Convergence Diagnostics with CODA

As previously discussed, a value from the target density π is only obtained when the number of iterations of the Markov chain approaches infinity. In practice, this is not attainable and a value obtained at a sufficiently large iteration is taken. The difficulty is to decide how large this iteration, resp. the burnin period should be. There is no simple aswer to this question and most efforts have been directed at studying the convergence characteristics of the chain. There are two main ways to approach the study of convergence.

1. The first one is more theoretical and tries to measure distances and establish bounds on distribution functions generated from the chain. In particular, one can study the *total variation distance* between the distribution of the chain at iteration j and the limiting distribution π. This is an area of ongoing research but so far, the results have had little impact on practical work.

2. The study of convergence of the chain can also be approached from a statistical perspective. i.e. by analysing the properties of the observed output from the chain. This is an empirical as opposed to a theoretical treatment and obviously more practical. However, it can never guarantee convergence because it is only based on observations from the chain.

CODA is a menu-driven set of R functions for analysing output (bugsoutput.txt and bugsoutputIndex.txt files) obtained by WInBUGS. CODA performs convergence diagnostics and statistical and graphical output analyses. Text output from CODA is displayed on screen and stored in a file called CODA.LOG. Hard copies of the graphical output may be obtained directly by using the "print" option in the graphics device window, or by storing the image as a postscript file. You can find a documentation on CODA, *CODA: Convergence Diagnosis and Output Analysis Software for Gibbs sampling output, Version 0.3. MRC Biostatistics Unit, Cambridge,* written by Best, Cowles, and Vines (1995), MRC Biostatistics Unit, at

`http://www.mrc-bsu.cam.ac.uk/bugs/classic/coda04/readme.shtml`

CODA is being maintained and distributed by the same research group reponsible for BUGS.

Martyn Plummer has translated and further developed CODA for R, the freeware equivalent of S-plus. CODA version 0.5-1 for R can be obtained from his website:

`http://www-fis.iarc.fr/coda/`

which contains a link to the Comprehensive R Archive Network (CRAN)

`http://cran.r-project.org/mirrors.html`

Background Reading:

There are two useful reviews of MCMC convergence diagnostics. The first one is by the original author of CODA:

1. Cowles, MK and Carlin, BP (1995) Markov Chain Monte Carlo diagnostics: A comparative review, *J Amer Stat Soc* 91, 883-904.
2. Brooks, SP and Roberts, GO (1998) Assessing convergence of Markov Chain Monte Carlo algorithms, *Statistics and Computing* 8, 319-335.

If you use R and have downloaded and installed the CODA package, you need to attach it

```
> library(coda)
```

and then invoke CODA with the command

```
> codamenu()
```

It will display the startup menu:

```
CODA startup menu

1: Read BUGS output files
2: Use an mcmc object
3: Quit

Selection:
```

After specifying the files with the samples and reading in the data, CODA comes up with the CODA Main Menu:

```
CODA Main Menu
**************

1: Output Analysis
2: Diagnostics
3: List/Change Defaults
4: Quit
Selection:
```

The Output Analysis encompasses plots and statistics:

```
CODA Output Analysis Menu
*************************

1: Plots
2: Statistics
3: List/Change Defaults
4: Return to Main Menu
Selection:
```

Option 1 of the CODA Output Analysis Menu provides traceplots and kernel density plots of all monitored variables. The general approach to monitoring convergence of Markov chain simulations is to look at traceplots of at least two parallel sequences, typically four or more. To obtain these, it is desirable to choose starting points which are widely dispersed.

Option 2 of the CODA Output Analysis Menu supplies more detailed summary statistics of all marginal posterior distributions than the stats command of BUGS already gives you. Furthermore, you can change defaults (like 2.5%- and 97.5%-iles to 5% and 95%-iles, for example).

The CODA Diagnostics Menu provides some of the most often used tests of convergence.

```
CODA Diagnostics Menu
*********************

1: Geweke
2: Gelman and Rubin
3: Raftery and Lewis
4: Heidelberger and Welch
5: Autocorrelations
6: Cross-Correlations
7: List/Change Defaults
8: Return to Main Menu
Selection:
```

We will discuss these in a bit more detail.

1. **Geweke (1992)** proposes a convergence diagnostic based on standard time-series methods. It is based on a single chain and is appropriate when convergence of the mean (of some function) of the sampled variable is of interest. The chain is divided into 2 “windows” containing the first 10% and the last 50% of the iterates. If the whole chain is stationary, the means of the values early and late in the sequence should be similar. The convergence diagnostic Z is the difference between the 2 means divided by the asymptotic standard error of their difference. As $n \to \infty$, the sampling distribution of Z goes to $N(0,1)$ if the chain has converged. Hence values of Z which fall in the extreme tails of $N(0,1)$ indicate that the chain has not yet converged. CODA also gives you the option to plot the Z-scores.

2. **Gelman and Rubin’s (1992)** approach to monitoring convergence is based on detecting when the Markov chains have forgotten their starting points, by comparing several sequences drawn from different starting points and checking that they are indistinguishable. There are many ways to compare parallel sequences, the most obvious approach being to look at overlaid traceplots and see if the two sequences can be distinguished.

