Chapter 6
Checking Convergence to Posterior Distribution

OpenBUGS uses Markov chain Monte Carlo (MCMC) sampling to generate values directly from the target posterior distribution. There are theoretical results that guarantee convergence to the posterior distribution, under very general conditions, as described in [1]. However, from a practical perspective it takes time for the MCMC sampling to converge to the posterior distribution; any values sampled prior to convergence should not be used to estimate parameter values. For simple problems involving one parameter, such as p in the binomial distribution or λ in the Poisson distribution, 1,000 iterations will be more than sufficient for convergence. In more complicated problems, which usually involve inference for more than one parameter, this may not be the case, and the user will have to check for convergence. This chapter presents qualitative and quantitative convergence checks that an analyst can use, using features built into OpenBUGS. We address three important issues related to convergence:

1. Convergence to the joint posterior distribution,
2. Coverage of the parameter space,
3. Number of samples needed after convergence.

6.1 Qualitative Convergence Checks

For problems with more than one parameter, which are the problems in which convergence becomes an issue, we recommend that at least two chains be used, with starting values that are dispersed around the estimated mode of the posterior distribution. Usually the analysis will not be very sensitive to the initial values selected for the chains, but this is not always the case. For example, when modeling population variability (see Chap. 7), there are mathematical approaches that can be used to select initial values.

D. Kelly and C. Smith, *Bayesian Inference for Probabilistic Risk Assessment*,
Springer Series in Reliability Engineering, DOI: 10.1007/978-1-84996-187-5_6,

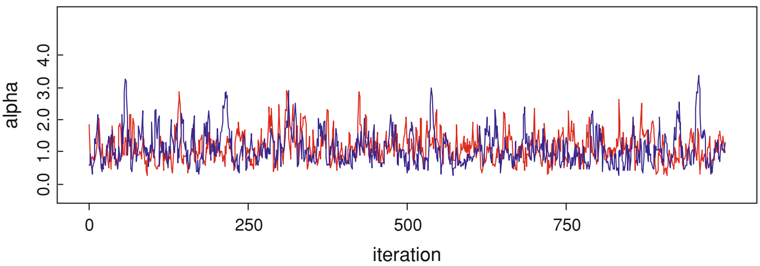
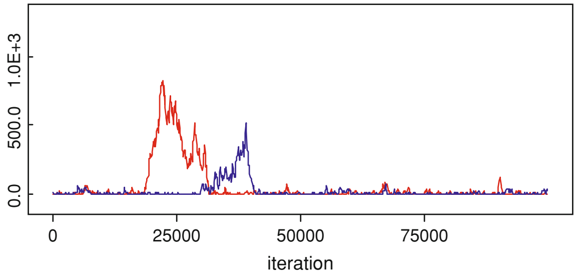


Fig. 6.1 History plot showing two well-mixed chains, indicative of convergence

Fig. 6.2 History plot showing two poorly-mixed chains, indicative of failure to converge



After the model has been compiled (remember to specify the number of chains before compiling the model) and initial values have been loaded, specify the nodes to be monitored. All parameters should be monitored and convergence should be checked for each of these monitored nodes. Now run 100–1,000 samples and select *History* from the *Inference* menu in OpenBUGS to generate a trace of these samples for each monitored node. A plot like the one shown in Fig. 6.1, in which the two chains are well mixed, is indicative of convergence.

In contrast, Fig. 6.2 shows a case in which the chains are not well mixed, which indicates that more iterations must be run to achieve convergence.[1]

6.2 Quantitative Convergence Checks

OpenBUGS has a built-in convergence diagnostic that can be used in conjunction with the history plots shown above to help the user decide when enough burn-in samples have been taken. It requires two or more chains, and is based on an

[1] It is possible, especially with highly correlated parameters, that there will be difficulty in getting the chains to mix, despite convergence. Since we cannot readily distinguish between the two problems, we will refer to poor chain mixing as being a sign of failure to achieve convergence. Regardless of the source of the lack of mixing, the estimates should not be used until the problem is rectified, perhaps by reparameterizing the problem in terms of parameters that are less strongly correlated.

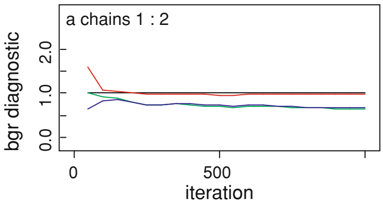


Fig. 6.3 BGR plot illustrating convergence

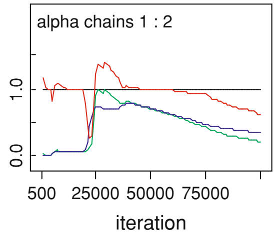


Fig. 6.4 BGR plot of history in Fig. 6.2, illustrating lack of convergence

analysis of the variance within- and between-chains. If the chains have converged, all chains should have approximately the same within-chain variance, and this should be approximately equal to the between-chain variance estimate. The Brooks-Gelman-Rubin (BGR) diagnostic in OpenBUGS looks at a ratio of these estimates, which is normalized to equal one when the chains have converged.

