with other kinds of joint distribution, such as those involving large spatial fields, which are often not expressed hierarchically (see, e.g., Cressie and Wikle, 2011).

3.5 *Practical issues*

The convergence of Cesàro averages theorem (Theorem 3.4) promises only asymptotic convergence of expectations (in mean square). We do not have an infinite amount of time and so we must plan for imperfect convergence. The two issues are:

1. At what n_0 has our Markov chain 'forgotten' its starting distribution, in which all probability is concentrated on a single element of $\mathcal{X}$? We can drop the first n_0 elements from our sequence.
2. If we stop at time n, how can we quantify the accuracy in our Cesàro average estimate of $\mathrm{E}\{g(X); \boldsymbol{\pi}\}$? By monitoring this value for increasing n, we can aim for a particular level of accuracy.

Both of these questions are under active development, and so it is sensible to augment the suggestions in this section with some more up-to-date reading.

3.5.1 *Assessing convergence*

We want to compute the expectation of $g(X)$ with respect to target distribution $X \sim \pi$, and we have arranged for $x_0, X_1, X_2, \ldots$ to be a sequence from a Markov chain with stationary distribution π satisfying the Ergodic Theorem, so that if $\bar{G}_n := n^{-1} \sum_{i=1}^{n} g(X_i)$, then

$$\bar{G}_n \xrightarrow{\text{m.s.}} \mathrm{E}\{g(X); \pi\}.$$

If we let $n \to \infty$ then what happens at the start of the chain does not matter to $\bar{G}_n$: it gets 'rubbed out' by what happens later on. But if n is stubbornly finite, as it must be in practice, then what happens at the start of the chain can still matter to the behaviour of $\bar{G}_n$.

At the start of the sequence we supplied the initial value $X_0 = x_0$, which is a degenerate probability distribution entirely concentrated on one element of $\mathcal{X}$, if this point is chosen deterministically (e.g., a prior mode). The n-step transition probabilities will converge to the stationary distribution geometrically, according to Theorem 3.2, but this does not mean that they will converge almost immediately for every possible starting point. So in practice we can improve the performance of $\bar{G}_n$ for finite n by discarding the first part of the sequence, for which the n-step transition probabilities have not yet converged. This first 'discardable' part is termed *burn-in*.

There are some simple heuristics for burn-in, such as 'discard the first 10% of your simulations'. However, it is much better to have a more adaptive method which is sensitive to the expectation(s) that are to be estimated. The method given here is due to Brooks

and Gelman (1998, sec. 3), and also described in Lunn et al. (2013, sec. 4.4.2).

Simulate m independent sequences each of length n, with starting points (x_0 values) which are well-dispersed relative to the target distribution (see below). Consider any scalar summary, $g : \mathcal{X} \to \mathbb{R}$. From each sequence, compute the width between the 10th and 90th percentiles from the *second half* of each sequence.[9] Denote these widths $w_1, \ldots, w_m$, and denote their arithmetic mean $\bar{w}$. Now repeat this process for the merged values from the second halves of all m sequences, to compute one value, denoted b.

[9] These percentiles appear in Brooks and Gelman (1998, p. 443).

If the sequences have converged after $n/2$ iterations, then $\bar{w}$ and b will both be estimating the same thing, namely the true width of the 80% equitailed credible region of $g(X)$. In this case the ratio $b/\bar{w}$ will be approximately 1. But if the sequences have not converged, then $\bar{w}$ will be an under-estimate, and b will be an over-estimate, if the starting points are well-dispersed relative to the target distribution. In this case the ratio $b/\bar{w}$ will be larger than 1. So values of $b/\bar{w}$ larger than, say, 1.05, indicate that more than $n/2$ iterations are required for convergence.[10]

[10] The threshold 1.05 appears in Lunn et al. (2013, p. 75).

This approach can be applied just to the functions of interest, or it can be applied to each of the components of x in turn, or perhaps just to a few of the more difficult ones. For safety, we would want all of the ratios to be no larger than, say, 1.05.

Well dispersed starting points can be tricky to achieve, without some knowledge of the properties of the target distribution. For Bayesian inference, though, we can use points sampled independently from the prior distribution, which is usually straightforward. If the prior distribution has a very flat distribution for the hyperparameters (see Section 2.6), then it is possible that the simulation algorithm for the Markov chain targeting the posterior distribution will perform badly for x_0 values sampled from the prior distribution. In this case, another option is to condition the prior distribution on a small fraction of the observations (to concentrate it a little), and use MCMC to sample the starting points from this distribution.

