Bayesian Inference and Computational Methods 2009

Homework 5

1. *Simulation of a general finite state space Markov chain.* The simple **R** function `markov` defined below takes as its arguments the transition matrix `P` of a finite state-space Markov chain, an initial state `x0` of the chain, and a discrete time `n` up to which the chain is to be simulated. The function returns as its value a simulation (realisation) of the chain consisting of the simulated states at times $1, 2, \ldots, n$.

```
markov = function(P, x0, n)
{
# R function to simulate the first n steps
# of a Markov chain with transition matrix P and
#initial state x0
  R = t(apply(P, 1, "cumsum"))
  x = numeric(n)
  r = R[x0, ]
  for(i in 1:n) {
    u = runif(1)
    x[i] = 1 + sum(u > r)
    r = R[x[i], ]
  }
  return(x)
}
```

(a) Understand carefully the workings of the function `markov`. In particular understand the role of the matrix `R`.

(b) Use the function to simulate 100000 steps of each of the Markov chains whose transition matrices are given below. In each case use the simulation to estimate the stationary distribution of the chain, and compare it with the true value.

i. The *device state* Markov chain of the lecture notes with transition matrix

$$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ 0 & \frac{3}{4} & \frac{1}{4} \\ 1 & 0 & 0 \end{pmatrix}.$$

ii. The *taxicab driver* Markov chain of Homework 4 with transition matrix

$$P = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{3}{4} & 0 & \frac{1}{4} \\ \frac{3}{4} & \frac{1}{4} & 0 \end{pmatrix}.$$

(c) Consider the Markov chain $\{X_n,\ n \geq 0\}$ with state space $\{0, 1, \ldots, c\}$ and transition probabilities given by

$$p_{i,i+1} = p, \quad p_{i,i-1} = 1 - p,$$

for $1 \leq i \leq c - 1$, where $0 < p < 1$, and

$$p_{0,1} = p_{c,c-1} = 1$$

(with $p_{ij} = 0$ otherwise). For the case $c = 10$, $p = 1/3$, use the function `markov` to estimate the stationary distribution of this chain, and also the expected value of the random variables X_n under this stationary distribution. Compare your results with the exact values of these quantities (which may be obtained via the solution of the *detailed balance equations*).

2. *Simulation of a distribution on the integers.* Suppose that a *target distribution* takes values on the set $\mathbb{Z}$ of integers, and has probability function *proportional* to some known function π. Assume also that the set S on which π takes nonzero values consists of *consecutive* integers. We may construct a Markov chain $\{X_n\}_{n\geq 0}$—formally with the state space $\mathbb{Z}$ but which never takes values outside the set S—whose *stationary distribution* is the given *target* by using an instance of the *random-walk Metropolis algorithm* as follows.

   - The *proposal* is to jump one state up or one state down with equal probabilities 1/2, i.e. the matrix $q(\cdot,\cdot)$ of *proposal* probabilities is given by, for all x,

$$\begin{aligned} q(x, x-1) &= \frac{1}{2} \\ q(x, x+1) &= \frac{1}{2} \\ q(x, y) &= 0 \qquad \text{otherwise.} \end{aligned}$$

   - The *acceptance probabilities* $\alpha(\cdot,\cdot)$ are now given as usual by, for each x,

$$\begin{aligned} \alpha(x, x-1) &= \min\left(1, \frac{\pi(x-1)}{\pi(x)}\right) \\ \alpha(x, x+1) &= \min\left(1, \frac{\pi(x+1)}{\pi(x)}\right) \end{aligned}$$

     (and in the event of a proposal to move to $x-1$ or $x+1$ *not* being accepted, the chain remains in the same state x for the next time instant).

   This algorithm is implemented by the **R** function `rwm.int` available at
   http://www.ma.hw.ac.uk/~stan/f73bi/R/
   List the function for details on how to use it. Note in particular that the first argument of this function is the name of another **R** function to evaluate $\pi(x)$ at any integer x.