To implement the BGR diagnostic, run at least two chains, as described above, then select *bgr diag* from the *Inference* menu. The resulting plot will have three curves. The estimate of the within-chain variance is shown in blue, the between-chain estimate is in green, and the normalized BGR ratio is shown in red. The BGR ratio is expected to start out greater than one if the initial values are widely dispersed. The heuristic is that this ratio should be less than about 1.2 for convergence. In addition, the between-chain and within-chain estimates shown by the green and blue curves should be stable. Right-clicking on the BGR graph allows the analyst to bring up a table of the values over the history.

Figure 6.3 shows a typical BGR plot for a problem that has converged. Figure 6.4 shows the plot of BGR for the history in Fig. 6.2, confirming the failure to converge suggested by the history plot.

6.3 Ensuring Adequate Coverage of Posterior Distribution

The second issue the analyst faces is whether the chains are providing good coverage of the posterior distribution. It can happen, especially because the samples from MCMC are dependent, that a chain becomes stuck in a particular region of the parameter space. If this happens, the resulting parameter estimates can be substantially in error. Running multiple chains with widely dispersed initial values is the first line of defense against this problem. OpenBUGS also calculates a

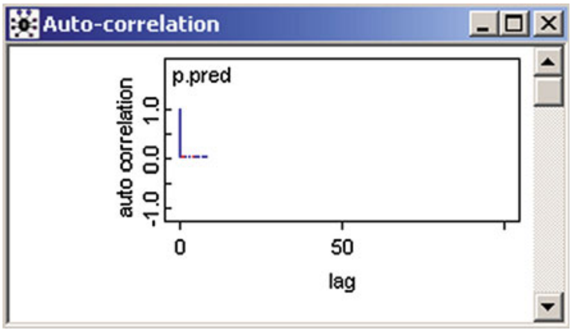


Fig. 6.5 Plot of lag autocorrelation coefficient exhibiting rapid decrease to zero with increasing lag

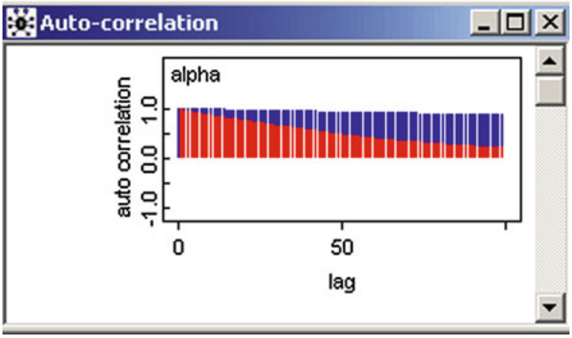


Fig. 6.6 Plot of lag autocorrelation that may be indicative of poor coverage of posterior distribution

lag autocorrelation coefficient, which measures the degree of dependence between samples from the chains. Ideally, this dependence will fall to zero very quickly as the lag increases; high correlation over long distances in the chain can be an indicator of inadequate coverage of the full posterior distribution. Figure 6.5 shows a plot of the lag autocorrelation that is indicative of good coverage, because the dependence falls quickly to zero as the lag increases. Conversely, Fig. 6.6 shows a plot in which the MCMC samples exhibit a high degree of dependence, even with high lag. Such a plot may be indicative that the chains are not providing good coverage of the posterior distribution.

6.4 Determining Adequate Sample Size

The third issue related to convergence is determining how many samples to take following burn-in (i.e., convergence to the posterior distribution). The discussion in this section is predicated upon having good coverage of the posterior distribution.

The uncertainty of a parameter estimate can be decomposed into two parts, measured by the variance, or its square root, the standard deviation: the "true" uncertainty and the additional uncertainty introduced by Monte Carlo sampling. OpenBUGS computes a sample standard deviation for each monitored stochastic node, which is a measure of the overall uncertainty. It also computes "MC error," which is a measure of the second component, the uncertainty introduced by Monte Carlo sampling. We need to have enough samples after convergence so that MC error is a small contributor to the overall standard deviation, no more than a few percent. The OpenBUGS User Manual suggests 5% as an upper limit; in practice, for the types of problems encountered in PRA, we feel that 2–3% is a better guideline.

Reference

1. Robert CP, Casella G (2010) Monte Carlo statistical methods, 2nd edn. Springer, Berlin