice properties for the sum.

Cauchy example revisited

We can now understand the estimator $\hat{\theta}_4$ in the Cauchy example. Recall that we want to estimate

$$\mathbb{E}\mathbb{I}_{X>2} = \int h(x)f(x)\mathrm{dx}$$

where $h(x) = \mathbb{I}_{x>2}$ and $f(x) = \frac{1}{\pi(1+x^2)}$.
Noticing that for large x, $f(x)$ is similar to the density

$$g(x) = 2/x^2 \text{ for } x > 2.$$

suggests $g()$ might be a good importance density. We can sample from g by letting $X_i = 1/U_i$ where $U_i \sim U[0, \frac{1}{2}]$ (inversion method). Thus our estimator is

$$\hat{\theta} = \frac{1}{n}\sum_{i=1}^{n} h(x_i)\frac{f(x_i)}{g(x_i)} = \frac{1}{n}\sum_{i=1}^{n}\frac{x_i^2}{2\pi(1+x_i^2)}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\frac{u_i^{-2}}{2\pi(1+u_i^{-2})} = \hat{\theta}_4$$

The variance of the estimator

Since the $\boldsymbol{X}_i$s are iid, $\quad \mathbb{V}\mathrm{ar}(\hat{I}) = \dfrac{\sigma^2}{n}, \quad$ where

$$\begin{aligned}
\sigma^2 &= \mathbb{V}\mathrm{ar}_g\{h(\boldsymbol{X})w(\boldsymbol{X})\} = \mathbb{E}\{h(\boldsymbol{X})^2 w(\boldsymbol{X})^2\} - \mathbb{E}\{h(\boldsymbol{X})w(\boldsymbol{X})\}^2 \\
&= \int h(\boldsymbol{x})^2 w(\boldsymbol{x})^2 g(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} - \mathrm{I}^2 \\
&= \int \frac{h(\boldsymbol{x})^2 f(\boldsymbol{x})^2}{g(\boldsymbol{x})} \, \mathrm{d}\boldsymbol{x} - \mathrm{I}^2 \quad \text{since} \quad \mathrm{g}(\boldsymbol{x}) = \frac{\mathrm{f}(\boldsymbol{x})}{\mathrm{w}(\boldsymbol{x})}.
\end{aligned}$$

We do not of course know σ^2 in practice, but we can see that $\hat{I}$ will be a better estimator if we can make $w(\boldsymbol{X})$ less variable. Our objective, therefore, is to find a distribution $g(\boldsymbol{x})$ that we know how to obtain independent samples from, and which mimics $h(\boldsymbol{x})f(\boldsymbol{x})$ as closely as possible.

Optimal choice of g

Theorem The choice of $g = g^* = \frac{|h(x)|f(x)}{\int |h(z)|f(z)\mathrm{dz}}$ minimises the variance of the estimator (1).

Proof We've seen that it is sufficient to minimise

$$\int \frac{h^2(\boldsymbol{x})f^2(\boldsymbol{x})}{g(\boldsymbol{x})}\,\mathrm{d}\boldsymbol{x} = \mathbb{E}_{\mathrm{g}}\left(\frac{\mathrm{h}^2(\mathrm{X})\mathrm{f}^2(\mathrm{X})}{\mathrm{g}^2(\mathrm{X})}\right)$$

and using Jensen's inequality we can see that

$$\begin{aligned}\mathbb{E}_g\left(\frac{h^2(X)f^2(X)}{g^2(X)}\right) &\geq \left(\mathbb{E}_g\left[\frac{|h(X)|f(X)}{g(X)}\right]\right)^2 \\ &= \left(\int |h(x)|f(x)\mathrm{dx}\right)^2\end{aligned}$$

and that this lower bound is achieved by choosing $g = g^*$.

NB: We won't be able to calculate g^*! But the theorem suggests that choosing g to look like hf will be a good choice.