   For each of the following sets S and functions π, use this scheme to simulate a chain whose stationary distribution is proportional to π. In each case use the chain to estimate the probability that the target distribution takes the value x for each $0 \leq x \leq 10$, and also to estimate the *mean* and the *standard deviation* of the target distribution (where these exist). Compare your answers with known results where possible. [Note also that you should probably use a burn-in period of around 1000 steps of the chain (in this simple problem it would probably also be safe to have no burn-in period), followed by a simulation consisting of, say, $100,000$ further steps of the chain.]

   (a) $S = \{0, 1, \ldots, 10\}$ and $\pi(x) = 1$ for all $x \in S$;

   (b) $S = \{0, 1, \ldots, 10\}$ and $\pi(x) = 1/2^x$ for all $x \in S$;

   (c) $S = \{0, 1, 2, \ldots\}$ and $\pi(x) = 4^x/x!$ for all $x \in S$;

   (d) $S = \{1, 2, \ldots\}$ and $\pi(x) = x^{-3}$ for all $x \in S$.

3. *Random-walk Metropolis.* Recall that the Metropolis algorithm is the special case of the Metropolis-Hastings algorithm in which the *proposal* $q(\cdot, \cdot)$ satisfies $q(x, y) = q(y, x)$ for all $x, y \in S$. Suppose now that the state space S is either the set of integers or the real line. The *random-walk Metropolis algorithm* further requires that $q(x, y) = q(|y - x|)$ for some function q (on the positive integers or positive real line as appropriate). Thus at each step the proposal is to make a jump from the current state, the distribution of the size of the jump being always the same and symmetric about 0. As usual for the Metropolis algorithm, given that the proposal is to move from x to y, the acceptance probability is

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

Design and implement a random-walk Metropolis algorithm to simulate from a $\Gamma(2, 1)$ distribution using a normal distribution as the the proposal distribution. (Note that we require that the size of the proposed jump should have mean 0, or equivalently that the mean of the proposal distribution should be equal to the current state.)

   (a) By examining a time series plot of the output from the chain investigate how the mixing properties of the chain depend on the variance of the proposal distribution.

   (b) Investigate how the dynamics of your chain are affected by the choice of initial state.

Hint: you may find helpful the **R** function `rwm.norm` available at
`http://www.ma.hw.ac.uk/~stan/f73bi/R/`
As usual, list the function for details on how to use it. Note in particular that the first argument of this function is the name of another **R** function to evaluate a function proportional to the target density.

4. *Allowed and forbidden states.*

   (a) Suppose that r balls are distributed among n consecutive boxes, subject to the constraints that no box may contain more than one ball and no three consecutive boxes may each contain a ball, and in which all allowed assignments of balls to boxes are equally likely. Write a programme, for general n and $r < n$, to implement the Metropolis algorithm, in which the *proposal* is to choose at random a pair of boxes and to swap their contents. (Since the target distribution assigns the same probability to all allowed states, the proposal is accepted if and only if the resulting state is allowed.) The program should calculate an estimate of the probability $\pi(i)$ that each box i contains a ball. In the case $n = 20$, $r = 6$, investigate how this probability varies with i.

   (b) Consider also the modification to the above example in which the number of balls is not fixed and in which, again subject to the above constraints, all assignments of balls to boxes are equally likely. A simple proposal is now to choose any box at random and to change its occupancy state. Again write a programme, for general n and $r < n$, to implement the suggested algorithm and to calculate an estimate of the probability $\pi(i)$ that each box i contains a ball. In the case $n = 20$ investigate how this probability varies with i.

5. *Bayesian inference for the gamma distribution with known shape parameter.* A set of observations $y_1, y_2, \ldots, y_n$ is known to be reasonably modelled as a random sample from a $\Gamma(4, \lambda)$ distribution.

   (a) Suppose that the prior distribution for λ is taken to be Exp(0.1) and that, for a set of $n = 20$ observations, it is found that $\sum_{i=1}^{20} y_i = 64.0$. Show that the posterior distribution of λ is given by
   $$\pi(\lambda; \boldsymbol{y}) \propto \lambda^{80} e^{-64.1\,\lambda}.$$

   (b) Use a *random-walk Metropolis algorithm* with normally distributed step sizes to simulate from this posterior distribution. Use trial and error to identify a suitable standard deviation of the step-size. *Hint:* you may again find helpful the **R** function `rwm.norm` available at
   http://www.ma.hw.ac.uk/~stan/f73bi/R/

   (c) Estimate the mean and variance of the posterior distribution of λ from a suitably long run of the chain. Examine the shape of the histogram for this posterior distribution and estimate its mode. In all cases compare your results with the theoretical values—which are obtainable analytically for this particularly simple posterior distribution.