   A more quantitative approach to answer the question “*Are the sequences much farther apart than we could expect, based on their internal variability?*” is based on the analysis of variance: Approximate convergence is diagnosed when the variance between the different sequences is no larger than the variance within each individual sequence. Assume we have m parallel simulations each of length n of the variable X. The values are denoted by $x_{ij}, i = 1, \ldots, m, j = 1 \ldots, n$. The between-sequence variance B and the within-sequence variance W is computed:

$$\begin{aligned} B &= \frac{n}{m-1} \sum_{i=1}^{m} (\bar{x}_{i.} - \bar{x}_{..})^2 \\ W &= \frac{1}{m} \sum_{i=1}^{m} s_i^2 \text{ where } s_i^2 = \frac{1}{n-1} \sum_{j=1}^{n} (x_{ij} - \bar{x}_{i.})^2. \end{aligned}$$

From the two variance components, two estimates of the variance of X in the target distribution are constructed: First

$$V\hat{a}r(X) = \frac{n-1}{n} W + \frac{1}{n} B$$

is an estimate of the variance that is *unbiased* under stationarity (that is, if the starting points of the simulations were actually drawn form the target distribution), but is an *overestimate* unter the more realistic assumption that the starting points are overdispersed.

For any finite n, the within-sequence variance W should *underestimate* the variance of X because the individual sequences have had no time to range over all of the target distribution and, will have less variability. In the limit, as $n \to \infty$, both $V\hat{a}r(X)$ and W approach $Var(X)$, but from opposite directions.

One can now monitor the convergence of the MC by estimating the factor by which the conservative estimate of the distribution of X might be reduced: that is, the ratio between the estimated upper and lower bounds for the standard deviation of X, which is called *estimated potential scale reduction* or *shrink factor*:

$$\sqrt{\hat{R}} = \sqrt{\frac{V\hat{a}r(X)}{W}}.$$

As the simulation converges, the shrink factor declines to 1, meaning that the parallel Markov chains are essentially overlapping. If the shrink factor is high, then one should proceed with further simulations.

The Gelman and Rubin diagnostics calculated by CODA are the 50% and 97.5% quantiles of the sampling distribution for the shrink factor. These quantiles are estimated from the second half of each chain only. You can also produce plots to illustrate these diagnostics, i.e. overlaid traceplots and shrink factor plots.

3. **Raftery and Lewis (1992)** developed a method that can be used to determine the number of burn-in iterations to discard and the *thinning*, i.e. the number k for storing every kth iterations. This will reduce the amount of storage, especially when consecutive iterations are highly correlated, necessitating a long run.

   You must specify the quantile to be estimated (default is 2.5%), the desired degree of accuracy for the estimate of this quantile (default = ±0.005), and the required probability of attaining this degree of accuracy (default is 0.95). CODA then reports

   - Nmin, the minimum number of iterations that would be needed to estimate the specified quantile to the desired precision if the samples were independent. This is a theoretical value based on the binomial variance and provides a lower bound for the run-length of the Gibbs sampler. Nmin will increase as the required probability and degree of accuracy increase.

- N, the total number of iterations that should be run for each variable.
- M, the number of initial iterations to discard as the burn-in.
- k, the thinning interval to be used.
- $I = \frac{N}{Nmin}$, which measures the increase in number of iterations needed to reach convergence due to dependence between the samples in the chain. Values of I much greater than 1 ($I > 5$ indicate high within-chain correlations and likely convergence failure and reparametrization is advised.

4. **Heidelberger and Welch' (1993)** method basically uses the Cramer-von-Mises statistic to test the null hypothesis that the sampled values for each variable form a stationary process. If the null hypothesis is rejected for a given variable, the test is repeated after discarding the first 10% of iterations. If the hypothesis is again rejected, a further 10% of iterations are discarded. This process is repeated until either a portion of the chain passes the test, or 50% of the iterations have been discarded and H_0 is still rejected. In the latter case, CODA reports the Cramer-von-Mises statistic and indicates that the stationarity test was failed.

   If the stationarity test is passed, CODA reports the number of iterations to keep, the number of initial iterations to discard and the Cramer-von Mises statistic. A halfwidth test is then carried out as follows: for each variable the portion of the data which passed the stationarity test is used to estimate the asymptotic standard error of the mean via a time-series method. CODA reports the sample mean of the retained iterates and the halfwidth of the associated 95% confidence interval for this mean. If the halfwidth is less than ϵ (default is 0.1) times the sample mean, the halfwidth test is passed and the retained sample is deemed to estimate the posterior mean with acceptable precision. If the halfwidth test is failed, this implies a longer run is needed.

5. CODA calculates **autocorrelations** within each chain for each monitored variable at lags of 1, 5, 10, and 50. High autocorrelations within chains indicate slow mixing and slow convergence. Reparametrizations might help. It might be necessary to increase the thinning interval to achieve a less highly correlated sample. CODA also produces autocorrelation plots.

6. CODA provides a table of **cross-correlations** between the monitored variables for each chain. High correlations among parameters are associated with slow convergence and may indicate a need for reparametrizations. CODA also gives you the option to plot cross-correlations.