Unnormalised densites

Suppose we only know f upto a normalising constant, i.e., we know

$$f(x) = \frac{f_1(x)}{c} \qquad \text{where } c = \int f_1(x) \mathrm{dx}$$

We can still use importance sampling

Importance sampling with unnormalised densites
Let $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ be independently and identically distributed random variables with common density $g(\boldsymbol{x})$. Define $\tilde{w}(\boldsymbol{x}) = f_1(\boldsymbol{x})/g(\boldsymbol{x})$. Estimate I by

$$\hat{I} = \frac{\sum_{i=1}^{n} \tilde{w}(\boldsymbol{X}_i) h(\boldsymbol{X}_i)}{\sum_{i=1}^{n} \tilde{w}(\boldsymbol{X}_i)}$$

Alternatively, we can write this as

$$\hat{I} = \sum_{i=1}^{n} w_i h(\boldsymbol{X}_i) \qquad \text{where} \qquad w_i = \frac{\tilde{w}(X_i)}{\sum \tilde{w}(X_i)}$$

$\frac{1}{n}\sum \tilde{w}(\boldsymbol{X}_i)$ is an unbiased estimator of c as

$$\mathbb{E}_g \tilde{w}(X) = \int \frac{f_1(x)}{g(x)} g(x) \mathrm{d}x = \int f_1(x) \mathrm{d}x = c.$$

When we use unnormalised densities, $\hat{I}$ is a biased estimator of I, however it is possible to prove that we still have $\hat{I} \to I$ almost surely as $n \to \infty$.

This will be important when we use importance sampling to estimate Bayesian quantities.

Effective sample size

How variable the weights are tells us how efficient our choice of g is.

In the best case, where $g = f$, then $\tilde{w}(X) = 1$ so that $w_i = \frac{1}{n}$, which is the case in plain Monte Carlo. In this case $\mathbb{V}\text{ar}(w(X)) = 0$.

If f and g are very different, then the weights will be very variable, and we can find that one or two particles (X_i) dominate the sum.

We often calculate the **effective sample size**

$$ESS = \frac{1}{\sum w_i^2}$$

- In the best case, $w_i = \frac{1}{n}$ and ESS$= n$ - so we have an effective sample size equal to the true sample size.
- The worst case is when one of the $w_i = 1$ and all the others are equal to zero. Then ESS$= 1$, i.e., we effectively have only a single sample.

We want to choose g so that the ESS is large.

3.7 Variance reduction techniques

Antithetic variables

The method of antithetic variables uses two correlated estimators and combines them to get an estimator with a lower variance (i.e. a better estimator).

Suppose we have two different estimators $\hat{\theta}_1$ and $\hat{\theta}_2$ of θ,

- with the same mean and variance
- but which are negatively correlated

Define $\hat{\theta}_3 = \frac{1}{2}(\hat{\theta}_1 + \hat{\theta}_2)$. Then

$$\begin{aligned}
\mathbb{V}\text{ar}(\hat{\theta}_3) &= \frac{1}{4}(\mathbb{V}\text{ar}(\hat{\theta}_1) + \mathbb{V}\text{ar}(\hat{\theta}_2) + 2\mathbb{C}\text{ov}(\hat{\theta}_1, \hat{\theta}_2)) \\
&= \frac{1}{2}(\mathbb{V}\text{ar}(\hat{\theta}_1) + \mathbb{C}\text{ov}(\hat{\theta}_1, \hat{\theta}_2)) \\
&< \frac{1}{2}\mathbb{V}\text{ar}(\hat{\theta}_1)
\end{aligned}$$

This is twice the cost of computing $\hat{\theta}_1$ but the variance is more than halved!

Antithetic variables - II

We need to find two estimators which are negatively correlated. This can be done as follows:

- If $U \sim U[0,1]$ then $1-U \sim U[0,1]$ also.
- If F is the distribution function of X then $X_1 = F^{-1}(U)$ and $X_2 = F^{-1}(1-U)$ are both distributed according to F
- and $\mathbb{C}\mathrm{ov}(X_1, X_2) < 0$.