   (d) Estimate a 90% equal-tailed credible interval for λ from the output of the chain. (You can do this using the **R** function `sort` which arranges the elements of a vector in increasing order. The end points of a credible interval can then be obtained from the ordered Markov chain output. Alternatively, and more simply, you can use the **R** function `quantile`.)

   (e) How does the posterior distribution of λ change when its prior is taken instead to be Exp(1)?

6. *Bayesian inference for the truncated Poisson distribution.* In a random sample of size 30 from an assumed Poisson distribution with unknown parameter (mean) λ, 20 of the observations were as follows

```
3  4  4  4  4  5  5  6  7  7  7  8  8  8  9  9 10 10 10 10
```

   while a further 10 observations took values which were greater than 10.

   (a) Show that the likelihood function for λ, given these data, is given by
   $$L(\lambda) \propto e^{-20\lambda} \lambda^{138} \left(1 - e^{-\lambda} \sum_{i=0}^{10} \frac{\lambda^i}{i!}\right)^{10}.$$

   (b) Suppose that the prior distribution for λ is taken to be Exp(0.1). Find the posterior distribution of λ.

   (c) Use a *random-walk Metropolis algorithm* with normally distributed step sizes to simulate from this posterior distribution. Use trial and error to identify a suitable standard deviation of the step-size.

   (d) Estimate the posterior mean and variance of λ from a suitably long run of the chain. Examine the shape of the histogram for λ and estimate its posterior mode. Estimate also a 90% equal-tailed credible interval for λ from the output of the chain.

   (e) In fact the data were a random sample from a Pois(9) distribution. Generate a sample of size 300 from this distribution, similarly failing to record the precise values of observations in excess of 10. Investigate the extent to which inference about λ is improved by the use of this larger sample.

Bayesian Inference and Computational Methods 2009

Homework 5 – Solutions

1. *Simulation of a general finite state space Markov chain.*

   (a) *Exercise.*

   (b) i. Suitable **R** code is

```
> P = matrix(nr=3,c(1/2,0,1,1/4,3/4,0,1/4,1/4,0)) # construct P
> P                                                # check
     [,1] [,2] [,3]
[1,]  0.5 0.25 0.25
[2,]  0.0 0.75 0.25
[3,]  1.0 0.00 0.00

> sim = markov(P,1,100000)                # simulate 100000 steps
> table(sim)/100000                       # estimate stationary dist
sim
      1       2       3
0.39919 0.40009 0.20072
```

   Note that the estimate compares well with the true value $\boldsymbol{\pi} = (0.4, 0.4, 0.2)$

   ii. *Exercise.*

   (c) The chain may be simulated, and its stationary distribution estimated, as in the example above. Note that the function `markov` and its output assume the states to be labelled $1, 2, \ldots, 11$ rather than $0, 1, \ldots, 10$. Hence, if `sim` is again the vector of simulated states, appropriate code to estimate the mean of the stationary distribution is

```
mean(sim) - 1
```

   Solution of the *detailed balance equations* gives that the true stationary distribution $\boldsymbol{\pi}$ is given by

$$\pi_n = \frac{3}{2^n}\pi_0, \qquad n = 1, 2, \ldots 9,$$
$$\pi_{10} = \frac{1}{2^9}\pi_0.$$

   where $\pi_0 = \left(1 - 2^{-8}\right)^{-1}$.

2. *Simulation of a distribution on the integers.*

   (a) This is similar to 2b below. It is only necessary to modify the **R** function calculating the *target distribution* by replacing `2^(-y)` by `1`.

