6. Implementation of Bayesian inference

Objective

To introduce the main numerical methods that can be used to evaluate the integrals necessary in many Bayesian problems. In particular, we concentrate on MCMC and Gibbs sampling approaches.

Recommended reading

- Wiper, M.P. (2007). Introduction to Markov chain Monte Carlo simulation. In *Encyclopedia of Statistics in Quality and Reliability*, Wiley, pp 1014–1020.

Introduction

We have seen that numerical procedures are often needed in Bayesian inference for the computation of the posterior distribution

$$p(\boldsymbol{\theta}|\mathbf{x}) = \frac{f(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int f(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})\, d\boldsymbol{\theta}}$$

and for the computation of posterior moments, predictive distributions etc. The different techniques which might be applied are as follows:

- Numerical integration,
- Gaussian approximations (considered in chapter 8),
- Monte Carlo approaches:
  - ◇ direct methods,
  - ◇ via Markov chains.

Numerical integration

Many numerical integration techniques have been developed. See for example Ausín (2007) or the Wikipedia

http://en.wikipedia.org/wiki/Numerical_integration

for fuller reviews.

One of the simplest approaches is *Simpson's rule*. Supposing that we wish to evaluate the (one dimensional) integral

$$I = \int_a^b g(x)\, dx,$$

in its most simple form, Simpson's rule suggests approximating the integral using

$$I \approx \frac{b-a}{6}\left[g(a) + 4g\left(\frac{a+b}{2}\right) + g(b)\right].$$

This approximation can be improved by subdividing the interval $[a, b]$ into an even number, say N, subintervals

$$[a, a+h) \cup \cdots \cup [a+(N-1)h, a+Nh = b].$$

Using Simpson's rule in each subinterval $[a+jh, a+(j+2)h)$ leads to the final estimate

$$\begin{aligned} I \quad \approx \quad & \frac{h}{3}\left[g(a) + 4g(a+h) + 2g(a+2h) + \cdots\right. \\ & +2g(a+(N-2)h) + 4g(a+(N-1)h) + \\ & \left.+g(a+Nh)\right]. \end{aligned}$$

Example 47

Suppose that we wish to estimate the constant of a beta density, $X \sim \mathcal{B}(7, 10)$, with density function

$$\pi(x) \propto x^6(1-x)^9 \quad \text{for } 0 < x < 1.$$

We shall try to estimate the beta function, $B(7,10) = \int_0^1 x^6(1-x)^9 dx$, using Simpson's rule. Setting $h = 0.1$, we find the following table:

θ	$\theta^6(1-\theta)^9$	$\int_0^\theta \phi^6(1-\phi)^9\, d\phi$
.0	$0.00000E-00$	$0.00000E-00$
.1	$3.87420E-07$	
.2	$8.58994E-06$	$3.37987E-07$
.3	$2.94178E-05$	
.4	$4.12783E-05$	$5.92263E-06$
.5	$3.05176E-05$	
.6	$1.22306E-05$	$1.17753E-05$
.7	$2.31568E-06$	
.8	$1.34218E-07$	$1.24962E-05$
.9	$5.31441E-10$	
1.0	$0.00000E-00$	$1.25007E-05$

The true value of the integral is $B(7,10) = 1.24875E-05$. Using Simpson's rule with $h = 0.05$ gives the result $1.24876E-05$.

An improvement on Simpson's basic rule is the adaptive Simpson's rule, which does not fix the number of subintervals a priori, but instead, continues to subdivide the intervals until the estimated error reaches a given tolerance.

Alternative approaches and problems

Other rules have been designed which take into account the form of the integrand. For example, Gaussian quadrature approaches use an approximation

$$I = \int_a^b g(x)\, dx \approx \sum_{i=1}^{N} w_i g(x_i)$$

where the points x_i are determined as the roots of a class of *orthogonal polynomials*.

The main problem with numerical integration approaches is the curse of dimensionality. As the dimension of the integral increases, the number of function evaluations necessary to achieve a given tolerance increases very rapidly. Thus, in general, such methods are not employed for higher than two or three dimensional integrals.

Monte Carlo approaches

We have seen the basic Monte Carlo method earlier in chapter 3. Suppose that we have $X \sim \pi$ and that we wish to estimate the mean of some functional $E[g(X)]$. Then given a sample, $\mathbf{x}$, of size n from π, we can estimate

$$\bar{g}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} g(x_i) \approx E[g(X)].$$

When $E[g^2(X)]$ exists, then we can estimate the sample variance using

$$V[g(\mathbf{X})] = \frac{1}{N}\int (g(x) - E[g(X)])^2 \approx \frac{1}{N^2}\sum_{i=1}^{N}(g(x_i) - \bar{g}(\mathbf{x}))^2.$$

In many cases however, there is no straightforward way of generating a sample directly from π. In such cases, two main alternatives have been considered: importance sampling and rejection sampling.

Importance sampling

Suppose that sampling from π is complicated. Suppose instead that we can easily sample from another density, say f. Now we can write the expected value of $g(X)$ (under π) as

$$\begin{aligned} E_\pi[g(X)] &= \int g(x)\pi(x)\,dx \\ &= \int \frac{g(x)\pi(x)}{f(x)} f(x)\,dx \\ &= E_f[w(X)g(X)] \end{aligned}$$

where $w(X) = \frac{\pi(X)}{f(X)}$. Thus, we can approximate the expectation by generating a sample of size N from f and using

$$E_\pi[g(X)] \approx \frac{1}{N}\sum_{i=1}^{N} w(x_i)g(x_i) \quad \text{where } w(x_i) = \frac{\pi(x_i)}{f(x_i)} \text{ for } i = 1, \ldots, N.$$

Furthermore, if the density π is known only up to an integration constant, C, then we can extend the approximation to give

$$E_\pi[g(X)] \approx \frac{\sum_{i=1}^N w(x_i)g(x_i)}{\sum_{i=1}^N w(x_i)}$$

where the denominator (divided by N) is an approximation of C.

In general, the choice of *importance function*, f, will strongly influence the efficiency of this algorithm. One should first note that the variance of the importance sampling estimator of $E[g(X)]$ is finite only when the expectation

$$E_f\left[w(X)^2 g(X)^2\right] = E_\pi\left[g(X)^2 w(X)\right] = \int g(x)^2 \frac{\pi(x)^2}{f(x)}\, dx < \infty.$$

This implies that we cannot choose importance functions with lighter tails than π. In the Bayesian context, where we often wish to estimate various posterior expectations, then an efficient importance function will be similar to the true posterior, but with heavier tails.

Example 48

Consider Example 47 where we have $X \sim \mathcal{B}(7, 10)$ so that $\pi(x) \propto x^6(1-x)^9$, and suppose that we wish to estimate the beta function $B(7, 10)$ and the posterior mean

$$E[X] = \frac{1}{B(7,10)} \int_0^1 x^7(1-x)^9 \, dx.$$

One possibility is to use importance sampling with a uniform importance function. In this case, we have importance weights

$$w(x) = \frac{\pi(x)}{1} = x^6(1-x)^9$$

and given a uniform sample of size N, we can estimate $\sum_{i=1}^N w(x_i) \approx B(7, 10)$ and $\frac{\sum_{i=1}^N x_i w(x_i)}{\sum_{i=1}^N w(x_i)} \approx E[X]$.

This is easily programmed in Matlab

```
alpha=7; beta=10; n=1000;
x=rand(1,n);
w=(alpha-1)*log(x)+(beta-1)*log(1-x);
w=exp(w);
betafunctn=sum(w);
w=w/sum(w);
meanx=sum(w.*x);
```

Given an importance sample of size $N = 1000$, the beta function was estimated to be $1.2779E-005$ (true value $1.24875E-005$) and the posterior mean was estimated at 0.4044 (true mean 0.4118). In this example, sample sizes of over 100000 are needed to achieve more than 3 figure accuracy.

Problems

One problem with this approach is that if the importance function is not similar to π (or $|g| \times \pi$) so that the centre of π (or $|g| \times \pi$) is in the tail of the importance function, then this can lead to many of the importance weights being very small and thus, the integral estimates may be largely determined by a very small number of data.

A second problem is that the importance sampling method does not provide a sample from π. This can be remedied by using sampling importance resampling (Rubin 1987).

The SIR algorithm

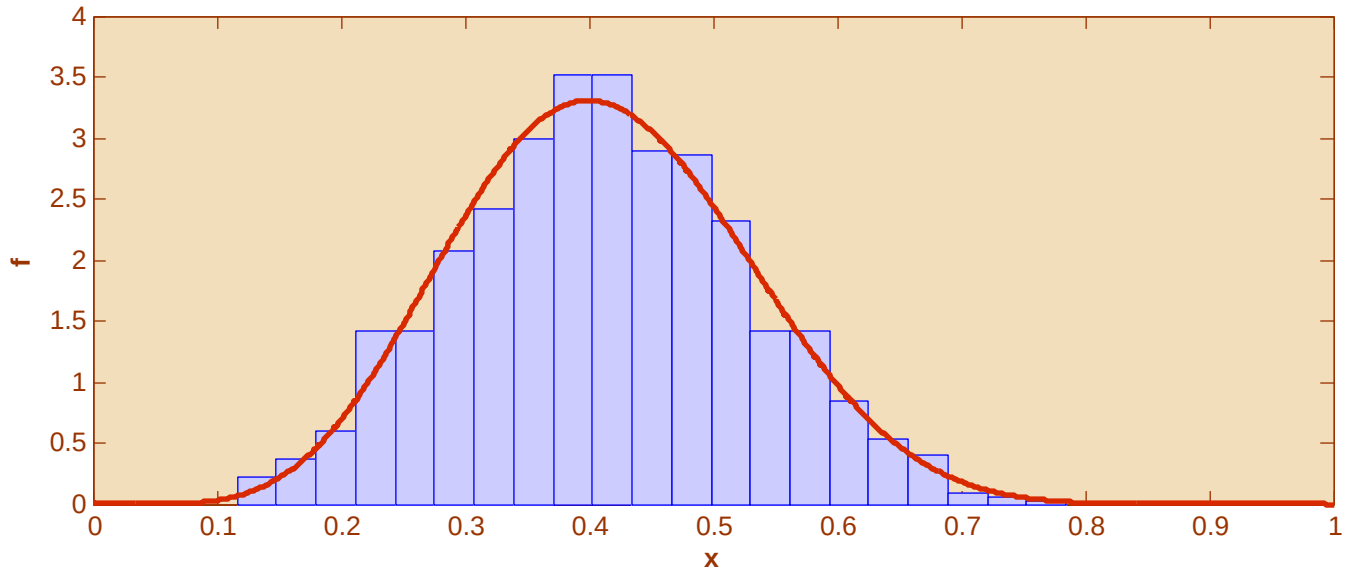
One way of obtaining an approximate sample from π is by subsampling.

If the weights, $w(x_i)$ are normalized so that we define

$$w_i = \frac{w(x_i)}{\sum_{i=1}^{N} w(x_i)}$$

then we can generate an approximate sample, $\tilde{\mathbf{x}}$, of size $M < N$ from π by setting $\tilde{x}_j = x_i$ with probability w_i for $i = 1, \ldots, N$ and $j = 1, \ldots, M$.

The following diagram shows the data simulated using a resample of size 1000 from an importance sample of size 10000.



The sampled data well approximate the beta density.

The rejection algorithm

As with standard Monte Carlo, we assume we wish to generate a sample from $\pi(x)$, which is known only up to a constant. Then the rejection approach chooses to generate data from a proposal distribution, $h(x)$, such that

$$\pi(x) < Mh(x)$$

for some given $M > 0$. The algorithm proceeds as follows.

`For` $i = 1, \ldots, N$`:`

1. `Generate` $\tilde{x}_i \sim h$`,`

2. `Generate` $u_i \sim \mathcal{U}(0, 1)$`,`

3. `If` $Mu_i h(x_i) < \pi(\tilde{x}_i)$ `set` $x_i = \tilde{x}_i$`.`

4. `Otherwise, repeat from step 1.`

Proof that this algorithm generates a sample from π

Consider $P(X < c)$, where X is generated from this algorithm. We have:

$$
\begin{aligned}
P(X \le c) &= P\left(\tilde{X} \le c | U < \pi(\tilde{X})/(Mh(\tilde{X}))\right) \quad \text{where } \tilde{X} \sim h \text{ and } U \sim \mathcal{U}(0,1) \\
&= \frac{P\left(\tilde{X} \le c, U < \pi(\tilde{X})/(Mh(\tilde{X}))\right)}{P\left(U < \pi(\tilde{X})/(Mh(\tilde{X}))\right)} \\
&= \frac{\int_{-\infty}^{c} \int_{0}^{\pi(\tilde{x})/(Mh(\tilde{x}))} h(\tilde{x})\, du\, d\tilde{x}}{\int_{-\infty}^{\infty} \int_{0}^{\pi(\tilde{x})/(Mh(\tilde{x}))} h(\tilde{x})\, du\, d\tilde{x}} \\
&= \frac{\int_{-\infty}^{c} \frac{\pi(\tilde{x})}{Mh(\tilde{x})} h(\tilde{x})\, d\tilde{x}}{\int_{-\infty}^{\infty} \frac{\pi(\tilde{x})}{Mh(\tilde{x})} h(\tilde{x})\, d\tilde{x}} \\
&= \frac{\int_{-\infty}^{c} \pi(\tilde{x})\, dx}{\int_{-\infty}^{\infty} \pi(\tilde{x})\, dx} = P(X < c) \quad \text{where } X \sim \pi.
\end{aligned}
$$

This algorithm clearly reduces to standard Monte Carlo sampling when $h = \pi$. Otherwise, as with importance sampling, it is necessary that the tails of the proposal distribution are thicker than those of π.