Proof (non-examinable):
Let $h(u) = F^{-1}(u)$. Then $h(u)$ is a non-decreasing function.
We need to show

$$\mathbb{E}h(U)h(1-U) \leq (\mathbb{E}h(U))^2$$

Let $Q = \mathbb{E}h(U)$. The since h is non-decreasing on $[0,1]$

$$h(0) \leq Q \leq h(1)$$

Let $f(y) = \int_0^y h(1-x)dx - Qy$ on $[0,1]$
Then $f(0) = f(1) = 0$ and

$$f'(y) = h(1-y) - Q$$

is also a non-increasing function.
Since $f'(0) = h(1) - Q \geq 0$ and $f'(1) = h(0) - Q \leq 0$ we must have

$$f(u) \geq 0 \text{ on } [0,1]$$

Therefore

$$\begin{aligned} 0 \leq \int_0^1 f(y)h'(y)dy &= [fh]_0^1 - \int f'h(y)dy \\ &= -\int_0^1 f'(y)h(y)dy \end{aligned}$$

Therefore

$$\int_0^1 f'(y)h(y) = \int_0^1 h(y)(h(1-y)-Q)dy = \int_0^1 h(y)h(1-y)dy - Q^2 \leq 0$$

Hence $\int_0^1 h(y)h(1-y)dy \leq Q^2$ as required.

Cauchy Example Revisited

Above we used

$$\hat{\theta}_3 = \frac{1}{2} - \frac{2}{n} \sum_{i=1}^{n} \left[\frac{1}{\pi(1+u_i^2)} \right]$$

as an estimator of $\mathbb{P}(X > 2)$ where $X \sim Cauchy$.

An estimator with a smaller variance can be found using antithetic variables

$$\frac{1}{2} \left(\frac{1}{2} - \frac{2}{n} \sum_{i=1}^{n} \left[\frac{1}{\pi(1+u_i^2)} \right] + \frac{1}{2} - \frac{2}{n} \sum_{i=1}^{n} \left[\frac{1}{\pi(1+(2-u_i)^2)} \right] \right)$$

which gives

$$\hat{\theta}_{antithetic} = \frac{1}{2} - \frac{1}{n} \sum_{i=1}^{n} \left[\frac{1}{\pi(1+u_i^2)} + \frac{1}{\pi(1+(2-u_i)^2)} \right]$$

The for $n = 10$ we find the variance of $\hat{\theta}_3$ is 2.7×10^{-4} whereas the variance of $\hat{\theta}_{antithetic}$ is 5.5×10^{-6} - a substantial improvement.

3.8 Bayesian inference

Unnormalised densities frequently occur when we are doing Bayesian inference.

Suppose we are interested in some posterior expectation, for example, the posterior mean:

$$I = \mathbb{E}(\theta|x) = \int \theta f(\theta|x) \mathrm{d}\theta$$

where

$$f(\theta|x) = \frac{f(\theta)f(x|\theta)}{f(x)} \qquad \text{by Bayes theorem.}$$

The denominator $f(x) = \int f(\theta)f(x|\theta)\mathrm{d}x$ is often intractable and unknown, and so we instead work with the unnormalised density

$$f_1(\theta|x) = f(\theta)f(x|\theta) = \text{ prior } \times \text{ likelihood}$$

Rejection sampling for Bayesian inference

You may have seen in MAS364 (or Autumn of MAS6004) how to sample from a posterior distribution using MCMC. We can also use rejection sampling, or estimate posterior expectations using importance sampling.

So to sample posterior samples of θ from $f(\theta|x)$, using proposal density g (assuming $f_1(\theta|x)/g(\theta) \leq M$ for all θ), we can do

1. Simulate $\theta \sim g(\cdot)$
2. Accept θ with probability

$$\frac{f(\theta)f(x|\theta)}{Mg(\theta)}$$

   otherwise reject θ.

