Comment

Neal Madras

Professors Gelman, Rubin and Geyer have presented us with many interesting ideas to think about. These two papers have been billed as representing opposing sides in a debate, but to my mind they complement more than they contradict each other. This is probably because I find the "debate" somewhat artificial, with each side having its merits as well as its limitations. It is true that one long run gives more statistical information per CPU minute than do several shorter runs (because of warm-up time); however, if you suspect slow mixing for some particular reason, then you should do your best to investigate its likely sources – and multiple runs with intelligently selected starting points are probably the most natural way to detect such a problem. I think that most researchers who use Monte Carlo simulation will find useful things in both papers.

Guarantees are hard to come by in iterative simulation. At present, few Markov chains are sufficiently tractable to yield good rigorous upper bounds on the amount of time necessary to run simulations (although the last paragraph of Geyer's Section 3.5 is a nice observation that may make even relatively weak bounds useful in some cases). What can we do when rigorous analysis eludes us?

Geyer claims that guarantees can come from "experiments with a range of sampling schemes proceeding in small steps from schemes known to mix rapidly to the scheme of interest, making sure at each step that the run is long enough by comparing it to the runs already done." This is a very appealing idea, and it may work very well sometimes, but it is hard for me to believe that this method comes with general guarantees. The problem is that two Markov chains whose transition probabilities are very close may have very different properties. For example, the Ising model in statistical physics is a family of distributions indexed by a parameter $T > 0$ ("temperature") and having a *critical value* T_c (corresponding to a *phase transition* in the associated infinite system). There is a standard implementation of the Gibbs sampler (or the Metropolis algorithm) to the Ising model, giving a Markov chain at each temperature T. We need not be concerned with the details of the model and implementation [which may be found in Geman and Geman (1984) or Ripley and Kirkland (1990)], except to say that the Markov chain is rapidly mixing if T is large and slowly mixing if T is small, and that the change occurs fairly abruptly in a small neighborhood of T_c. Consider a sequence of schemes in which the ith scheme corresponds to the Markov chain at temperature T_i, where $T_1 > T_2 > \cdots$. If $T_3 > T_c > T_4$, say, then the small value of σ^2 at T_3 may provide a false sense of security at T_4, and the entire run at T_4 may be too short to permit the chain to escape some subset of the state space, which is metastable at T_4 but not at T_3. The resulting estimate of σ^2 at T_4 would be far too small. Similar abrupt changes can occur in simulated annealing, perhaps at more than one "critical value" [see p. 677 of Kirkpatrick, Gelatt and Vecchi (1983)].

It is tempting to say that we can avoid such pitfalls if we are careful enough, but the point remains that important questions must be answered before accepting Geyer's claim. First of all, what can be said rigorously about a continuum of sampling schemes? How small should the steps be to ensure that σ^2 does not change too much from one scheme to the next, if it is indeed possible to make such assurances at all? Second, is the procedure feasible in practice? Perhaps for real problems it is just too time-consuming to run lots of simulations of different sampling schemes, and it would be just as efficient and informative to run the scheme of interest for ten times as long. These are intriguing problems that are worth investigating, both theoretically and experimentally.

To provide a context for my remaining remarks, I shall briefly describe one Monte Carlo study of a simple lattice model of polymers known as the *self-avoiding walk*. A linear polymer is a molecule that consists of many "monomers" (groups of atoms) joined sequentially by chemical bonds. The spatial configuration of a linear polymer with N (monomer–monomer) bonds can be modeled as a random walk path $W \equiv w(0), w(1), \ldots, w(N)$ on the three-dimensional integer lattice $\mathbf{Z}^3$; here $w(i)$ represents the location of the ith monomer, and $w(i)$ and $w(i+1)$ are always nearest neighbours in the lattice. But two monomers cannot occupy the same position in space; the simplest model that captures this effect is the *self-avoiding walk*, which is defined by requiring $w(0), \ldots, w(N)$ to be *distinct* sites of $\mathbf{Z}^3$.

Let S_N^o be the set of all N-step random walk paths in $\mathbf{Z}^3$ that start at the origin, and let S_N^* be the subset of walks in S_N^o that are self-avoiding. Let P_N^o and P_N^* denote the uniform probability measures on S_N^o and S_N^*,

Neal Madras is Associate Professor and NSERC University Research Fellow, Department of Mathematics and Statistics, York University, 4700 Keele Street, North York, Ontario M3J 1P3, Canada.

respectively, and let E_N° and E_N^* denote their respective expectations. It is elementary that $E_N^\circ||w(N)||^2 = N$; in contrast, it is believed that $E_N^*||w(N)||^2 \sim CN^q$ as $N \to \infty$, where q and C are constants and $q = 1.18\ldots$. The exponent q is important because it is believed to equal the corresponding exponent for real linear polymers in dilute solution with good solvents.