The main problem with this approach is finding a good proposal distribution so that only a small number of candidates are rejected. Note that the probability of accepting a draw (assuming that π is properly scaled to integrate to 1) is

$$\begin{aligned} P\left(U < \frac{\pi(\tilde{X})}{Mh(\tilde{X})}\right) &= \int_{-\infty}^{\infty} \int_{0}^{\pi(\tilde{x})/(Mh(\tilde{x}))} h(\tilde{x})\, du\, d\tilde{x} \\ &= \int_{-\infty}^{\infty} \frac{\pi(\tilde{x})}{Mh(\tilde{x})} h(\tilde{x})\, d\tilde{x} \\ &= \frac{1}{M} \end{aligned}$$

so that we would like M to be as close to 1 as possible.

Example 49

Suppose that we wish to simulate from a truncated normal distribution $X \sim \mathcal{TN}(0,1)$ where $X > \alpha > 0$. One way to do this would be to sample directly from the $\mathcal{N}(0,1)$ density and simply reject those values that fall below α. However, this method could be very inefficient if α is large. In this case, an alternative is to use a shifted, exponential distribution

$$h(x) = \lambda e^{-\lambda(x-\alpha)} \quad \text{for } x > \alpha$$

as an envelope function. (More sophisticated algorithms are proposed by Geweke (1991) and Robert (1995).)

Writing the normal density as $\pi(x) = ce^{-\frac{x^2}{2}}$, where $\frac{1}{c} = \sqrt{2\pi}\,(1 - \Phi(\alpha))$, then,

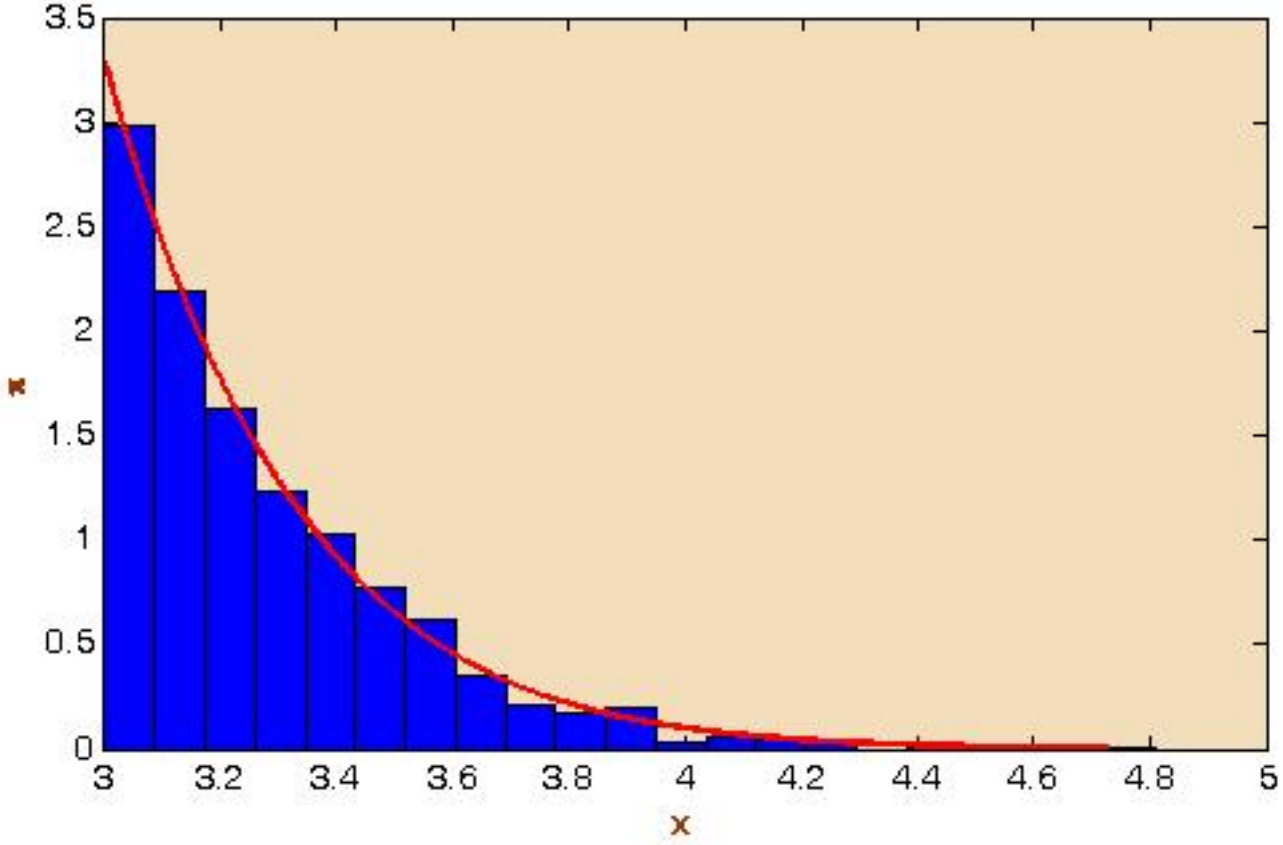
$$\begin{aligned}
\frac{\pi(x)}{h(x)} &= \frac{c}{\lambda}\exp\left(-\frac{x^2}{2} + \lambda(x-\alpha)\right) \quad \text{for } x > \alpha \\
&\leq \frac{c}{\lambda}\exp\left(\max_{x>\alpha}\left[-\frac{x^2}{2} + \lambda(x-\alpha)\right]\right) \\
&= M_1(\lambda)I(\lambda > \alpha) + M_2(\lambda)I(\lambda \leq \alpha) \quad \text{where } I \text{ is the indicator and} \\
M_1(\lambda) &= \frac{c}{\lambda}\exp\left(\frac{\lambda^2}{2} - \alpha\lambda\right) \\
M_2(\lambda) &= \frac{c}{\lambda}\exp\left(-\frac{\alpha^2}{2}\right)
\end{aligned}$$

To get the most efficient routine we should choose λ to minimize this function. However, it is simpler to minimize only M_2. This gives $\lambda = \alpha$.

In this case, the probability that a generated candidate, $\tilde{x}$ is accepted is

$$\frac{\pi(\tilde{x})}{h(\tilde{x})M_2(\alpha)} = \exp\left(-\frac{1}{2}(\tilde{x}^2 + \alpha^2) + \alpha\tilde{x}\right).$$

The following is the fitted distribution when $\alpha = 3$. Only 86 out of 1000 proposed values were rejected. We can see that the fit is very good. The probability of accepting a value generated from an untruncated normal distribution is only 0.0013.


3.5
3
2.5
2
1.5
1
0.5
0
3
3.2
3.4
3.6
3.8
4
4.2
4.4
4.6
4.8
5
x

Envelope methods

These are refinements of the basic algorithm based on bounding the target density from above and below. Suppose that we can find a proposal density h and a (non-negative) function g such that

$$g(x) \leq \pi(x) \leq Mh(x) \quad \text{for all } x.$$

Then, the following algorithm generates a variable, X, with distribution π.

1. Generate $\tilde{X} \sim h$ and $U \sim \mathcal{U}(0,1)$.

2. If $U \leq \frac{g(\tilde{X})}{Mh(\tilde{X})}$ let $X = \tilde{X}$.

3. Otherwise, let $\tilde{X} = X$ if $U \leq \frac{\pi(\tilde{X})}{Mh(\tilde{X})}$

4. Otherwise, repeat from step 1.

The advantage of this algorithm is that the number of necessary evaluations of π are reduced, and instead, we often only need to evaluate the (simpler) densities, g and h. The probability that π does not have to be evaluated is $\frac{1}{M}\int g(x)\,dx$ which reflects the potential gain in using this approach.

One particular case that allows for the simple construction of bounding functions is when the density, π, is $log\ concave$.

Definition 11

A density $f(x)$ is said to be log concave if $\frac{\partial^2}{\partial x^2} f(x) < 0 \ \forall\ x$.

Most exponential family densities are log-concave. For example, if $X|\theta \sim \mathcal{N}(\theta, 1)$, then $\frac{\partial^2}{\partial x^2} f(x|\theta) = -1$.

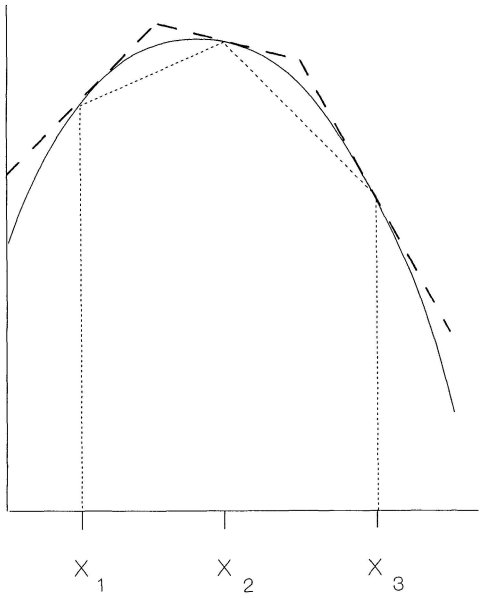
Adaptive rejection sampling

This algorithm developed by Gilks (1992) and Gilks and Wild (1992) gives a general method of constructing the bounding functions g and h when the target density, π, is log concave.

Suppose that S_n is a set of points x_i for $i = 0, 1, \ldots, n+1$ in the support of π such that $\log \pi(x_i)$ is known up to the same constant. As log π is concave, then the line $L_{i,i+1}$ going through $(x_i, \log \pi(x_i))$ and $(x_{i+1}, \log \pi(x_{i+1}))$ lies below the graph of $\log \pi$ in $(x_i, x_{i+1}]$ and lies above the graph outside this interval. Thus, for the interval $(x_i, x_{i+1}]$, we can define $\bar{\phi}_n(x) = \min L_{i-1,i}(x), L_{i+1,i+2}(x)$ and $\underline{\phi}_n(x) = L_{i,i+1}(x)$ which bound $\log \pi$. Defining $H_n(x) = \exp(\bar{\phi}_n(x))$ and $g_n(x) = \exp(\underline{\phi}_n(x))$, we have

$$g_n(x) \leq \pi(x) \leq H_n(x) = M_n h_n(x) \text{say}$$

where h_n is a density function.



An advantage of this approach is that if a generated value is rejected, it can then be added to the set S_n which improves the bounds on π at the next step. This leads to the following generic algorithm.

The ARS algorithm

1. Initialize n and S_n.

2. Generate $\tilde{X} \sim h_n$ and $U \sim \mathcal{U}(0,1)$.

3. If $U < g_n(\tilde{X})/h_n(\tilde{X})$ then set $X = \tilde{X}$.

4. Otherwise, if $U < \pi(\tilde{X})/(M_n h_n(\tilde{X})$, set $X = \tilde{X}$.

5. Otherwise, set $n = n + 1$, $S_{n+1} = S_n \cup \tilde{X}$ and repeat from 2.

The big advantage of this algorithm is its universality. As long as π is known to be log concave, it can always be used.

MCMC methods

As simple Monte Carlo algorithms are not always straightforward to implement, another alternative is to use algorithms which generate approximate Monte Carlo samples. The most popular approach is Markov chain Monte Carlo or MCMC which samples from a Markov chain whose limit distribution is the distribution from which we wish to sample.

Markov chains

Definition 12

A *Markov chain*, $\{X_t\}$, is defined to be a sequence of variables, $X_0, X_1, X_2, \ldots$, such that the distribution of X_t given the previous values $X_0, \ldots, X_{t-1}$ only depends on X_{t-1}, so that

$$P(X_t \in A | X_0 = x_0, X_1 = x_1, \ldots, X_{t-1} = x_{t-1}) = P(X_t \in A | X_{t-1} = x_{t-1})$$

for all $A, x_1, \ldots, x_{t-1}$.

Most Markov chains that we deal with are *time-homogeneous*, that is

$$P(X_{t+k} \in A | X_t = x) = P(X_k \in A | X_0 = x) \quad \text{for any } k.$$

A simple example of a time-homogeneous Markov chain is a random walk

$$X_t = X_{t-1} + \epsilon_t \quad \text{where } \epsilon_t \sim \mathcal{N}\left(0, \sigma^2\right)$$

It is clear that a time-homogeneous Markov chain is completely defined by the initial state, X_0, and by the *transition kernel*,

$$P(x, y) = P(X_{t+1} = y | X_t = x).$$

For most problems of interest, the Markov chain will take values in a continuous, multivariate state space. However, we shall assume initially that the state space is finite and countable, so that we can assume that $X_t \in \{1, 2, \ldots, k\}$ for some k.

In this case, we can define the *t-step transition probabilities*

$$p_{ij}(t) = P(X_t = j | X_0 = i)$$

and then, we can consider the conditions under which these probabilities converge, i.e. that

$$p_{ij}(t) \to \pi(j) \quad \text{as } t \to \infty.$$

Definition 13

A Markov chain is said to be $irreducible$ if for every i, j, there exists some t such that $p_{ij}(t) > 0$.

Irreducibility implies that it is possible to visit every state in the chain starting from any initial state.

Definition 14

A state, i, of a Markov chain is said to be $recurrent$ if return to state i is certain, that is if we define τ_i to be the number of steps needed to return to state i, then $P(\tau_i < \infty) = 1$. A recurrent state is further said to be $positive\ recurrent$ if the expected return time is finite, so that $E[\tau_i] < \infty$.