If we use $g(\theta) = f(\theta)$, ie, use the prior as the proposal, then this reduces to accept θ with probability $\frac{f(x|\theta)}{M}$, but this is usually inefficient (ie, M is large, so the acceptance rate $1/M$ is small).

Importance sampling for Bayesian inference

Suppose we wish to estimate the posterior expectation

$$\mathbb{E}(r(\theta)|\mathbf{x}) = \int r(\theta) f(\theta|\mathbf{x}) \mathrm{d}\theta$$

We could use importance sampling, using the prior distribution as the importance distribution, ie, $g = f$.
If we do not know $f(\mathbf{x})$ then we can use the following importance sampling approach:

- Simulate $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n$ from the prior $f(\boldsymbol{\theta})$
- Set $\tilde{w}_i = f(\mathbf{x}|\boldsymbol{\theta})$
- Set $w_i = \tilde{w}_i / \sum \tilde{w}_i$ and estimate $\mathbb{E}(r(\boldsymbol{\theta})|\mathbf{x})$ by

$$\sum_{i=1}^{n} w_i r(\boldsymbol{\theta}_i)$$

This is inefficient if the prior is very different to the posterior as we will spend too much time sampling $\boldsymbol{\theta}_i$ where the likelihood is very small, and so the weights $w(\boldsymbol{\theta}_i)$ will also be very small.

If this is the case, then the effective sample size will be small.

Choice of g and the normal approximation

A more efficient alternative to using the prior distribution for g, is to build a normal approximation to the posterior and use this as g

Let $h(\boldsymbol{\theta}) = \log f(\boldsymbol{\theta}|\mathbf{x})$. Now define $\mathbf{m}$ to be posterior mode of $\boldsymbol{\theta}$, so $\mathbf{m}$ maximises both $f(\boldsymbol{\theta}|\mathbf{x})$ and $h(\boldsymbol{\theta})$.

We may need to use numerical optimisation to find $\mathbf{m}$, e.g. using the `optim` command in R.
We can then use a Taylor expansion of $h(\boldsymbol{\theta})$ around $\mathbf{m}$

$$h(\boldsymbol{\theta}) = h(\mathbf{m}) + (\boldsymbol{\theta} - \mathbf{m})^T \mathbf{h}'(\mathbf{m}) + \frac{1}{2}(\boldsymbol{\theta} - \mathbf{m})^T M(\boldsymbol{\theta} - \mathbf{m}) + \dots$$

to build a Gaussian approximation to the posterior (known as the Laplace approximation).
Here, $h'(\mathbf{m})$ the vector of first derivatives of $h(\boldsymbol{\theta})$, and M the matrix of second derivatives of $h(\boldsymbol{\theta})$, both evaluated at $\boldsymbol{\theta} = \mathbf{m}$.

Since $\mathbf{m}$ maxmises $h(\mathbf{m})$ we have $h'(\mathbf{m}) = \mathbf{0}$. Hence

$$f(\boldsymbol{\theta}|\mathbf{x}) = \exp\{h(\boldsymbol{\theta})\} \simeq \exp\{h(\mathbf{m})\} \exp\left\{-\frac{1}{2}(\boldsymbol{\theta} - \mathbf{m})^T V^{-1}(\boldsymbol{\theta} - \mathbf{m})\right\}, \tag{2}$$

where $-V^{-1} = M$.

Thus, our approximation of $f(\boldsymbol{\theta}|\mathbf{x})$ is a multivariate normal distribution, mean vector $\mathbf{m}$, variance matrix $-M^{-1}$. This will be a good approximation if posterior mass is concentrated around $\mathbf{m}$.

