STAT40850 - Bayesian Analysis

Computer Aided Lab 4 – Chain convergence

Submission deadline: April 17th at 10am.

1 Obtaining samples from the posterior distribution

Gibbs sampling and Metropolis-Hastings enable us to sample from a Markov chain whose stationary distribution is the posterior distribution of the parameters given the data. A key question is: how do we know when we have sampled enough values to treat them as though they came from the posterior? When we are happy that the samples we have obtained come from the posterior distribution we say that the MCMC has *converged*. Today we will look at some techniques for assessing the convergence of MCMC runs.

If we are to treat our posterior simulations as though they are iid samples from a stationary probability distribution, we would hope that the mean, variance, and other moments would be stable over the iterations we have run. Similarly, we would hope that the correlation between the samples at different lags (ie autocorrelation) should be small. A useful method for determining whether the runs have converged is to run the Gibbs sampling steps using different starting values. Each different run is called a *chain*. If a chain has converged, then it should be similar in distribution to other chains. When multiple chains reach the same stationary distribution, we say that they have *coupled*. **coda** contains a number of simple techniques to determine whether chains have coupled.

We can help the convergence process by specifying our models suitably. If the parameters are highly correlated then they will tend to be unstable and we will struggle to estimate them using Gibbs sampling. If we can re-parameterise our models so that the coefficients are less correlated we will speed up convergence. We have already met such a situation in the linear regression examples of Lab 3 where we compared standard and mean-corrected linear regression. The mean-corrected regression produced less correlated parameters and will speed up the convergence of the chain, even though the two models are identical.

Task:

- Open the files `ratstumours.R` and `ratstumours.model`. The model implements the hierarchical rat tumours example from Lecture 16 (note that we are using different prior distributions to that used in lectures). Recall that this is an overdispersed model for the Binomial counts data. Run the model for 1000 iterations whilst monitoring the parameters `pi`, `alpha` and `beta`. Look at the posterior samples of the parameters as trace plots. Do the means and variances appear stable over iterations?
- Now use coda.samps with 2 different chains. To do this, simply change the number of chains to 2 and set inits in `jags.model` to use the two sets of starting values given using inits=list(inits1,inits2). Look at the trace plots again
  plot(rats.samps,ask=TRUE)
  (setting ask to TRUE means `R` will pause between plots until you hit return). Do the chains appear to have coupled?

2 Checking for chain convergence

Aside from simple inspections of the trace plots, **coda** provides some specific visual tools for assessing convergence. The simplest of these is an autocorrelation plot given by autocorr.plot() or acfplot() . These plots show the within-chain autocorrelation at different lags. Ideally all autocorrelations should be zero after lag 1.

Task:

- For the rats tumour examples above, what do the autocorrelation plots look like? Are there autocorrelations beyond the first lag?

A more sophisticated measure of assessing convergence is provided by the *Brooks-Gelman-Rubin* shrink factor (known by **coda** as gelman.diag). The diagnostic requires at least 2 chains to have been run before it can be calculated and it works by comparing the between-chains and within-chains sums of squares, form which we can determine the performance of the sampler. Ideally the between-chains distances should be small in comparison to the within-chains. When the value of the BGR shrink factor approaches 1, we can say that we are likely to be sampling from the posterior distribution. We will cover this diagnostic in more detail in lectures.

Task:

- Run the multiple chains version of the rat tumours example above, and compute the BGR diagnostic for the chains. The BGR shrink factor can report convergence too early if it happens to be close to 1 by chance. By calculating the shrink factor at several points in time, gelman.plot() shows if the shrink factor is still fluctuating. Look at plots using gelman.plot(). Do they look like they have converged? If so, at what iteration would you be happy to believe you are now sampling from the posterior distribution?

3 Dealing with convergence problems

In many Bayesian models, we are likely to suffer from posterior samples that do not immediately appear to have come from the posterior distribution. This can be because we have chosen poor starting values, or because the model we are fitting is poorly parameterised or too sophisticated to fit quickly. When we have such problems we may remove the first set of iterations from our subsequent analysis. The iterations we remove are known as the *burn-in period*. Similarly, if we are experiencing a problem with autocorrelation, we may only keep every kth iteration. This process is known as *thinning*. The autocorrelation plot and the BGR diagnostic may help us in determining how many iterations we remove as a burn-in, and the degree to which we need to thin the values.

Most importantly, we should only perform subsequent posterior analysis on the values remaining after the discarded iterations are removed. Otherwise we are using values which have not come from the posterior distribution.

Task:

- After running the rats tumours data with multiple chains, change the `n`.adapt argument in `jags`.model() and `thin` argument in `coda`.samples() to remove the burn-in period and thin the values. Run the model for a further 1000 iterations; have the chains have fully converged?

4 Homework: hepatitis vaccinations

Write code to fit the hierarchical linear model of Example 3 in Lecture 16 to the Hepatitis data set in `hepatitis.dat`. Write a short report ($\approx$ 2 pages) outlining the model you have fitted and the steps you have taken to ensure convergence. Give an estimate of the mean slope and mean intercept including a 95% credible interval. Include your code as an appendix.

The data are y_{ij} as the log anti-HB titre (the amount of surface antibody) for the jth observation on the ith infant. There are $n = 52$ infants and $m = 7$ observations on each. Observations are taken for each infant at 7 set times after the vaccination was administered. These are set at $t = 1.0, 2.5, 5.0, 7.0, 10.0, 15.0, 20.0$ minutes. A suitable model is expected to use $\log(t)$ as an explanatory variable (as opposed to t in the lecture; all else the same). The data y_{ij} in the file `hepatitis.dat`
Hint: Recall from Lab 2 that `dgamma(0.001,0.001)` can be used as the Jeffreys' prior on a variance parameter i.e. $p(\phi) \propto \frac{1}{\phi}$.