Definition 15

The $period$ of a state, i, is defined to be $d(i) = \gcd\{t : p_{ii}(t) > 0\}$. A state with period 1 is said to be $aperiodic$.

It can be shown that if any state of an irreducible chain is positive recurrent, then all states are and also that all states in such a chain have the same period.

The equilibrium distribution of a Markov chain

Theorem 28

For an irreducible, positive definite, aperiodic Markov chain with t step transition density $p_{ij}(t)$, then a unique equilibrium distribution π exists so that for all i, j,

$$\pi(j) = \lim_{t \to \infty} p_{ij}(t).$$

Proof ∎

It can be shown that a sufficient condition for the existence of a unique stationary distribution is *reversibility*. A Markov chain with transition probabilities $p_{ij} = P(X_{t+1} = j | X_t = i)$ is said to be reversible if there exists a probability density π that satisfies *detailed balance*, so that for any i, j, then

$$p_{ij}\pi(i) = p_{ji}\pi(j).$$

Markov chains with continuous state space

It is possible to extend the previous arguments to Markov chains with a continuous state space, although the conditions for the equilibrium distribution are slightly more technical, see e.g. Robert and Casella (2004). In this case, given a transition kernel, $P(x, y)$, then a stationary distribution π must satisfy

$$\pi(y) = \int P(x, y)\pi(x)\, dx.$$

From a Bayesian viewpoint, the objective of the MCMC approach is thus to construct a Markov chain with a given stationary distribution π which is the Bayesian posterior distribution.

The Metropolis Hastings algorithm

Metropolis

This is a general algorithm for constructing a Markov chain and was introduced by Metropolis et al (1953) and extended by Hastings (1970). The general algorithm for generating a chain with equilibrium distribution π is as follows:

The algorithm

1. Given the current value, $X_t = x$, generate a candidate value, y, from a proposal density $q(y|x)$.

2. Calculate the acceptance probability

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

3. With probability $\alpha(x, y)$ define $X_{t+1} = y$ and otherwise reject the proposed value and set $X_{t+1} = x$.

4. Repeat until convergence is judged and a sample of the desired size is obtained.

Why does this algorithm work?

The transition kernel of a move from x to y is

$$P(x,y) = \alpha(x,y)q(y|x) + \left(1 - \int \alpha(x,y)q(y|x)\,dy\right)\delta_x$$

where δ_x is the Dirac delta function at x.

Now, it is easy to show that

$$\pi(x)q(y|x)\alpha(x,y) = \pi(y)q(x|y)\alpha(y,x)$$

and that

$$\left(1 - \int \alpha(x,y)q(y|x)\,dy\right)\delta_x = \left(1 - \int \alpha(y,x)q(x|y)\,dx\right)\delta_y.$$

This implies that we have detailed balance

$$\pi(x)P(x,y) = \pi(y)P(y,x)$$

so that π is a stationary distribution of the chain. ∎

It is important to notice that the Metropolis Hastings acceptance probability only depends on pi through the ratio $\pi(y)/\pi(x)$. This is particularly useful in the Bayesian context, when the form of the posterior distribution is usually known up to the constant of integration.

Also note that when the proposal density $q(y|x) = \pi(y)$, then the Metropolis Hastings acceptance probability is exactly 1 and the algorithm is the same as standard Monte Carlo sampling.

Choosing a proposal density

One might expect that the Metropolis Hastings algorithm would be more efficient if $\alpha(x, y)$ was high. Unfortunately, this is not usually the case. In Roberts et al (1997), it is recommended that for high dimensional models, the acceptance rate for random-walk algorithms (see later) should be around 25% whereas in models of dimension 1 or 2, this should be around 50%.

However, general results are not available and the efficiency of a Metropolis Hastings algorithm is usually heavily dependent on the proposal density $q(y|x)$.

The independence and Metropolis samplers

The *independence sampler* defines a proposal density $q(y|x) = q(y)$ independent of x. This will often work well if the density q is similar to π, although with somewhat heavier tails, similarly to the Monte Carlo rejection sampler.

Another alternative is the *Metropolis* (1953), sampler which has the property that $q(x|y) = q(y|x)$. One small advantage of this approach is that the acceptance probability simplifies down to the

$$\alpha(x, y) = \frac{\pi(y)}{\pi(x)}.$$

A special case is the *random walk Metropolis* algorithm which assumes that $q(y|x) = q(|y - x|)$. For example, in univariate problems, one might consider a normal proposal density $q(y|x) = \mathcal{N}\left(x, \sigma^2\right)$ where the value of σ can be adjusted to achieve an acceptable acceptance rate.

Example 50

Suppose that we observe a sample of size n from a Cauchy distribution, $X|\theta \sim \mathcal{C}(\theta, 1)$, that is

$$f(x|\theta) = \frac{1}{\pi\left(1 + (x-\theta)^2\right)} \quad \text{for } -\infty < x < \infty$$

Given a uniform prior for θ, then the posterior distribution is

$$p(\theta|\mathbf{x}) \propto \prod_{i=1}^{n} \frac{1}{1 + (x_i - \theta)^2}.$$

One way of sampling this distribution is to use a random walk Metropolis algorithm. We could use a Cauchy proposal density, $\tilde{\theta}|\theta \sim \mathcal{C}(\theta, \sigma)$, so that

$$q(\tilde{\theta}|\theta) = \frac{1}{\pi\sigma\left(1 + \left(\frac{\tilde{\theta}-\theta}{\sigma}\right)^2\right)}.$$

The scale parameter, σ, can be adjusted to achieve the desired acceptance rate.

In this case, the probability of accepting a proposed value, $\tilde{\theta}$ given the current value, θ, is

$$\alpha(\theta, \tilde{\theta}) = \min\left\{1, \prod_{i=1}^{n} \frac{1+(x_i-\theta)^2}{1+(x_i-\tilde{\theta})^2}\right\}.$$

As an alternative, an independence sampler could be proposed. In this case, we might assume a Cauchy proposal distribution, $\tilde{\theta} \sim \mathcal{C}(m, \tau)$, where the location parameter, m, is the sample median. In this case, the acceptance probability is

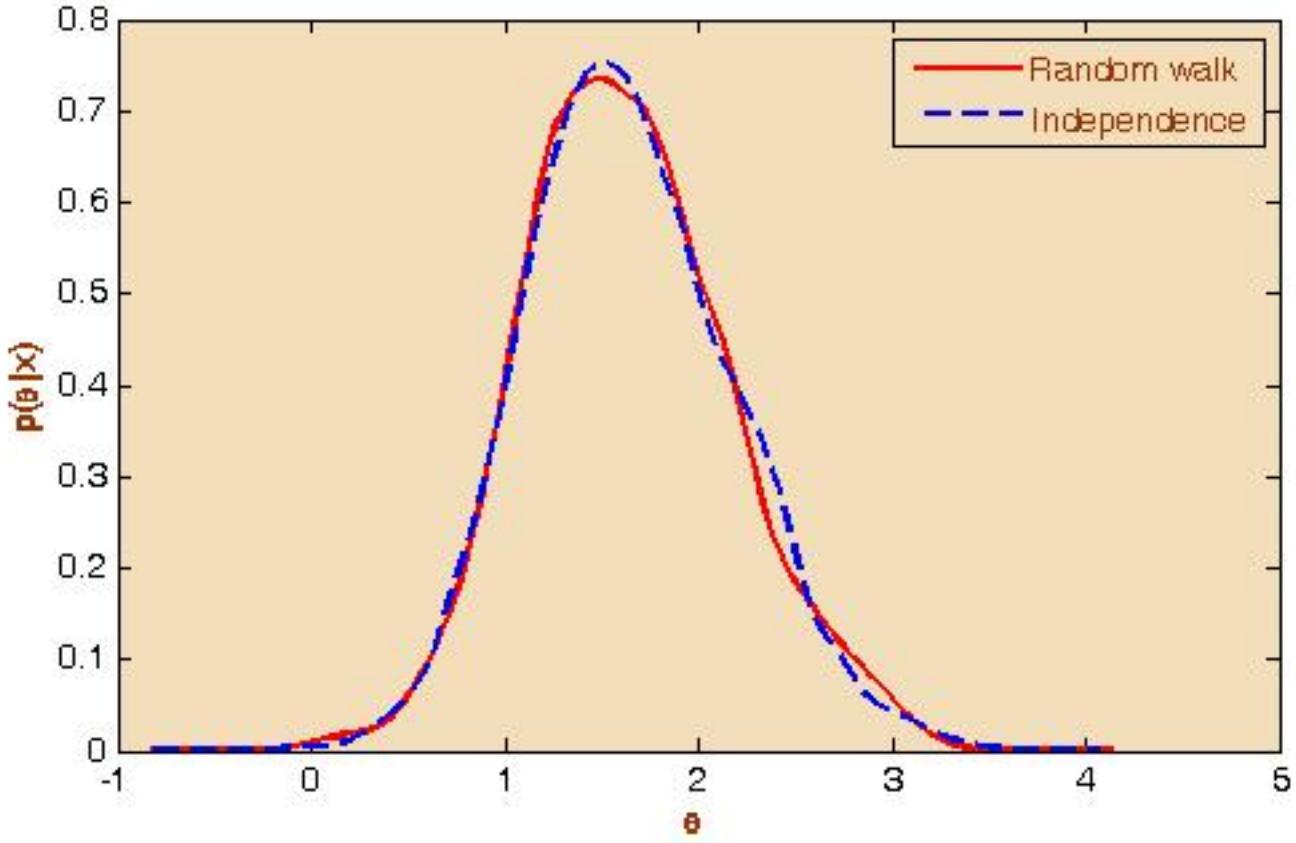
$$\min\left\{1, \prod_{i=1}^{n} \frac{1+(x_i-\theta)^2}{1+(x_i-\tilde{\theta})^2} \frac{1+\left(\frac{\tilde{\theta}-m}{\tau}\right)^2}{1+\left(\frac{\theta-m}{\tau}\right)^2}\right\}.$$

A sample of 10 data were generated from a Cauchy distribution, $X \sim \mathcal{C}(1, 1)$, with the following results:

$$\mathbf{x} = \begin{array}{rrrrr} -5.1091 & -0.7651 & 0.9261 & 1.0232 & 1.1669 \\ 1.2702 & 2.4846 & 2.5375 & 3.3476 & 3.6066 \end{array}$$

Both the random walk sampler (with $\sigma = 0.3$) and the independence sampler (with $\tau = 0.5$) were run for 10000 iterations, starting from the sample median. For the random walk sampler, 66.7% of the proposed values were accepted and for the independence sampler, around 52% of the proposals were accepted. The samplers took a few seconds to run in each case.

Kernel density estimates of the posterior density of θ given a uniform prior are given in the following diagram.



The estimated density function is approximately the same in each case.

Block Metropolis Hastings

When the dimension of X is large, then it can often be difficult to find a reasonable proposal density. In this case, it is sensible to divide X into blocks, say $X = (X_1, \ldots, X_k)$ and construct a chain with these smaller blocks.

Suppose initially that $X = (X_1, X_2)$ and define two proposal densities $q_1(y_1|x_1, x_2)$, $q_2(y_2|x_1, x_2)$ to generate candidate values for each component.

Then, define the acceptance probabilities

$$\begin{aligned} \alpha_1(x_1, y_1|x_2) &= \min\left\{1, \frac{\pi(y_1|x_2)q_1(x_1|y_1, x_2)}{\pi(x_1|x_2)q_1(y_1|x_1, x_2)}\right\} \\ \alpha_2(x_2, y_2|x_1) &= \min\left\{1, \frac{\pi(y_2|x_1)q_2(x_2|x_1, y_2)}{\pi(x_2|x_1)q_2(y_2|x_1, x_2)}\right\} \end{aligned}$$

where the densities $\pi(x_1|x_2)$ and $\pi(x_2|x_1)$ are the conditional densities and $\pi(x_1|x_2) \propto \pi(x_1, x_2)$.

The algorithm now proceeds by successively sampling from each block in turn.

The slice sampler

The slice sampler (Neal 2003) is an attractive approach when the state space is relatively low dimensional. The general algorithm for sampling from π is

1. `Given a current value,` X_t, `simulate` $U_{t+1} \sim \mathcal{U}[0, \pi(X_t)]$.

2. `Simulate` $X_{t+1} \sim \mathcal{U}[A_{t+1}]$, `where` $A_{t+1} = \{x : \pi(x) \geq U_{t+1}\}$.

It is clearly unimportant whether the constant of integration is known or not.