NB: We do not need $f(\mathbf{x})$ to obtain M, since

$$h(\boldsymbol{\theta}) = \log f(\boldsymbol{\theta}|\mathbf{x}) = \log f(\boldsymbol{\theta}) + \log f(\mathbf{x}|\boldsymbol{\theta}) - \log f(\mathbf{x}),$$

so $\log f(\mathbf{x})$ will disappear when we differentiate $h(\boldsymbol{\theta})$.

Assessing convergence

Suppose we wish to estimate $\mathbb{E}\{r(\boldsymbol{\theta})|\mathbf{x}\}$ for some $r(\boldsymbol{\theta})$. If $f(\mathbf{x})$ known, then

$$\hat{\mathbb{E}}\{r(\boldsymbol{\theta})|\mathbf{x}\} = \frac{1}{n}\sum_{i=1}^{n} r(\boldsymbol{\theta}_i)w(\boldsymbol{\theta}_i),$$

and can use central limit theorem to obtain a confidence interval for $\mathbb{E}\{r(\boldsymbol{\theta})|\mathbf{x}\}$, as in MC integration.
We can check our estimate by
1) Increasing the sample size n to check the stability of any estimate.
2) Increasing the standard deviation in the $g(\boldsymbol{\theta})$ density, to check stability to the choice of g, e.g., if we're using a normal approximation, we could multiply V by 4 etc.

Example: leukaemia data

Patients suffering from leukaemia are given a drug, 6-mercaptopurine (6-MP), and the number of days x_i until freedom from symptoms is recorded of patient i:

$$6^*, 6, 6, 6, 7, 9^*, 10^*, 10, 11^*, 13, 16, 17^*,$$

$$19^*, 20^*, 22, 23, 25^*, 32^*, 32^*, 34^*, 35^*.$$

A * denotes censored observation.
Will suppose that time x to the event of interest follows a *Weibull* distribution:

$$f(x|\alpha, \beta) = \alpha\beta(\beta x)^{\alpha-1} \exp\{-(\beta x)^\alpha\}$$

for $x > 0$.
For censored observations, we have

$$P(x > t|\alpha, \beta) = \exp\{-(\beta t)^\alpha\}.$$

Example: leukaemia data

Likelihood

Define

- d: number of uncensored observations,
- $\sum_u \log x_i$: sum of logs of all uncensored observations.

Writing $\boldsymbol{\theta} = (\alpha, \beta)^T$, the log likelihood is then given by

$$\log f(\mathbf{x}|\boldsymbol{\theta}) = d \log \alpha + \alpha d \log \beta + (\alpha - 1) \sum_u \log x_i - \beta^\alpha \sum_{i=1}^n x_i^\alpha .$$

Suppose our prior distributions for α and β are both exponential with

$$\begin{aligned} f(\alpha) &= 0.001 \exp(-0.001\alpha), \\ f(\beta) &= 0.001 \exp(-0.001\beta). \end{aligned}$$

Example: leukaemia data

Building an approximation to the posterior

1) **Obtain the posterior mode of $\boldsymbol{\theta}$.** Maximise log posteior, i.e.

$$h(\boldsymbol{\theta}) = d\log\alpha + \alpha d\log\beta + (\alpha-1)\sum_{u}\log x_i - \beta^{\alpha}\sum_{i=1}^{n} x_i^{\alpha} - 0.001\alpha - 0.001\beta +$$

for some constant K.

In R, we can find the mode to be $\mathbf{m} = (1.354, 0.030)$ using the `optim` command.