To estimate q, one needs an efficient method for generating walks with distribution P_N^*. This is a nontrivial problem, but it can be done using a Markov chain method known as the *pivot algorithm* [due to Lal (1969)]. One iteration of the pivot algorithm consists of choosing a site at random on the current walk and applying a randomly chosen reflection or rotation to the part of the walk subsequent to the chosen site; the resulting walk is accepted if and only if it is self-avoiding. Little is known rigorously about this chain except that it is irreducible. However, for the analogous chain for P_N° on S_N° (in which every proposed walk is accepted), Madras and Sokal (1988) proved that (in Geyer's notation) (1) $1 - \lambda_{max} \approx 1/N$, and (2) the "excess variance" σ^2/γ_0 of the quantity $g(W) = ||w(N)||^2$ is $2 \log N + O(1)$, which is much smaller than the universal upper bound $(1 + \lambda_{max})/(1 - \lambda_{max}) \approx N$. Thus the correlations for the quantity of interest decay much more rapidly than do those for the chain as a whole. Simulations indicate that a similar phenomenon occurs for the self-avoiding case as well. This is another example of the caveat that Geyer mentions in Section 3.5.

A related phenomenon is that the length of time needed to approach equilibrium is governed by λ_{max} and not by σ^2/γ_0. In principle, one should not begin sampling before the entire chain has reached equilibrium, and we may not be able to see whether the chain is in equilibrium by just looking at the time series of a single quantity whose autocovariances may decay atypically fast. To be careful, we may need to monitor several different functions that we believe summarize the important features of our chain. (For self-avoiding walks, these might include the number of right-angle turns in the walk, the size of the smallest cube containing the walk, and so on.) The "warm-up" problem is not always easy to handle, and I was surprised by Geyer's dismissal of the need for "formal analysis."

Returning to the pivot algorithm, how should we guard against seriously underestimating its asymptotic variance σ^2? One way is Geyer's continuous family of schemes, where the initial (rapidly mixing) scheme is the one for P_N° and the intermediate schemes are the analogues for probability measures on S_N° that penalize self-intersections without prohibiting them. But as I argued above, this method alone may not eliminate all reasonable doubts. As a check against having been overly optimistic, we could also follow Gelman and Rubin's approach and use several runs. However, one major difficulty with adapting their approach to general state spaces such as S_N^* lies in the creation of an overdispersed starting distribution. Except for SIR, the methods of Gelman and Rubin's Section 2.1 are not applicable to our target distribution P_N^*. Of course, what we would really like to do is generate W's from S_N^* so that $g(W)$ is overdispersed, but it is not at all clear how to do this. One possibility is to choose several initial walks deterministically (some very straight, others very compact), but we could aspire to do better. (For this particular chain, a fortuitous way out of this dilemma is described in the next paragraph.)

My final remark is a counterexample to Gelman and Rubin's assertion that "if we could start by sampling from the target distribution, then we do not need iterative simulation at all" (Section 3.2). Consider this scenario: You have an hour of CPU time available. You have a Markov chain with small "excess variance" σ^2/γ_0, but you don't know how long the chain takes to reach equilibrium. You also have a noniterative procedure that generates variates from the *exact* target distribution, but it does so relatively slowly; say it requires about 5 minutes per variate. You could use the whole hour to get 12 i.i.d. variates, but a much better plan is to invest 5 minutes to get one variate, which permits us to run the Markov chain from an initial value that we know is *exactly in equilibrium*. And in fact, this is precisely the situation for self-avoiding walks: A noniterative method known as *dimerization* generates walks from the exact distribution P_N^*, not efficiently enough to produce many i.i.d. walks, but well enough to get a single initial value for the pivot algorithm, at least for $N \le 3{,}000$ (Madras and Sokal, 1988). Alternatively, dimerization allows us to use the method of Gelman and Rubin without having to use an approximate starting distribution.

In conclusion, useful rigorous bounds are as yet beyond reach in the great majority of simulations, and we must live in a world without guarantees. All that we can do is try to understand our model as best we can and have a variety of tools at our disposal to guard against certain dangers. There are no methods that work well in every situation; our choice of tools depends upon the particular problem.

ACKNOWLEDGMENT

This research was partially supported by the Natural Sciences and Engineering Research Council of Canada.