Example

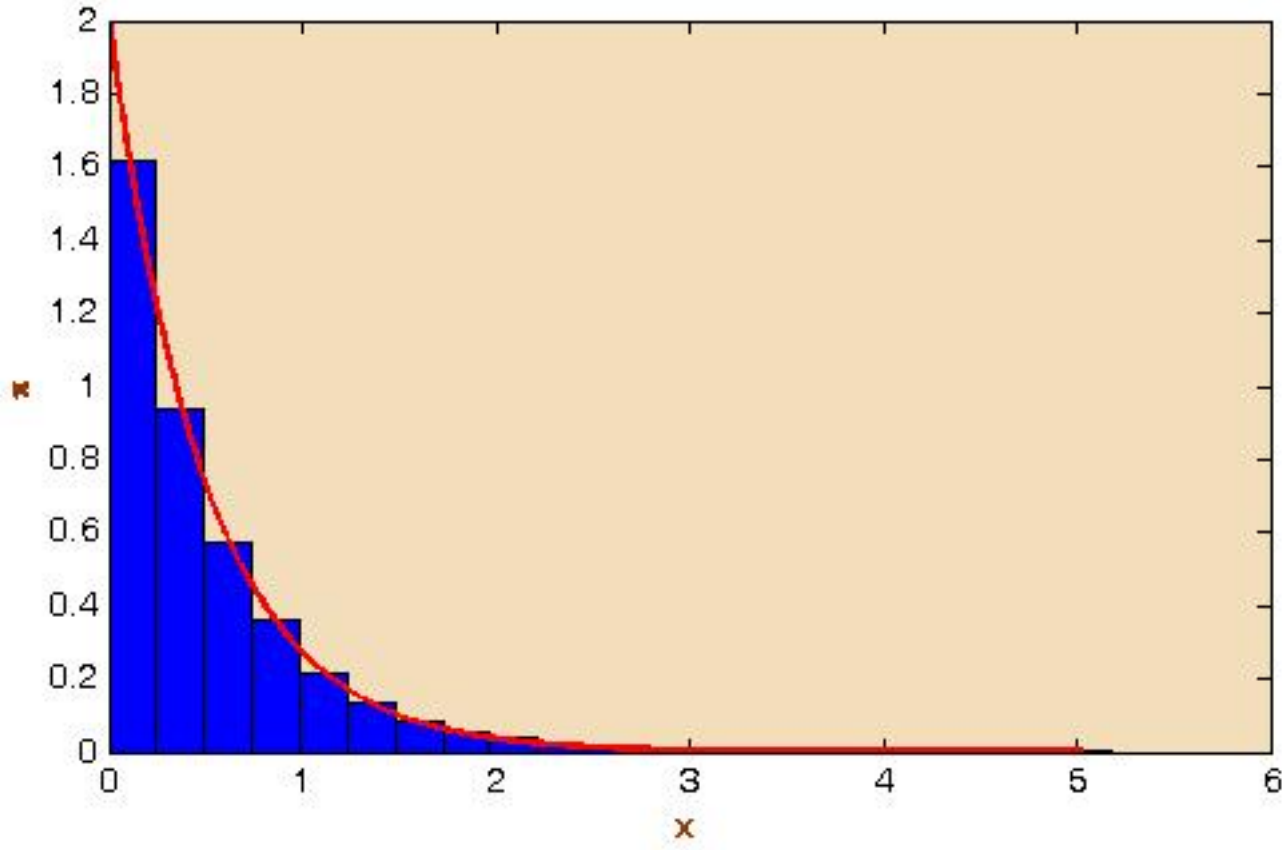
Example 51

Suppose that we wish to sample from an exponential density $X \sim \mathcal{E}(\lambda)$. Then, we know that $\pi(x) \propto e^{-\lambda x}$ and a slice sampler could proceed as follows.

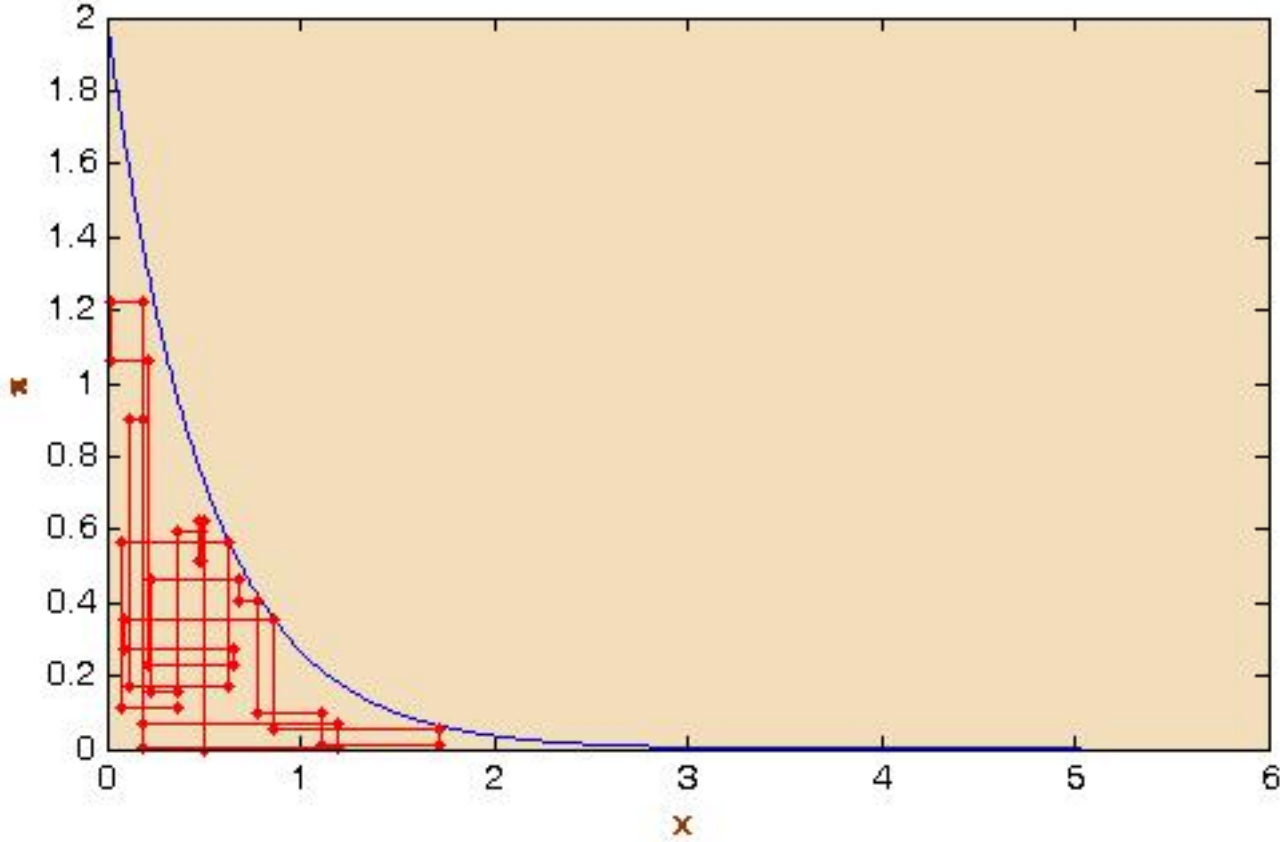
1. `Given` X_t, `generate` $U_{t+1} \sim \mathcal{U}\left[0, e^{-\lambda x_t}\right]$.

2. `Generate` $X_{t+1} \sim \mathcal{U}\left[0, -\frac{1}{\lambda}\log U_{t+1}\right]$.

The following diagram illustrates the results of 10000 iterations when $\lambda = 2$.



The final diagram illustrates how the chain moves.


2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
y
0
1
2
3
4
5
6
x

Gibbs sampling

We have seen Gibbs sampling previously in chapter 3. If we assume that $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_k)$ has joint distribution π and that the conditional distributions $\pi_i(\mathbf{X}_i|\mathbf{X}_{-i})$ are all available, where $\mathbf{X}_{-1} = (\mathbf{x}_1, \ldots, \mathbf{X}_{i-1}, \mathbf{X}_{i+1}, \ldots, \mathbf{X}_k)$ then the Gibbs sampler generates an (approximate) sample from π by successively sampling from these conditional densities. Thus, assuming that the current values are $\mathbf{x}_t$, then the algorithm is the following

1. Generate $\mathbf{x}_{1,t+1} \sim \pi_1(\cdot|\mathbf{x}_{-1,t})$.

2. Generate $\mathbf{x}_{2,t+1} \sim \pi_2(\cdot|x_{1,t+1}, x_{3,t}, \ldots, x_{k,t})$.

3. $\vdots$

4. Generate $\mathbf{x}_{k,t} \sim \pi_k(\cdot|\mathbf{x}_{-k,t+1})$

We can note that Gibbs sampling is a particular version of block Metropolis Hastings algorithm where the proposal distribution for $\mathbf{X}_i$ is exactly the conditional distribution $\pi_i(\mathbf{X}_i|\mathbf{X}_{-1})$ so that the acceptance probability is always equal to 1.

Gibbs sampling can be applied in a remarkably large number of problems.

Example 52

Suppose that the lifetime (in hours) of a machine, Y, has normal distribution so that $Y|X = x \sim \mathcal{N}(x, 1)$ and that we observe n machines during α hours. If, at the end of this time, n_1 machines have failed, with failure times $y_1, \ldots, y_{n_1}$ and $n_2 = n - n_1$ machines are still working, then the likelihood function is

$$l(x|\mathbf{y}) \propto \exp\left(-\frac{n_1}{2}(x - \bar{y}_1)^2\right)\left(1 - \Phi\left(\alpha - x\right)\right)^{n_2}$$

where $\bar{y}_1 = \frac{1}{n_1}\sum_{i=1}^{n_1} y_i$. Thus, an explicit form for the posterior of x (supposing a uniform prior) is unavailable.

However, suppose that we knew the true values of the latent variables, $Y_{n_1+1} = y_{n_1+1}, \ldots, Y_n = y_n$. Then it is clear that $X|\mathbf{y} \sim \mathcal{N}\left(\bar{y}, \frac{1}{n}\right)$ where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$.

Also, for $i = n_1 + 1, \ldots, n$, we have $Y_i | X = x, Y_i > \alpha \sim \mathcal{TN}(x, 1)$ truncated onto the region $Y_i > \alpha$. Therefore, we can set up a simple Gibbs sampling algorithm to estimate the posterior density of x as follows.

1. Set $t = 0$ and fix an initial value x_0.

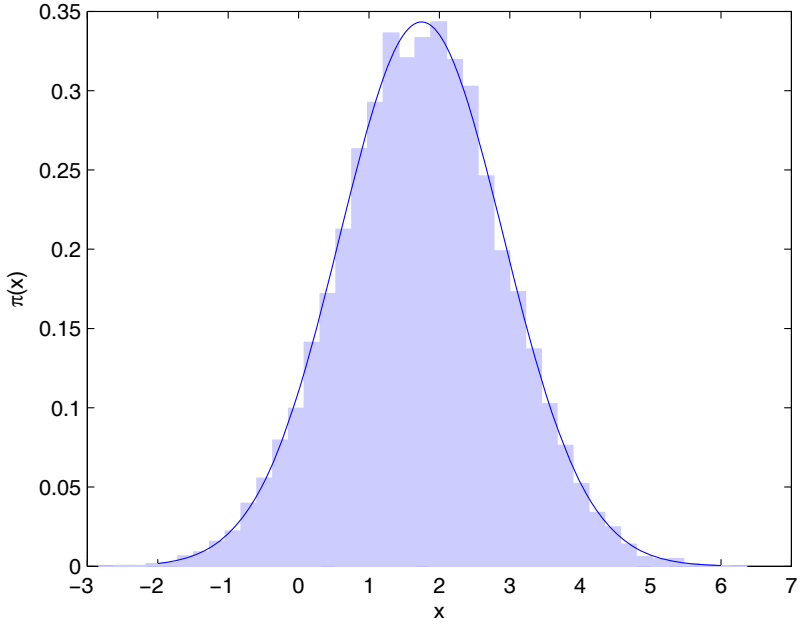
2. For $i = n_1 + 1, \ldots, n$, generate $y_{i,t} \sim \mathcal{TN}(x_t, 1)$.

3. Calculate $\bar{y}_t = \frac{1}{n}\left(\sum_{i=1}^{n_1} y_i + \sum_{i=n_1+1}^{n} y_{i,t}\right)$.

4. Generate $x_{t+1} \sim \mathcal{N}\left(\bar{y}_t, \frac{1}{n}\right)$.

5. $t = t + 1$. Go to 2.

Wiper (2007) considers a sample of 20 normally distributed lifetimes with mean μ and standard deviation 5, $(Y|X = x \sim \mathcal{N}(x, 25))$, where 14 data less than 5 are observed completely and have mean 0.94 and the remaining 6 data are truncated so that it is only known that they take values greater than 5 and a uniform prior distribution for x is assumed. The following diagram shows a histogram of the values of x generated from 10000 iterations of the Gibbs algorithm along with a fitted density.



Universal implementation of Gibbs sampling algorithms

In many cases, the conditional distributions used in Gibbs samplers are log-concave. In these cases, universal Gibbs samplers can be set up by using the ARS to sample these conditional distributions.

For non log-concave distributions, the adaptive rejection Metropolis sampler (ARMS) was introduced in Gilks et al (1995).

These algorithms form the basis of the `Winbugs` program.

A disadvantage of such universal algorithms is however that they are often inefficient. It is generally preferable to implement specific algorithms tailored to the particular problem.

