Comment

Nicholas G. Polson

The goal of Markov chain Monte Carlo (MCMC) algorithms is to provide answers in multidimensional statistical models that are computationally quicker than other techniques such as importance sampling or numerical integration. However, not all MCMC algorithms lead to procedures that are more efficient than these other techniques. Both Geyer, and Gelman and Rubin propose empirical approaches to assess when a particular MCMC procedure is useful. Unfortunately, I am skeptical about the potential for any empirical diagnostics in the MCMC setting.

Two desirable properties for a MCMC procedure are: (1) to provide estimators that, with arbitrarily high probability, approximate the quantity of interest to any specified level of accuracy and (2) to perform this task "quickly," say in polynomial time. Unfortunately, the two procedures presented cannot establish these properties. Geyer proposes one long run of the chain together with a time-series analysis of the output, whereas Gelman and Rubin propose multiple runs using an overdispersed initial distribution together with a diagnostic approach to stop the chain. I will show that when (1) and (2) are satisfied there is no need to "diagnose" convergence or perform a time-series analysis of the chain. One long run of the chain is sufficient and run lengths can be bounded using the second eigenvalue of the Markov chain. These bounds are stronger than those obtained using central limit theorem arguments. On the other hand, when (2) does not hold, it is theoretically unclear whether the output has any informational content and whether diagnostics, multiple runs or time-series analyses of the chain can help solve (1). In this discussion I will focus on techniques for checking (2) for any MCMC procedure.

In the following I will discuss several topics related to properties (1) and (2): fast convergence of the chain, selection of the "burn-in" period, how long to run the chain and caveats associated with a purely diagnostic approach. To introduce notation and fix ideas, consider a time reversible ergodic Markov chain defined on a finite state space, V. Let π be the distribution that we wish to sample from and h: $V \to \Re$ be the functional whose expectation under π, $E_\pi(h)$, is the quantity of interest. Imagine V, as the computer does, to be a fine discretisation of the k dimensional parameter space. I define "quick" algorithms as those that satisfy (1) in $o(|V|)$ rather than $O(|V|)$ operations [importance sampling and numerical integration are $O(|V|)$]. This criterion is central to the issue of why randomised algorithms can be more powerful than deterministic algorithms. Basically, MCMC algorithms meeting this criterion are superior to numerical integration strategies and are called provably convergent.

Let P denote the transition matrix of the chain designed with π as the unique stationary distribution. The efficiency of the algorithm, as we will see, depends on the rate of convergence of the chain, which in turn depends crucially on the second eigenvalue, λ_1. One long run of the chain turns out to be sufficient to generate samples from π and to draw inferences about $E_\pi(h)$. I will show that there is no need to "diagnose" convergence or perform a time-series analysis of the chain *as long as* an a priori bound on λ_1 is available.

To assess the efficiency of a MCMC procedure one may proceed as follows: by Perron-Frobenius theory, the L^1 distance between the distribution after t steps of the chain, $P^t(\varphi)$, and the stationary distribution, $\pi(\varphi)$, is geometrically bounded as

$$\sum_{\varphi} | P^t(\varphi) - \pi(\varphi) | \leq \frac{\lambda_1^t}{\sqrt{\pi(\varphi_0)}},$$

where φ_0 is the initial starting point of the chain and the negative eigenvalues are assumed to be bounded below by $-\lambda_1$ [see, e.g., Diaconis and Stroock (1991)]. Therefore, one can achieve a desired level of sampling accuracy, ε, by running the chain for $T = \log(\varepsilon\sqrt{\pi(\varphi_0)})/\log(1/\lambda_1)$ steps. At first sight this is appealing and looks straightforward to implement and there is no need to "diagnose" convergence from the realised chain. However, for the algorithm to be computationally efficient we need T to be small relative to $|V|$. More precisely we need $T = o(|V|)$. Notice that since $|V|$ is exponential (in k), this essentially requires T to be polynomial (in k).

Demonstrating that a MCMC algorithm is provably (polynomial time) convergent can be a difficult problem. Several papers describe techniques for checking fast convergence of a MCMC algorithm. Applegate, Kannan and Polson (1990) provide a bound for T for Gibbs and Metropolis algorithms by using the notion of conductance to obtain a bound for λ_1 and hence T. Conductance (Sinclair and Jerrum, 1989) is widely used in computer science and has the following intuitive definition: the chain will converge rapidly if the escape probability for each subset S of states is high, as measured by $\Sigma_{\theta\in S,\varphi\notin S}\, \pi(\theta)P(\theta,\varphi)/\Sigma_{\theta\in S}\, \pi(\theta)$, where $P(\theta,\varphi)$

Nicholas G. Polson is Assistant Professor, Graduate School of Business, University of Chicago, Chicago, Illinois 60637.