   (b) Appropriate **R** code, using the supplied functions is

```
> sim = rwm.int(targ2b,0,101000)     # simulate 101000 steps
> length(sim)                        # check length of result
[1] 101001
> sim = sim[-(1:1001)]               # remove burn-in period
> length(sim)
[1] 100000
> table(sim)/100000                  # estimate stat dist of chain
sim
      0       1       2       3       4       5       6       7
0.50170 0.24864 0.12308 0.06189 0.03115 0.01678 0.00811 0.00423
      8       9      10
0.00214 0.00139 0.00089
```

The normalising constant required to turn the given function π into a distribution is $k = (2 - 2^{-10})^{-1}$, so that the required target distribution is given by

0	1	2	3	4	5	6	7
0.50024	0.25012	0.12506	0.06253	0.03127	0.01563	0.00782	0.00391

8	9	10
0.00195	0.00098	0.00049

(c) This is similar to 2b above. Note that the target distribution is here easily seem to be Pois(4).

(d) Again this is similar to 2b above. The normalising constant is here not so easily determined theoretically.

3. *Random-walk Metropolis.* Note that this question is mostly a practical exercise using the supplied **R** functions.

   (a) Good values of the variance (for good mixing) are those comparable in magnitude to 1.

   (b) Initial states x where x is large (and so has low density under the given target distribution) require sufficient *burn-in* time for the simulated chain to move to the high density region.

4. *Allowed and forbidden states.* These are practical exercises.

5. *Bayesian inference for the gamma distribution with known shape parameter.*

   (a) The density of the $\Gamma(4, \lambda)$ distribution is given by $f_\lambda(y) = \frac{1}{6}\lambda^4 y^3 e^{-\lambda y}$ for $y \geq 0$ (and $f_\lambda(y) = 0$ for $y < 0$), and so for the given data, the likelihood function is given by
$$L(\lambda; \boldsymbol{y}) \propto \lambda^{80} e^{-64\,\lambda},$$
again for $y \geq 0$. The prior distribution for λ is given by $\pi(\lambda) \propto e^{-0.1\,\lambda}$ for $y \geq 0$. Hence the posterior distribution is as given.

   (b) A suitable **R** function to calculate the target posterior distribution (which is in fact $\Gamma(81, 64.1)$) except for the normalising constant, which we pretend we do not know, is given by

```
targ55 = function(lambda) {
  targ = 0
  if (lambda >= 0) targ = lambda^80 * exp(-64.1*lambda)
  return(targ)
}
```

   A suitable standard deviation of the step size for the suggested *random-walk Metropolis algorithm* (to simulate a Markov chain whose stationary distribution is the required posterior distribution) with normally distributed steps is any value reasonably comparable with 1.

   (c) As usual a long run of the chain may be simulated and a suitable burn-in period removed. Since the stationary distribution of the chain is the required (target) posterior distribution for λ, the mean and variance of this posterior distribution may be estimated from the sample mean and variance of the sequence of observed states. The true mean and variance of the posterior distribution are 1.2637 and 0.019714 respectively, while its mode is 1.2480.

   (d) Letting `sim` denote the sequence of successive states in a sufficiently long run of the chain, and again recalling that their sample distribution should estimate the posterior distribution for λ, a 90% equal-tailed credible interval for λ is most easily obtained via the following **R** code

```
> quantile(sim,c(0.05,0.95))
    5%    95%
1.0419 1.5048
```

(e) When the prior distribution for λ is taken instead to be Exp(1), the posterior distribution changes very little, reflecting the relative insensitivity of the latter to the prior when there is sufficient data.

6. (a) Let $*$ denote the "observation" that a value greater than 10 has been obtained. Then the probability function for y is given by

$$f(y) = \begin{cases} e^{-\lambda}\dfrac{\lambda^y}{y!}, & y = 0, 1, \ldots, 10 \\ 1 - e^{-\lambda}\sum_{i=0}^{10}\dfrac{\lambda^i}{i!}, & y = *. \end{cases}$$

Hence, for the given data, the likelihood function is as given.

(b) The posterior distribution for λ is given by

$$\pi(\lambda \,|\, \boldsymbol{y}) \propto e^{-20.1\,\lambda}\lambda^{138}\left(1 - e^{-\lambda}\sum_{i=0}^{10}\frac{\lambda^i}{i!}\right)^{10}.$$

(c) *Practical exercise.*

(d) *Practical exercise.*

(e) *Practical exercise.*