MCMC convergence assessment

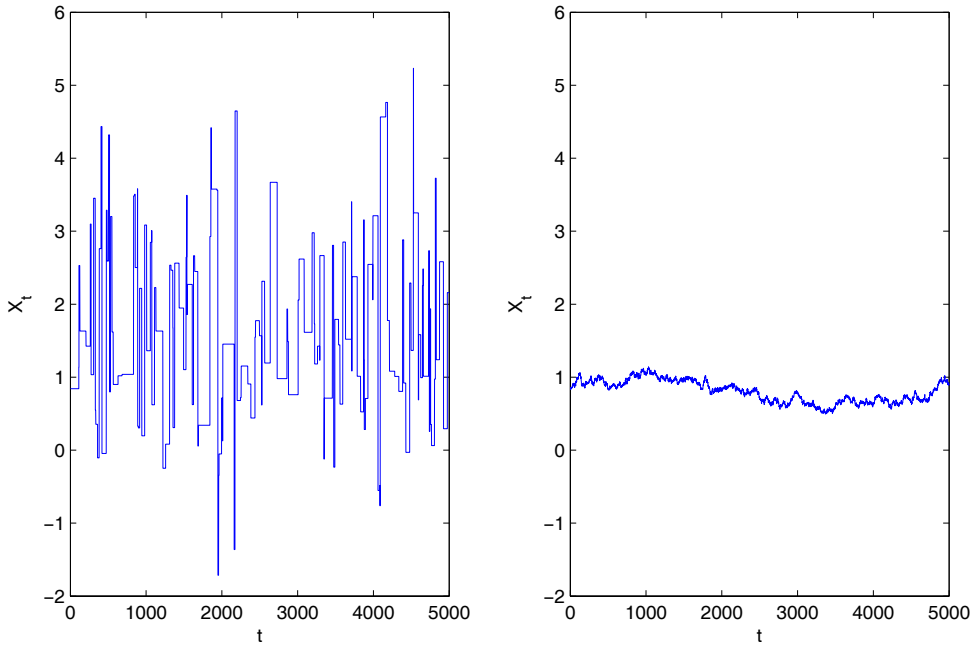
When running an MCMC algorithm, it is important to assess when the sampled values X_t have approximately converged to the stationary distribution π. This will depend on how well the MCMC algorithm is able to explore the state space also on the correlation between the X_t's.

Secondly, we need to assess the convergence of MCMC averages, e.g. $\frac{1}{T}\sum_{t=1}^{T} X_t \rightarrow E[X_t]$ and finally we need to be able to assess how close a given sample is to being independent and identically distributed.

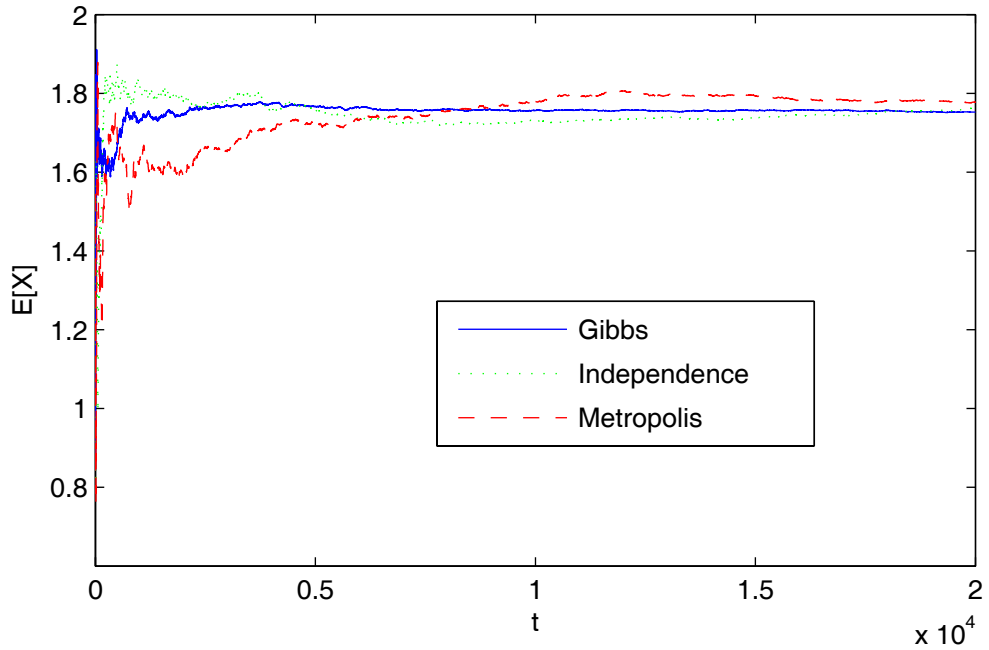
One possibility is to consider running the chain various times with different, disperse starting values. Then, we could assess the convergence of the chain by examining when sample means of the functions of interest generated from each run have converged. Other, formal diagnostics are given in Gelman and Rubin (1992).

The alternative is to use a single run of the chain.

In this case, we can produce graphs of X_t against t to show the mixing of the chain and any deviations from stationarity. The following diagram from Wiper (2007) shows examples of badly mixing chains.

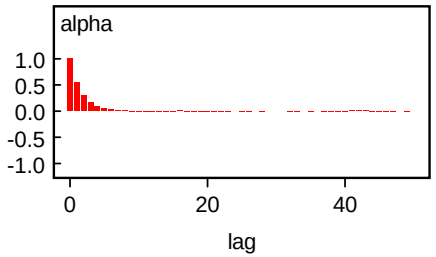


Secondly, we can plot running means of the parameters of interest to see when they have converged. The following diagram shows the estimates of $E[X|\mathbf{y}]$ used from running 3 different algorithms for the problem of Example 52.



It can be seen that the means appear to have converged after about 10000 iterations. Thus, one possibility is to run the sampler for longer, using these initial iterations as a $burn\ in$ period.

thirdly, we can plot the autocorrelation functions of the generated values. In general, as we are generating from a Markov chain, the successive values, X_t, will be positively correlated. Thus, if we wish to estimate, for example, the variance of X, then we must take this correlation into account. The following diagram shows the ACF of the parameter α in the pump failure problem analyzed in Chapter 10.



The autocorrelation has disappeared after about lag 5. One possibility is thus to thin the sample, choosing just every 5th datum which are now, approximately independent.

Other algorithms

Reversible jump

This approach (Green 1995) is basically a Metropolis Hasting sampler, which allows the chain to move over a variably dimensioned model space.

Perfect sampling

This method, developed by Propp and Wilson (1996), uses the idea of coupling from the past in order to generate an exact MCMC sample from π, avoiding the need for converegence diagnostics. See e.g.

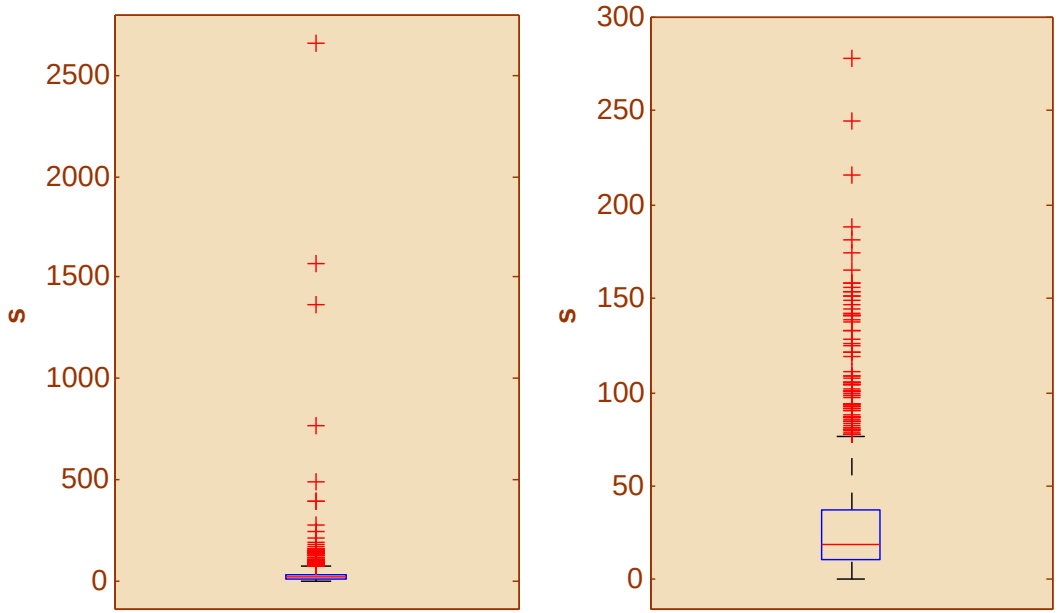
http://dbwilson.com/exact/

Particle filtering

This is an alternative approach to MCMC based on importance sampling and particularly suitable for sequential inference problems. See e.g. Doucet et al (2000) or

http://en.wikipedia.org/wiki/Particle_filter

Application III: Bayesian inference for the $dPlN$ distribution and the $M/G/c/c$ loss system



The boxplots show the times spent (lhs) and times spent below 300 days (rhs) of patients in a geriatric ward of a hospital.

These data have been analyzed previously using Coxian, phase type distributions by e.g. Ausín et al (2003). However, the data are long tailed and therefore, a heavy tailed model should be more appropriate.

Typically, long tailed data are modeled using a Pareto distribution, or a mixture of Pareto distributions (Ramírez et al 2008a) but although such a model can capture the tail behaviour, it does not capture the body of the distribution.

The double Pareto lognormal ($dPlN$) distribution has been recently introduced as a model for heavy tailed data by Reid and Jorgersen (2004).

The skewed Laplace and double Pareto lognormal distributions

It is easiest to define the $dPlN$ distribution by starting from the skewed Laplace distribution.

Definition 16

A random variable, Y is said to have a skewed Laplace distribution with parameters $\mu, \sigma, \alpha, \beta$, that is $Y \sim \mathcal{SL}(\mu, \sigma, \alpha, \beta)$, if

$$f_Y(y) = \frac{\alpha\beta}{\alpha+\beta}\phi\left(\frac{y-\mu}{\sigma}\right)\left[R(\alpha\sigma - (y-\mu)/\sigma) + R(\beta\sigma + (y-\mu)/\sigma)\right]$$

for $y \geq 0$, where $R(y)$ is Mills' ratio, that is

$$R(y) = \frac{1-\Phi(y)}{\phi(y)}.$$

If $Y \sim \mathcal{SL}(\mu, \sigma, \alpha, \beta)$ is a skewed Laplace random variable, then we can write

$$Y = Z + W$$

where $Z \sim \mathcal{N}\left(\mu, \sigma^2\right)$, $W = W_1 - W_2$ and $W_1 \sim \mathcal{E}(\alpha)$ and $W_2 \sim \mathcal{E}(\beta)$.

The conditional distributions of $Z|Y = y$ and $W_1|Y = y, Z = z$ are:

$$\begin{aligned} f_Z(z|y) &= p\frac{\frac{1}{\sigma}\phi\left(\frac{z-(\mu-\sigma^2\beta)}{\sigma}\right)}{\Phi^c\left(\frac{y-(\mu-\sigma^2\beta)}{\sigma}\right)}I_{z\geq y} + (1-p)\frac{\frac{1}{\sigma}\phi\left(\frac{z-(\mu+\sigma^2\alpha)}{\sigma}\right)}{\Phi^c\left(\frac{y-(\mu+\sigma^2\alpha)}{\sigma}\right)}I_{z<y} \quad \text{where} \\ p &= \frac{R(\beta\sigma + (y-\mu)/\sigma)}{R(\alpha\sigma - (y-\mu)/\sigma) + R(\beta\sigma + (y-\mu)/\sigma)} \qquad (1) \\ f_{W_1}(w_1|w) &= \frac{(\alpha+\beta)e^{-(\alpha+\beta)e_1}}{I_{w<0} + e^{-(\alpha+\beta)w}I_{w\geq 0}} \quad \text{for } e_1 > \max\{w, 0\}. \qquad (2) \end{aligned}$$

Definition 17

Let $Y \sim \mathcal{SL}(\mu, \sigma, \alpha, \beta)$. Then the distribution of $S = \exp(Y)$ is the double Pareto lognormal distribution and in particular, the mean of S is given by

$$E[S] = \frac{\alpha\beta}{(\alpha-1)(\beta+1)} e^{\mu+\frac{\sigma^2}{2}}$$

for $\alpha > 1$.

The density of the $dPlN$ distribution can be easily derived from the skewed Laplace density formula.

Bayesian inference for the $dPlN$ distribution

Reid and Jorgesen (2004) consider classical inference for this model using the EM algorithm. Bayesian inference is examined by Ramírez et al (2008b).

Suppose that we have standard prior distributions:

$$\begin{aligned}
\mu|\sigma^2 &\sim \mathcal{N}\left(m, \frac{\sigma^2}{k}\right) \\
\frac{1}{\sigma^2} &\sim \mathcal{G}\left(\frac{a}{2}, \frac{b}{2}\right) \\
\alpha &\sim \mathcal{G}\left(c_\alpha, d_\alpha\right) \\
\alpha &\sim \mathcal{G}\left(c_\beta, d_\beta\right)
\end{aligned}$$

Now, if we observe a sample $\mathbf{x} = (x_1, \ldots, x_n)$ from the $dPlN$ distribution, we can transform the data to $\mathbf{y} = (y_1, \ldots, y_n)$ where $y_i = \log x_i$ and $Y|\mu, \sigma, \alpha, \beta \sim \mathcal{SL}(\mu, \sigma, \alpha, \beta)$.

Clearly, the integration constant of the posterior density $p(\mu, \sigma, \alpha, \beta|\mathbf{y})$ and the marginal densities, $p(\mu|\mathbf{y}), \ldots, p(\beta|\mathbf{y})$ cannot be evaluated analytically. However it is possible to set up a Gibbs sampling scheme by introducing latent variables.

For $i = 1, \ldots, n$, we can define z_i, w_i such that $y_i = z_i + w_i$ and $Z_i|y_i, \mu, \sigma^2$ is generated from the mixture of truncated normal distributions as in Equation 1. Also, we can define w_{i1}, w_{i2} where $w_i = w_{i1} + w_{i2}$ and $W_{i1}|w_i, \alpha, \beta$ has a truncated exponential distribution as in Equation 2.

Conditional on the model parameters, both these distributions are easy to sample.