2) **Derive the matrix of second derivatives of $h(\boldsymbol{\theta})$.**

$$M = \left(\begin{array}{cc} \frac{\partial^2}{\partial \alpha^2} h(\boldsymbol{\theta}) & \frac{\partial^2}{\partial \alpha \partial \beta} h(\boldsymbol{\theta}) \\ \frac{\partial^2}{\partial \alpha \partial \beta} h(\boldsymbol{\theta}) & \frac{\partial^2}{\partial \beta^2} h(\boldsymbol{\theta}) \end{array} \right),$$

evaluated at $\boldsymbol{\theta} = \mathbf{m}$.

$$\begin{aligned}
\frac{\partial^2}{\partial \alpha^2} h(\boldsymbol{\theta}) &= -\frac{d}{\alpha^2} - \sum (\beta x_i)^\alpha (\log(\beta x_i))^2 \\
\frac{\partial^2}{\partial \beta^2} h(\boldsymbol{\theta}) &= \frac{1}{\beta^2} \left\{ \beta^\alpha \alpha (1-\alpha) \sum_{i=1}^{n} x_i^\alpha - d\alpha \right\}, \\
\frac{\partial^2}{\partial \alpha \partial \beta} h(\boldsymbol{\theta}) &= \frac{1}{\beta} \left[d - \beta^\alpha \left\{ \alpha \log \beta \sum_{i=1}^{n} x_i^\alpha + \sum_{i=1}^{n} x_i^\alpha + \alpha \sum_{i=1}^{n} x_i^\alpha \log x_i \right\} \right]
\end{aligned}$$

$$M = \left(\begin{array}{cc} -31.618 & 175.442 \\ 175.442 & -18806.085 \end{array} \right).$$

3) **Obtain the normal approximation to use as $g(\boldsymbol{\theta})$.**
$g(\boldsymbol{\theta})$: bivariate normal, mean $\mathbf{m}$, variance matrix $V = -M^{-1}$:

$$\boldsymbol{\theta} \sim N\left\{\begin{pmatrix} 1.354 \\ 0.030 \end{pmatrix}, \begin{pmatrix} 0.0334 & 0.0003 \\ 0.0003 & 0.00006 \end{pmatrix}\right\}$$

4) **Sample $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n$ from $g(\boldsymbol{\theta})$ and compute the importance weights** $w(\boldsymbol{\theta}_1), \ldots, w(\boldsymbol{\theta}_n)$. The weights are given by

$$w(\boldsymbol{\theta}_i) = \frac{\tilde{w}(\boldsymbol{\theta}_i)}{\sum_{i=1}^{n} \tilde{w}(\boldsymbol{\theta}_i)}, \qquad \text{with} \qquad \tilde{w}(\boldsymbol{\theta}_i) = \frac{f(\boldsymbol{\theta}_i) f(\mathbf{x}|\boldsymbol{\theta}_i)}{g(\boldsymbol{\theta}_i)}$$

NB the Gaussian approximation may give us negative samples. Since $\alpha > 0$ and $\beta > 0$, we shoould simply discard negative $\boldsymbol{\theta}$ values, i.e., use a truncated normal density for $g(\theta)$.

Note that when we compute $w(\boldsymbol{\theta}_i)$, it is not necessary to rescale $g(\theta)$ so that it integrates to 1, as any normalising constant in $g(\theta)$ will cancel.

5) **Estimate the posterior mean of $\boldsymbol{\theta}$**

We compute the estimate

$$\hat{E}(\boldsymbol{\theta}|\mathbf{x}) = \sum_{i=1}^{n} \boldsymbol{\theta}_i w(\boldsymbol{\theta}_i).$$

In R, with $n = 100000$, this gives $\hat{E}(\boldsymbol{\theta}|\mathbf{x}) = (1.346, 0.031)^T$.

6) **Check for convergence**

We repeat steps 4 and 5 with more dispersion in $g(\boldsymbol{\theta})$:

$g(\boldsymbol{\theta})$	$\hat{E}(\boldsymbol{\theta}\vert\mathbf{x})$
$N(\mathbf{m}, V)$	$(1.346, 0.031)^T$
$N(\mathbf{m}, 4V)$	$(1.384, 0.031)^T$
$N(\mathbf{m}, 16V)$	$(1.380, 0.031)^T$

Finally, double the sample size (no effect observed).
For percentiles, we can do resampling in R.

See computer class 5 for more details and code to implement this approach.