If the sequences have not converged, then all m of the sequences need to be simulated further (say another $2n$ iterations each) and then the test needs to be applied again, and so on. Otherwise, the first $n_0 = n/2$ iterations of each sequence are burn-in, and discarded. One or more of the sequences can be simulated further to increase the total number of available iterations above $mn/2$, if more iterations are required (see Section 3.5.3).

3.5.2 *Monte Carlo standard errors*

We now assume that burn-in has been discarded.

As above, specify $g : \mathcal{X} \to \mathbb{R}$ and let

$$\bar{G}_n := \frac{1}{n} \sum_{i=1}^{n} g(X_i). \tag{3.11}$$

The variance of $\bar{G}_n$ is

$$\operatorname{Var}(\bar{G}_n) = \frac{1}{n^2}\left[\sum_{i=1}^{n}\operatorname{Var}\{g(X_i)\} + 2\sum_{i=1}^{n-1}\sum_{j=2}^{n}\operatorname{Cov}\{g(X_i), g(X_j)\}\right],$$

where all variances and covariances are taken with respect to the target distribition π. This satisfies

$$n\operatorname{Var}\{\bar{G}_n\} \longrightarrow \sigma_g^2, \tag{3.12}$$

for some finite constant σ_g^2, using similar arguments to Section 3.2, in the proof of Theorem 3.3. Therefore, if we estimate σ_g^2 from the simulated sequence, then we can use

$$\mathrm{se}_{g,n} := \sigma_g/\sqrt{n} \tag{3.13}$$

as an estimate of the standard error of $\bar{G}_n$, termed the *Monte Carlo Standard Error* (MCSE). As usual (applying a Central Limit theorem),

$$\bar{g}_n \pm 2\,\mathrm{se}_{g,n}$$

is approximately a 95% confidence interval for $\mathrm{E}\{g(X);\pi\}$. If we estimate this interval for our n and it is too wide, then we simulate the sequence further. For example, if the interval is double what we can tolerate, then we need to increase n to $4n$, which means doing $3n$ additional iterations.

To estimate σ_g^2, consider the arithmetic mean of a sample of size a. If we divide the total sample of size n into q batches of size a, then for each batch we have

$$\bar{G}_{a,j} := \frac{1}{a}\sum_{i=a(j-1)+1}^{aj} g(X_i) \qquad j = 1,\ldots,q.$$

Each of these values has variance approximately σ_g^2/a from (3.12), and if a is large enough that the values are approximately uncorrelated, then we can estimate their common variance as

$$\frac{\sigma_g^2}{a} \approx \frac{1}{q-1}\sum_{j=1}^{q}(\bar{g}_{a,j} - \bar{g}_n)^2,$$

where $\bar{g}_{a,j}$ is the arithmetic mean from the jth batch of size a, and $\bar{g}_n$ is the arithmetic mean of the entire sample of size n. The $q-1$ in the denominator rather than q is the usual conservative adjustment to give an unbiased estimator. Hence

$$\sigma_g^2 \approx \frac{a}{q-1}\sum_{j=1}^{q}(\bar{g}_{a,j} - \bar{g}_n)^2. \tag{3.14}$$

The suggestion is to set $a = \lfloor\sqrt{n}\rfloor$.[11] This approach to estimating σ_g^2 is known as *batch means*.

[11] Appears in Lunn et al. (2013, p. 78).

3.5.3 *One long sequence or several short ones?*

If you have a fixed budget of CPU cycles, then you can spend them on simulating one long sequence of length mn, or on m independent sequences of length n.

If you are not going to check for convergence, then one long sequence is much more efficient, because you have to discard burn-in at the start of each sequence. So following a rule such as 'discard the first 10% of your sequence and hope for the best' you would do one long sequence, which will give you an estimate based on a total of $0.9mn$ iterations. But of course you run the risk of non-convergence, and your sequence may have a high autocorrelation, which means that the standard error is large, and may well be poorly estimated because your batches are correlated.

It is much safer, then, to do m independent sequences of length n. This way you can assess convergence, and find the n_0 which determines burn-in. Now your estimate will be based on $m(n - n_0)$ iterations, which may be less than $0.9mn$ iterations, but the good news is that the sequences will be independent, which means that your standard error will typically be smaller, and better estimated.