Conditional on $\mathbf{z} = (z_1, \ldots, z_n)$, then inference for μ, σ^2 is conjugate so that

$$\begin{aligned} \mu|\mathbf{z}, \sigma^2 &\sim \mathcal{N}\left(\frac{km + n\bar{z}}{k+n}, \frac{\sigma^2}{k+n}\right) \\ \frac{1}{\sigma^2}|\mathbf{z} &\sim \mathcal{G}\left(\frac{a+n}{2}, \frac{b + (n-1)s_z^2 + \frac{kn}{k+n}(m-\bar{z})^2}{2}\right). \end{aligned}$$

Conditional on $\mathbf{w}_1 = (w_{11}, \ldots, w_{n1})$ then

$$\alpha|\mathbf{w}_1 \sim \mathcal{G}\left(c_\alpha + n, d_\alpha + n\bar{w_1}.\right)$$

and, conditional on $\mathbf{w}_2 = (w_{21}, \ldots, w_{2n})$, then

$$\beta|\mathbf{w}_2 \sim \mathcal{G}\left(c_\beta + n, d_\beta + n\bar{w_2}.\right).$$

Thus, we can set up a Gibbs sampling algorithm:

Gibbs sampler

1. $t = 0$. Set initial values $\mu^{(0)}, \sigma^{(0)}, \alpha^0, \beta^{(0)}$.
2. For $i = 1, \ldots, n$
   (a) Generate $z_i^{(t)}$ from $f_Z(z|y_i, \mu^{(t-1)}, \sigma^{(t-1)}, \alpha^{t-1}, \beta^{(t-1)})$.
   (b) Set $w_i^{(t)} = y_i - z_i^{(t)}$
   (c) Generate $w_{i1}^{(t)}$ from $f_{W_1}(w_1|w_i^{(t)}, \alpha, \beta)$
   (d) Set $w_{2i}^{(t)} = w_i^{(t)} + w_{1i}^{(t)}$
3. Generate $\mu^{(t)}|\sigma^{(t-1)}, \mathbf{z}^{(t)}$ from $f(\mu|\sigma^{(t-1)}, \mathbf{z}^{(t)})$.
4. Generate $\sigma^{(t)}$ from $f(\sigma|\mathbf{z}^{(t)})$.
5. Generate $\alpha^{(t)}$ from $f(\alpha|\mathbf{w}_1^{(t)})$.
6. Generate $\beta^{(t)}$ from $f(\beta|\mathbf{w}_2^{(t)})$.
7. $t = t + 1$. Go to 2.

Problems

What priors should be used?

The natural choice would be to use the standard, improper priors $p(\mu, \tau) \propto \frac{1}{\tau}$, where $\tau = 1/\sigma^2$, $p(\alpha) \propto \frac{1}{\alpha}$ and $p(\beta) \propto \frac{1}{\beta}$. However, in this case, it is easy to show that the posterior distribution is improper, see e.g. Ramírez et al (2008b). In practice we use small but proper values of all parameters.

What initial values should we use?

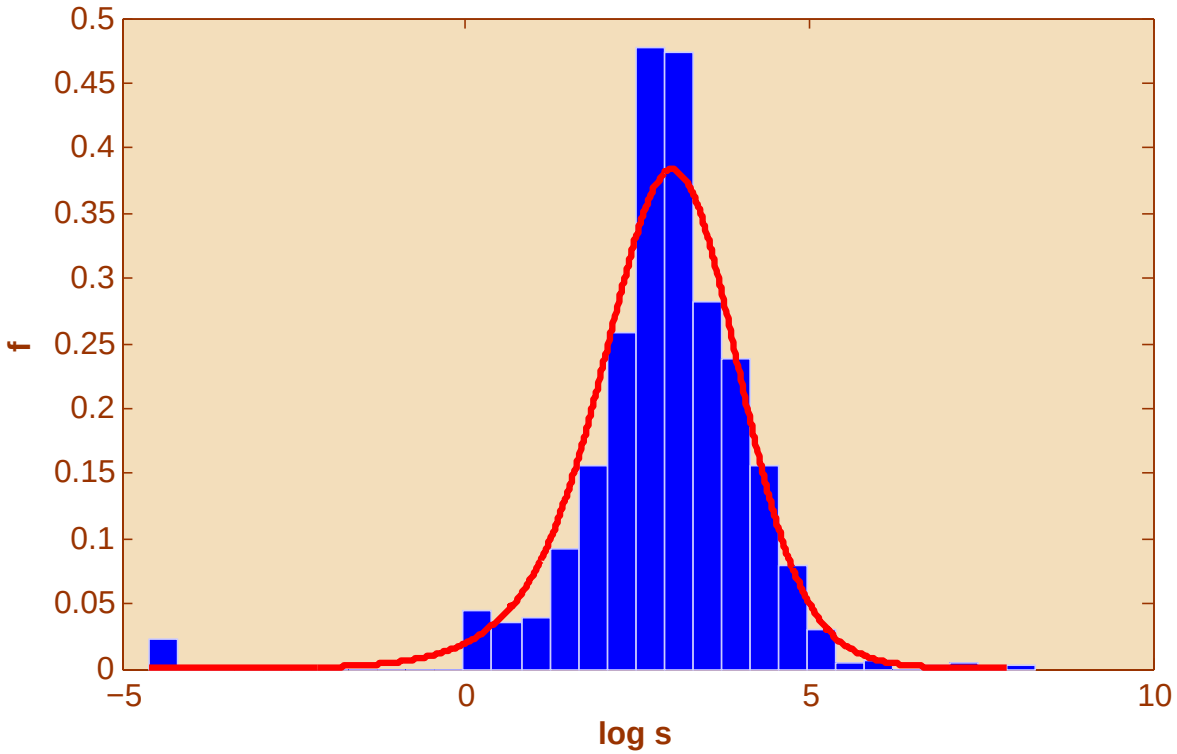
A reasonable choice is to use the maximum likelihood estimates (assuming these exist).

High autocorrelation

We are generating a lot of latent variables here. This leads to high autocorrelation. It is useful to thin the sampled data. We take every hundredth value generated.

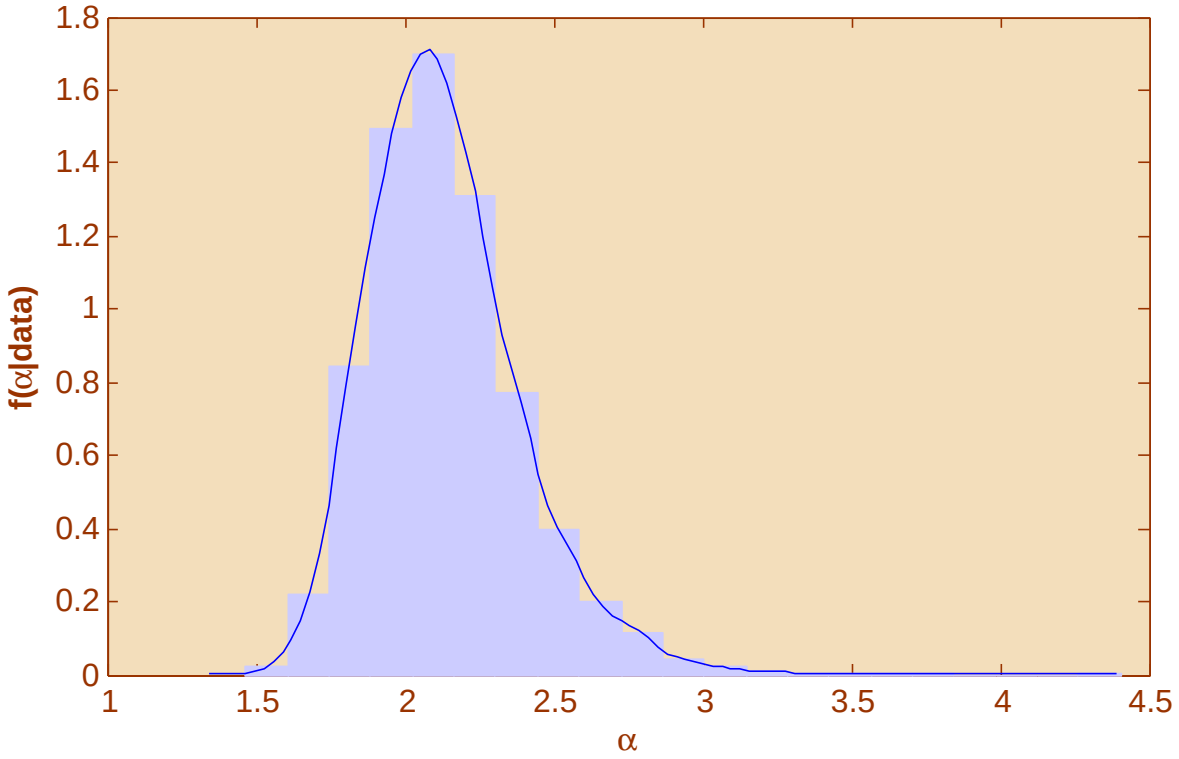
Fitted histogram for the hospital data

The diagram shows the predictive distribution of the logged hospital occupancy times. The fit seems fairly reasonable.



Are the data long tailed?

Recall that the mean of the $dPlN$ distributions only exists if $\alpha > 1$.



The probability that $\alpha < 1$ is virtually zero.

Characteristics of the hospital queueing system

We assume that the hospital has a finite number of beds, c. Patients arrive at the hospital according to a Poisson process with rate λ and are given a bed if one is available and are otherwise lost to the system.

The number of patients in the hospital system can be modeled as a $M/G/c$ Erlang loss system, that is a $M/G/c/c$ system with no queueing, see e.g. Jagerman (1974).

For an Erlang loss system, then the offered load, θ, is defined to be the expected number of arrivals in a service time, that is

$$\theta = \lambda E[S|\mu, \sigma, \alpha, \beta].$$

The equilibrium distribution of the number of occupied beds is given by

$$P(N_b = n|\theta) = \frac{\theta^n/n!}{\sum_{j=0}^{c} \theta^j/j!}$$

and therefore, the blocking probability or probability that an arriving patient is turned away is

$$B(c, \theta) = P(N_b = c|\theta) = \frac{\theta^c/c!}{\sum_{j=0}^{c} \theta^j/j!}$$

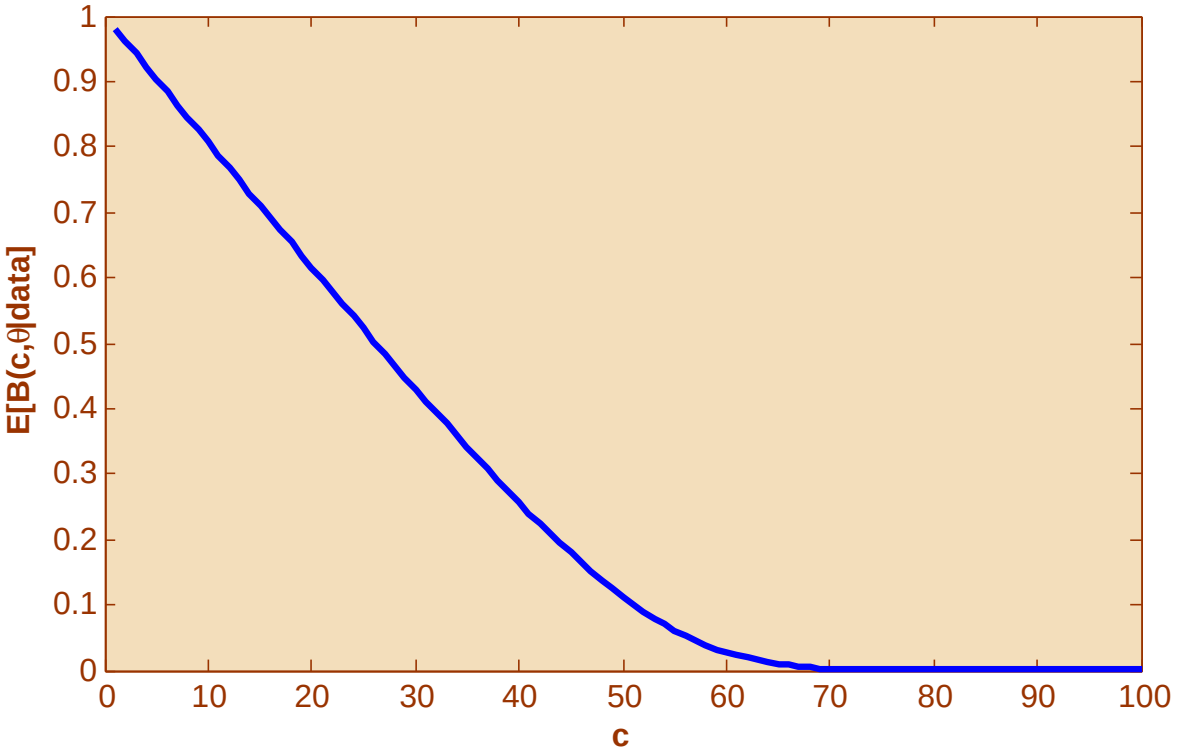
and the expected number of occupied beds is

$$E[N_b|\theta] = \theta \left(1 - B(c, \theta)\right).$$

Given λ, c and a Monte Carlo sample of service time parameters, the above quantities can be estimated by Rao Blackwellization.

Results for the hospital data

Following Ausín et al (2003), we shall suppose that the arrival rate is $\lambda = 1.5$. The diagram shows the blocking probabilities for different numbers of beds.



Optimizing the number of beds

Assume that the hospital accrues different costs for the total numbers of occupied and unoccupied beds and the number of patients that are turned away. The hospital gains profits for those patients treated. Assume then that we have c beds when we shall suppose:

- Cost per occupied bed per time unit is r_b so that the expected cost per time unit due to occupation of beds is $r_b E[N_b|\theta]$.

