HW6, Bayesian Modelling B 2016/17

Jonathan Rougier
School of Mathematics
University of Bristol UK

In the homeworks, questions with marks are officially 'exam-style', although you can expect any homework question to appear as an exam question, unless it is explicitly 'not examinable'.

Nothing to hand in, since the unit finishes today (3 Mar). The answers will be put on the unit webpage next Fri.

1. Have a really good look at Homework 5, to make sure you can follow all the steps of creating a statistical model and Gibbs-sampling its conditional distribution.

2. Show that the set of distributions which satisfy detailed balance with respect to P is a strict subset of the set of stationary distributions of P. [10 marks]

   Answer. $\boldsymbol{\mu}$ satisfies detailed balance with respect to P exactly when

   $$\mu_i P_{ij} = \mu_j P_{ji} \quad \text{for all } i, j.$$

   $\boldsymbol{\mu}$ is a stationary distribution of P exactly when

   $$\textstyle\sum_i \mu_i P_{ij} = \mu_j \quad \text{for all } j.$$

   Every $\boldsymbol{\mu}$ which satisfies detailed balance is a stationary distribution:

   $$\begin{aligned} \textstyle\sum_i \mu_i P_{ij} &= \textstyle\sum_i \mu_j P_{ji} && \text{by detailed balance} \\ &= \mu_j \textstyle\sum_i P_{ji} \\ &= \mu_j \end{aligned}$$

   because the rows of P sum to 1.

   To show that the converse does not hold, we must find a P and a $\boldsymbol{\mu}$ for which $\boldsymbol{\mu}$ is a stationary distribution, but does not satisfy detailed balance. The obvious place to look is periodic P's. Let

   $$P = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}.$$

   Then $\boldsymbol{\mu} = (1/3, 1/3, 1/3)$ is a stationary distribution. However, $\mu_1 P_{12} = 1/3$ but $\mu_2 P_{21} = 0$, and so detailed balance is not satisfied.

3. I was browsing a meta-analysis on the efficacy and safety of statin treatment (see `doi:10.1093/qjmed/hcq165`), which included the following passage:

In order to evaluate the relative effectiveness of each study drug on CVD mortality, we used the Lu-Ades method for combining indirect evidence in mixed-treatment comparisons.[31] We estimated the posterior densities for all unknown parameters using MCMC (Markov chain Monte Carlo) for each model. Each chain used 20 000 iterations with a burn-in of 20 000, thin of 5 and updates varying between 80 and 110. We used the same seed number (SEED = 314 159, equivalent to 10 pi) for all chains. The choice of burn-in was chosen according to Gelman–Rubin approach.[32] We applied the covariate of LDL-C change and also statin dosing (high or moderate determined by the Canadian Compendium of Pharmaceuticals and Specialties),[33] using an approach developed by Cooper *et al.*[34] We assessed convergence based on trace plots and time series plots. The accuracy of the posterior estimates was done by calculating the Monte Carlo error for each parameter. As a rule of thumb, the Monte Carlo error for each parameter of interest is less than ~5% of the sample standard deviation. All results for the mixed-treatment ana-

(a) Explain how the authors assessed the convergence of their MCMC sampler. [15 marks]

Answer. We are told that the authors used the Gelman-Rubin approach. Almost certainly they mean they inspected the `gelman.diag(rsam)` and `gelman.plot(rsam)` diagnostics, where `rsam` is the output from an MCMC sampler, as a `coda` object. These diagnostics require multiple chains, but we are not told how many they used. We are also told that convergence was assessed based on trace plots and time series plots. There seems to be some confusion about how these authors assessed convergence! The command `traceplot(rsam)` would produce time series plots, but perhaps they mean they looked at the autocovariance functions for each quantity? We don't know.

Let's assume that they used `gelman.diag(rsam)`. In this case they estimate a scalar uncertainty measure for each random quantity in two ways. First, by averaging the results from assessments made separately on multiple chains; call this $\bar{w}$. Secondly, by pooling all the chains together, call this b. If the initial values for the chains are widely dispersed relative to the target (the posterior distribution), then $\bar{w}$ will underestimate the uncertainty, and b will overestimate it, and hence $b/\bar{w}$ will exceed 1. Once the chain has converged, the ratio will be 1. So `gelman.diag(rsam)` computes the ratio $b/\bar{w}$, and also the upper bound on a 95% confidence interval for it, so that we can assess whether the ratio is at or above 1, for the number of iterations we have done. `gelman.plot(rsam)` does the same thing, except for longer and longer subchains, so that we can see from a plot when the ratio $b/\bar{w}$ first 'touches down' at 1.

(b) Explain how the authors assessed the accuracy of their MCMC estimates. [10 marks]

Answer. They computed the Monte Carlo Standard Error (MCSE) for each random quantity. There are a number of ways of doing this: I like batch means, using the `batchSE(rsam)` command. One long simulation is divided into q batches of length a, each one large enough that the sample mean from the batches are—one hopes!—mutually independent. Then these sample means are combined together as independent estimates of the true expectation, to give a measure of uncertainty, which takes account of the

number of batches. This approach can be applied to a single chain, but we get a much better estimate from multiple independent chains, because in this case estimates from independent chains are definitely mutually independent.

The authors satisfied themselves that the MCSE was less than 5% of the sample standard deviation for each random quantity. This is a common rule of thumb.

(c) Identify the information *not* given in this text which you would need to replicate their assessment of convergence and of the accuracy of their estimates. [10 marks]

Answer. See above. I would want to know how many chains they used for both parts. I like at least 8 for assessing convergence, and at least 4 for computing the MCSEs. I would want to see the diagnostics for convergence, say the figure produced by `gelman.plot(rsam)` for each random quantity. I would also want to see `summary(rsam)` which has the sample standard deviation and also the MCSE for the estimated expectation for each random quantity.

4. (Not examinable) Study the handout on convergence diagnostics, and create an R function `brooks.diag` which implements this test. You may find it helpful to look at the first few lines of `gelman.diag` in the `coda` package in R. Insert your `brooks.diag` function into your `Rats.R` script, to test the convergence of the sampler.

Answer. Here is my function.

```
#### Other Brooks & Gelman diagnostic

## from Brooks & Gelman (1998), sec 3; also Lunn et al (2013), The
## BUGS book, sec 4.4.2.

brooks.diag <- function(x, level = 0.8) {

  x <- as.mcmc.list(x)
  if (nchain(x) < 2)
    stop("You need at least two chains")
  if (level <= 0 || level >= 1)
    stop("\'level\' should lie between 0 and 1")
  alpha <- 1 - level
  probs <- c(alpha/2, 1 - alpha/2)

  ## apply diagnostics to second half of chain

  x <- window(x, start = end(x)/2 + 1)
  x <- lapply(x, as.matrix)

  ## each chain in turn

  widths <- sapply(x, function(y) {
    ci <- apply(y, 2, quantile, probs = probs)
    ci[2, ] - ci[1, ]
  })
  wbar <- rowMeans(widths)

  ## all chains together
```

```
  y <- do.call("rbind", x)
  ci <- apply(y, 2, quantile, probs = probs)
  b <- ci[2, ] - ci[1, ]

  b / wbar
}
```