- Cost r_e per time unit for every empty bed so the expected cost per time unit due to empty beds is $r_e(c - E[N_b|\theta])$.

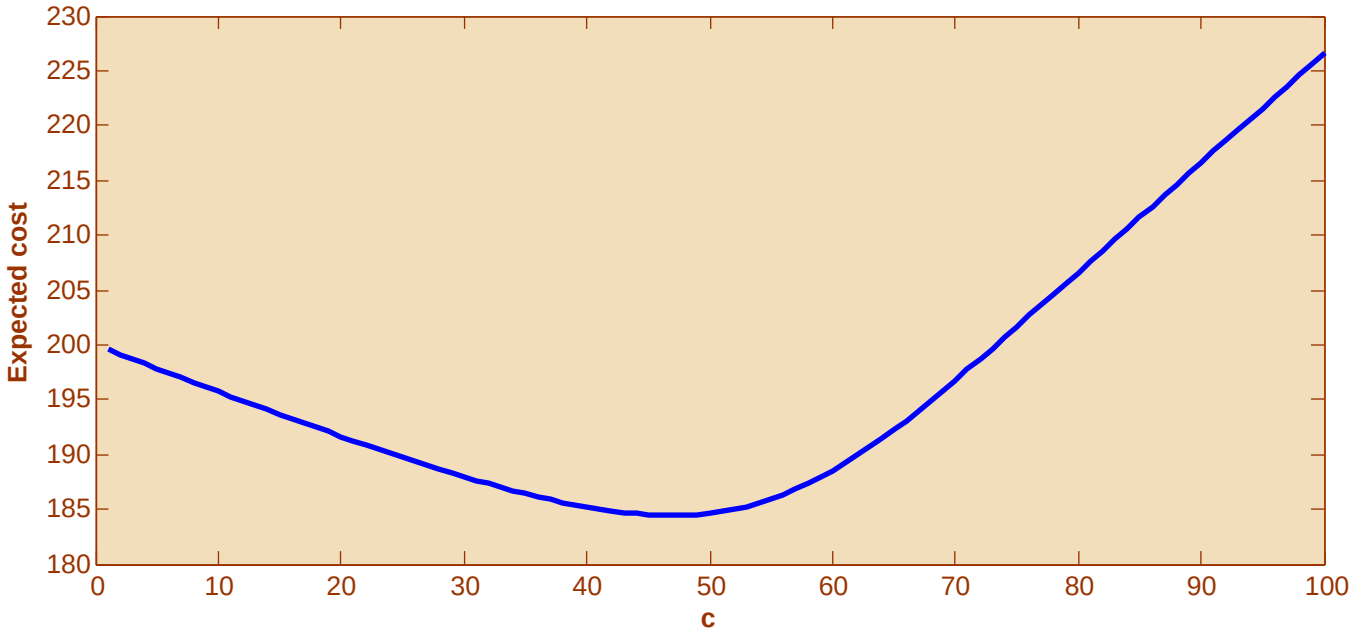
- Cost per patient turned away per time unit is r_l when the expected cost per time unit is $r_l B(c, \theta)$.

This leads to an expected loss per time unit

$$\begin{aligned} L(c|\lambda,\theta) &= r_b E[N_b|\theta] + r_e(c - E[N_b|\theta]) + r_l B(c,\theta) \\ &= (r_b - r_e)\theta + r_e c + \{(r_e - r_b)\theta + r_l \lambda B(c,\theta)\} \end{aligned}$$

Following Ausín et al (2003), we shall assume that $r_e = 1$, $r_l = 200$ and here we suppose that $r_b = 3$.

Then, we can calculate the number of beds which minimize the expected loss. This is the optimal number of beds from a Bayesian viewpoint.



The optimal number of beds given this loss function is 47 The results are slightly different to those in Ausín et al (2003) who found an optimal number of $c = 58$ with a similar loss function and an alternative service model.

Application IV: Weibull mixture models for heterogeneous survival data

The Weibull distribution is one of the most popular parametric models for survival and reliability. The density function and survival function of a Weibull distributed variable, $X \sim \mathcal{W}(\theta, a)$, are:

$$\begin{aligned} f_W(x|\theta, a) &= \theta a x^{a-1} e^{-\theta x^a} \\ \bar{F}(x|\theta, a) &= e^{-\theta x^a} \end{aligned}$$

The likelihood function

Consider two possible cases:

i We observe n complete lifetimes $\mathbf{x} = x_1, \ldots, x_n$

ii We observe n complete lifetimes as earlier and m truncated lifetimes $x_{m+1}, \ldots, x_{m+n}$ where it is supposed that the subjects are still living at these times.

In case `i.` the likelihood function is

$$l(\theta, a|\mathbf{x}) \propto \theta^n a^n \prod_{i=1}^{n} x_i^a e^{-\theta \sum_{i=1}^{n} x_i^a}$$

and in case `ii`, we have

$$l(\theta, a|\mathbf{x}) \propto \theta^n a^n \prod_{i=1}^{n} x_i^a e^{-\theta \sum_{i=1}^{m+n} x_i^a}$$

Prior and posterior distributions

Suppose that we set the following prior distributions

$$\begin{aligned} \theta &\sim \mathcal{G}(\alpha_\theta, \beta_\theta) \\ a &\sim \mathcal{G}(\alpha_a, \beta_a) \end{aligned}$$

then the conditional posterior distributions are

		Case 1	Case 2
$\theta \vert a, \mathbf{x}$	$\sim$	$\mathcal{G}\left(\alpha_\theta + n, \beta_\theta + \sum_{i=1}^{n} x_i^a\right)$	$\mathcal{G}\left(\alpha_\theta + n, \beta_\theta + \sum_{i=1}^{m+n} x_i^a\right)$
$f(a \vert \theta, \mathbf{x})$	$\propto$	$a^{\alpha_a+n} \prod_{i=1}^{n} x_i^a e^{-\left(\beta_a a + \theta \sum_{i=1}^{n} x_i^a\right)}$	$a^{\alpha_a+n} \prod_{i=1}^{n} x_i^a e^{-\left(\beta_a a + \theta \sum_{i=1}^{m+n} x_i^a\right)}$

Gibbs Sampling

It is easy to set up a Gibbs sampler as follows:

1) $t = 0$. Set initial value $a^{(0)}$.

2) Sample $\theta^{(t+1)}$ from $f\left(\theta|\mathbf{x}, a^{(t)}\right)$

3) Sample $a^{(t+1)}$ from $f\left(a|\mathbf{x}, \theta^{(t+1)}\right)$.

4) $t = t + 1$ Go to 2.

Clearly, step 2 is straightforward. For step 3, Tsionas (2002) uses a Metropolis Hastings step and Marín et al (2005) consider a slice sampler:

A slice sampler for sampling from $f(a|\mathbf{x}, \theta)$

2a) Simulate a uniform random variable; $u \sim \mathcal{U}\left[0, g\left(a^{(t)}|\mathbf{x}, \theta^{(t)}\right)\right]$ where g is the density formula on the previous page

2b) Simulate $a^{(t+1)}$ from a uniform distribution with support $S(u) = \{a : g(a) \geq u\}$.

In practice, the only difficulty with this algorithm is in evaluating the support $S(u)$, although as indicated in Neal (2003), this is straightforward to do by simply sampling from a uniform distribution over a slightly larger space and then checking that the constraint in 2b) is verified.

Mixtures of Weibull distributions

When the data are heterogeneous, it is more appropriate to consider a mixture model

$$f(x|k, \mathbf{w}, \boldsymbol{\theta}, \mathbf{a}) = \sum_{j=1}^{k} w_j f_W(x|\theta_j, a_j).$$

In this case, a natural prior for the weights is $\mathbf{w} \sim \mathcal{D}(\underbrace{c, \ldots, c}_{k})$ and we can use gamma priors for the remaining parameters as earlier.

However, given sample data, e.g. of type 1, then the likelihood becomes

$$l(\mathbf{w}, \boldsymbol{\theta}, \mathbf{a}|\mathbf{x}) \propto \prod_{i=1}^{n} \sum_{j=1}^{k} w_j f_W(x_i|\theta_j, a_j)$$

which contains k^n terms and for n relatively large, is intractable.

Simplifying the likelihood with latent variables

Let Z be a random variable such that $P(Z = z|k, \mathbf{w}) = w_z$. Then, if X comes from the mixture of Weibulls model, we can write

$$X|Z = z \sim \mathcal{W}(\theta_j, a_j).$$

Suppose now that for each observed datum, we know the values $\mathbf{z} = z_1, \ldots, z_n$. Then, the likelihood function simplifies to

$$l(\mathbf{w}, \boldsymbol{\theta}, \mathbf{a}, \mathbf{z}|\mathbf{x}) \propto \prod_{i=1}^{n} f_W(x_i|\theta_{z_i}, a_{z_i}).$$

Posterior distributions

It is now easy to show that

$$\begin{aligned} P(Z_i = z|\mathbf{x}, \mathbf{w}, \boldsymbol{\theta}, \mathbf{a}) &\propto w_z f_W(x_i|\theta_z, a_z) \\ \mathbf{w}|\mathbf{x}, \boldsymbol{\theta}, \mathbf{a}, \mathbf{z} &\sim \mathcal{D}(c + n_1, c + n_2, \ldots, c + n_z) \end{aligned}$$

where n_j are the number of data allocated to element j of the mixture.

The conditional posterior distributions for each θ_j and a_j are as earlier but only considering the sample data assigned to element j of the mixture.

Gibbs sampler

1) $t = 0$. Set initial values $\mathbf{w}^{(0)}, \boldsymbol{\theta}^{(0)}, \mathbf{a}^{(0)}$.

2) For $i = 1, \ldots, n$ sample $z_i^{(t+1)}$ from $P\left(z_i | \mathbf{x}, \mathbf{w}^{(t)}, \boldsymbol{\theta}^{(t)}, \mathbf{a}^{(t)}\right)$.

3) Sample $\mathbf{w}^{(t+1)}$ from $f\left(\mathbf{w} | \mathbf{x}, \mathbf{z}^{(t+1)}, \boldsymbol{\theta}^{(t)}, \mathbf{a}^{(t)}\right)$.

4) For $j = 1, \ldots, k$, sample $\theta_j^{(t+1)}$ from $f\left(\theta_j | \mathbf{x}, \mathbf{z}^{(t+1)}, \mathbf{w}^{(t+1)}, \mathbf{a}^{(t)}\right)$

5) For $j = 1, \ldots, k$, sample $a_j^{(t+1)}$ from $f\left(a | \mathbf{x}, \mathbf{z}^{(t+1)}, \mathbf{w}^{(t+1)}, \boldsymbol{\theta}^{(t+1)}\right)$.

6) $t = t + 1$ Go to 2.

Inference when k is unknown

In this case, we define a prior distribution for k, e.g. a truncated Poisson. Now, the previous algorithm can be thought of as giving inference for the model parameters conditional on k. We need to incorporate a step which allows for the possibility of changing k and these parameters. There are two possibilities: reversible jump (Richardson and Green 1997) or birth death MCMC (Stephens 2000). Here we use a birth death MCMC approach.

Simulated example

Sample of size 150 with 10% censoring simulated from a mixture of 3 Weibull distributions with weights $\mathbf{w} = (0.6, 0.3, 0.1)$ and parameters $\boldsymbol{\theta} = (0.1, 0.3, 0.5)$ and $\mathbf{a} = (0.5, 1, 2)$.

MCMC algorithm ran for 60000 iterations with 10000 to burn in.

The first graphs illustrate the convergence of the algorithm and the following graph shows the posterior distribution of k. The final graph shows a Kaplan Meier estimate of the survival curve as well as the fitted and true curves

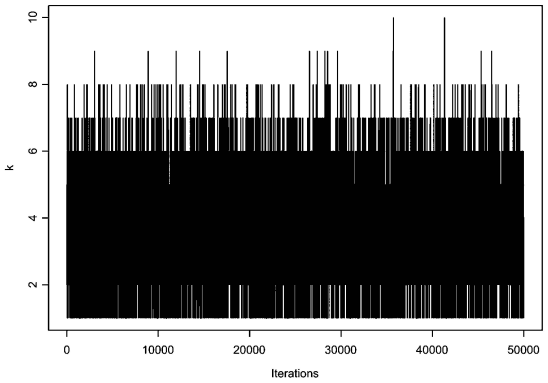


Figure 1. Plot of mixture size k versus iteration of the MCMC algorithm.

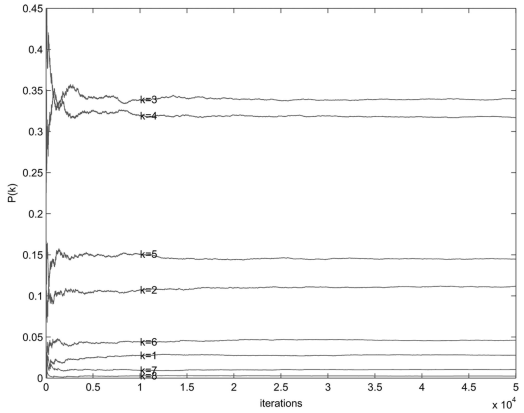


Figure 2. Plot of estimated $P(k \mid \text{data})$ versus iteration of the MCMC algorithm.

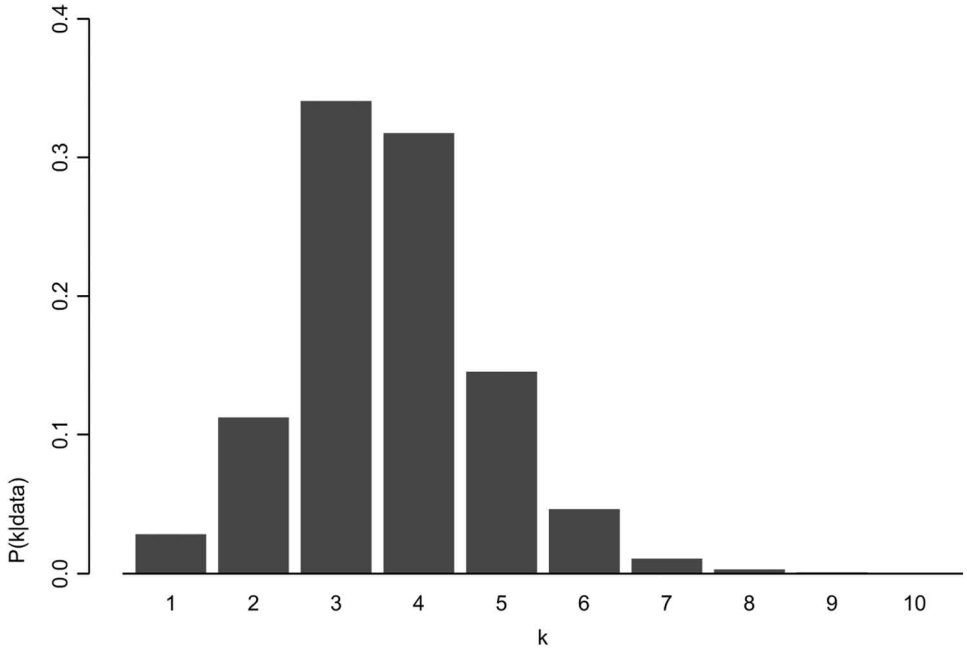


Figure 3. Posterior distribution of the mixture size k.

There is a high posterior probability of 3 components.

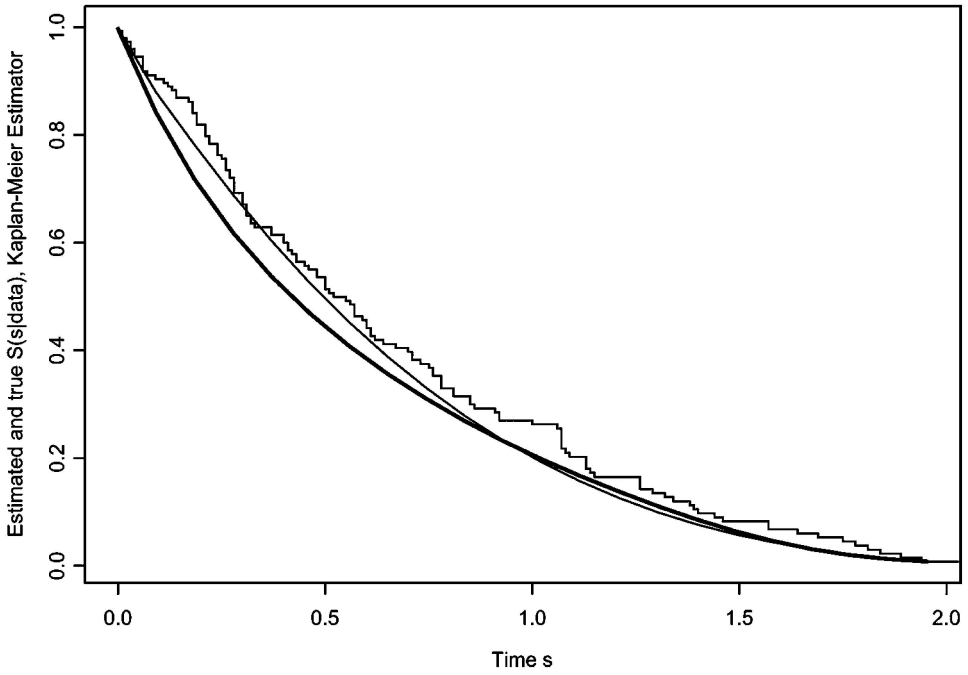


Figure 4. Expected survivor function (thin black line), true survivor function (thick black line) and Kaplan–Meier estimate (polygonal line).

The survival curve is better estimated by the predicted curve than by the KM estimate.

Real data example

Here we analyze data from 87 persons with lupus nephritis, see Abrahamowicz et al. (1996). These patients were studied over a 15-year time period, during which 35 deaths were recorded. In the original article, covariate information was used to study the effects of disease duration prior to diagnosis on the risk of mortality of patients, via a time dependent hazard rate model and suggest that the usual proportional hazards model fits the data reasonably well.

Here, we do not use the covariate information and use the Weibull mixture model to represent the possible inhomogeneity of the lifetime data.

The first graph shows a KM estimate and the fitted survival curve and the second graph shows the estimated hazard curve.

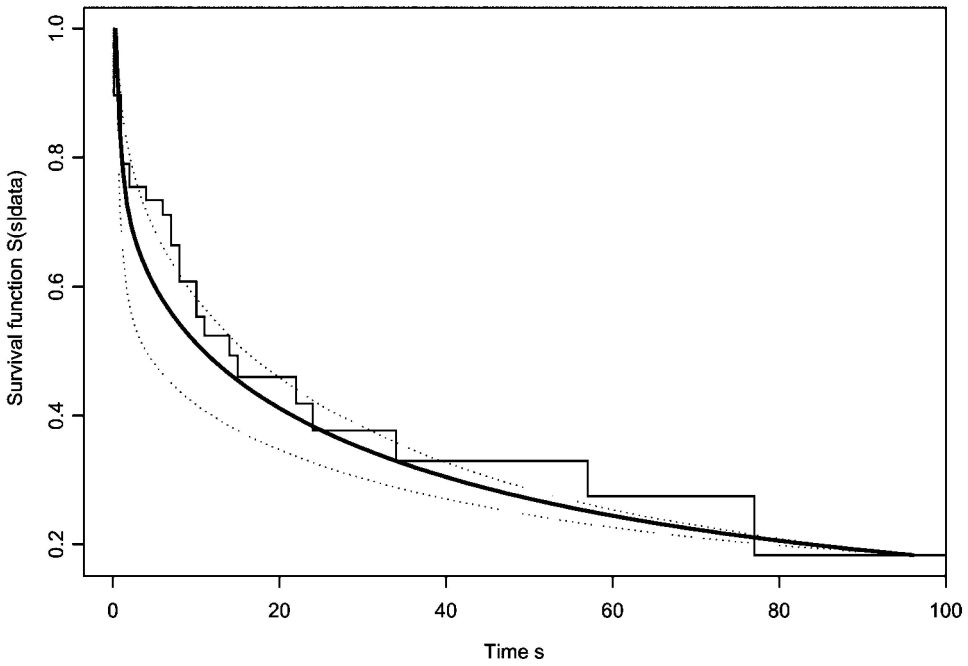


Figure 5. Fitted survival curve and Kaplan–Meier estimator for the Lupus data.

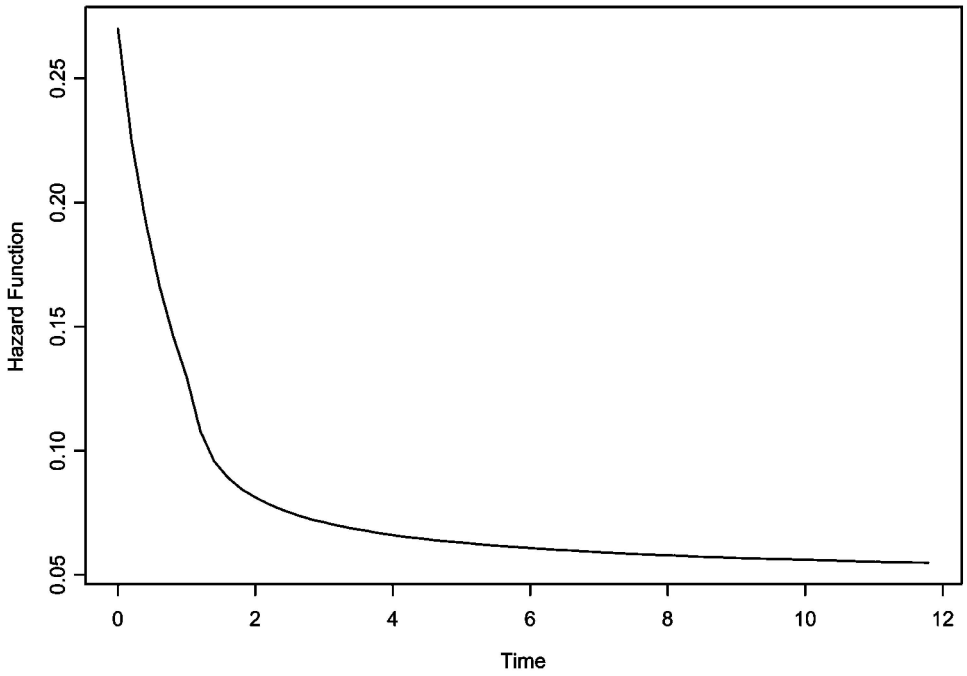


Figure 6. Expected hazard function for the Lupus data.

Finally, in Fig. 6, we illustrate the expected population hazard function for this data set. The expected hazard falls quite rapidly towards 0.05 but then decays very slowly.

These results would seem to suggest that patient death is more probable in the early stages of treatment, but that if a patient survives this phase, then they have a reasonable chance of longer term survival.

References

Abrahamowicz, M., Mackenzie, T., Esdaile, J. (1996). Time-dependent hazard ratio: modeling and hypothesis testing with application in lupus nephritis. *Journal of the American Statistical Association*, **91**, 1432-1439.

Ausín, M.C. (2007). An introduction to quadrature and other numerical integration techniques. In *Encyclopedia of Statistics in Quality and Reliability*, Wiley.

Bayesian modelling of hospital bed occupancy times using a mixed generalized Erlang distribution. In Bernardo, J.M., Bayarri, M.J., Berger, J.O., Dawid, A.P., Heckerman, D., Smith, A.F.M., West, M. (eds.), *Bayesian Statistics 7*, Oxford: University Press, 443–452.

Doucet, A., Godsill, S. and Andrieu, C. (2000). On Sequential Monte Carlo Methods for Bayesian Filtering. *Statistics and Computing*, **10**, 197–208. http://www.springerlink.com/content/q6452k2x37357l3r/.

Gelman, A. and Rubin, D.B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, **7**, 457–472.

Geweke, J. (1991). Efficient simulation from the Multivariate Normal and Student t-distribution subject to linear constrains. In *Computing Sciences and Statistics (Proc. 23rd Symp. Interface)*, American Statistical Association, 571–577.

Gilks, W.R. (1992). Derivative-free adaptive rejection sampling for Gibbs sampling. In *Bayesian Statistics 4*, (eds. Bernardo, J., Berger, J., Dawid, A.P., and Smith, A.F.M.) Oxford University Press.

Gilks, W.R. and Wild, P. (1992). Adaptive rejection sampling for Gibbs sampling. *Applied Statistics*, **41**, 337–348.

Gilks, W.R., Best, N.G. and Tan, K.K.C. (1995). Adaptive rejection Metropolis sampling. *Applied Statistics*, **44**, 455–472.

Green, P.J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, **82**, 711-732.

Hastings, W.K. (1970). Monte Carlo Sampling Methods Using Markov Chains and Their Applications, *Biometrika*, **57**, 97–109.

Jagerman, D.L. (1974). Some properties of the Erlang loss function, *Bell System Technical Journal*, **53**, 525–551.

Marín, J.M., Rodríguez-Bernal, M.T. and Wiper, M.P. (2005). Using Weibull Mixture Distributions to Model Heterogeneous Survival Data. *Communications in Statistics - Simulation and Computation*, **34**, 673–684.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E. (1953). Equations of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, **21**, 1087–1092.

Neal, R. (2003). Slice sampling (with discussion). *Annals of Statistics*, **31**, 705–767.

Propp, J. and Wilson, D. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. Random Structures and Algorithms, **9**, 223-252.

Ramirez, P., Lillo, R.E. and Wiper, M.P. (2008a). Bayesian Analysis of a Queueing

System with a Long-Tailed Arrival Process. *Communications in Statistics: Simulation and Computation*, **37**, 697–712.

Ramírez, P., Lillo, R.E., Wiper, M.P. and Wilson, S.P. (2008b). Inference for double Pareto lognormal queues with applications. *Working papers in Statistics and Econometrics*, **08-02**, Universidad Carlos III de Madrid.

Reed, W.J. and Jorgensen, M. (2004). The Double Pareto-Lognormal Distribution - A New Parametric Model for Size Distributions. *Communications in Statistics - Theory and Methods*, **33**, 1733–175.

Robert, C.P. (1995). Simulation of truncated normal random variables. *Statistics and Computing*, **5**, 121–125.

Robert, C.P. and Casella, G. (2004). *Monte Carlo Statistical Methods*. Berlin: Springer.

Roberts, G., Gelman, A. and Gilks, W.R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *Annals of Applied Probability*, **7**, 110–120.

Rubin, D.B. (1987). A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: the SIR algorithm. *Journal of the American Statistical Association*, **82**, 543–546.

Tsionas, E. G. (2002). Bayesian analysis of finite mixtures of Weibull distributions. *Communications in Statistics: Theory and Methods*, **31**, 37-48.

Wiper, M.P. (2007). Introduction to Markov chain Monte Carlo simulation. In

Encyclopedia of Statistics in Quality and Reliability, Wiley, 1014-1020.