magnitudes of dispersion hyperparameters which are often unknown. As an example, consider the simple balanced, additive, two-way ANOVA model,

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \varepsilon_{ijk}, \quad i = 1, \ldots, I, \quad j = 1, \ldots, J, k = 1, \ldots, K,$$

where $\varepsilon_{ijk} \sim N(0, \sigma_e^2)$, $\alpha_i \sim N(0, \sigma_\alpha^2)$, $\beta_j \sim N(0, \sigma_\beta^2)$ and we place a flat prior on μ. Let $\eta_i = \mu + \alpha_i$ and $\rho_j = \mu + \beta_j$, so that η_i centers α_i, and ρ_j centers β_j. Then we can consider four possible parameterizations: (1) μ-α-β; (2) μ-η-β; (3) μ-α-ρ; (4) μ-η-ρ. Gelfand, Sahu and Carlin (1994b) discuss, under varying relative magnitudes for σ_e, σ_α and σ_β, which of these parametrizations is best in terms of mixing (using the diagnostic of Gelman and Rubin, 1992b), which affects the rate of convergence, and in terms of within-chain autocorrelation, which affects the variability of resultant ergodic averages used for inference.

Each of the four parametrizations produces a distinct Gibbs sampler. Following our earlier remarks, we create a fifth MCMC algorithm, which consists of cycling through these four parametrizations in sequence, running one complete single-site updating for each. To keep matters simple, we fix the values of the variance components, set $I = J = K = 5$ and use a sample of data generated from our assumed likelihood. Two interesting cases are shown in Figures 1 and 2, which display monitoring plots, estimated Gelman and Rubin scale reduction factors (labeled "G & R") and lag 1 sample autocorrelations (labeled "acf1") for five initially overdispersed parallel chains of 500 iterations each under the five algorithms. (To conserve space, we show results only for α_1, α_2, β_1, β_2 and μ.) The first figure sets $\sigma_e = 1$, $\sigma_\alpha = 10$ and $\sigma_\beta = 1$, while the second sets $\sigma_e = 1$, $\sigma_\alpha = 10$ and $\sigma_\beta = 20$. In Figure 1, the algorithm based on parametrization #2 (α's centered) is unequivocally the best of the first four, as predicted by the theoretical work in Gelfand, Sahu and Carlin (1994a, b). Matters are less clear in Figure 2, with each of the individual parametrizations having problems with one or more of the parameters. Notice that in both figures, for each component of the parameter space, the fifth algorithm achieves mixing which is as good as that of *any* of the first four. In fact, in Figure 2, the behavior of μ is satisfactory *only* for this composite algorithm. Note also, however, that the lag 1 autocorrelations for the fifth algorithm are fairly high, arising as weighted averages of those from the first four, so the corresponding samples must be used carefully in computing expectations via Monte Carlo integration.

Hence with regard to convergence, in using deterministic cycling through a medley of transition kernels, the analyst is able to achieve the benefits of each (and possibly more) without having to identify their relative quality. The computational effort in switching transition kernels in our examples only requires changing from one linear parametrization to another, and thus is quite efficient. Lastly, in situations where Metropolis steps are to be used within Gibbs samplers, thus necessitating proposal densities, adaptive adjustment of the dispersion of these proposals can be implemented concurrently with the deterministic switching of transition kernels.

ACKNOWLEDGMENTS

The work of the first-named author was supported in part by NSF Grant DMS-93-01316, while the work of the second-named author was supported in part by National Institute of Allergy and Infectious Diseases (NIAID) FIRST Award 1-R29-AI33466.

Comment

Charles J. Geyer

The authors are to be congratulated on this very nice paper, a tour de force in which all of various aspects of MCMC are completely mastered. I find myself largely in agreement with everything in this paper. What comments I have are not really disagreements but mere differences in emphasis.

SEPARATION OF CONCERNS

Let me begin my comments with a digression. Dijkstra (1976) in his seminal book on formal analysis of the correctness of computer programs introduces the notion of "separation of concerns." In computing we have "the mathematical concerns about correctness [of algorithms and programs implementing them] and the engineering concerns about execution [speed, memory requirements, user-friendliness, featurality]" and these should be kept separate. There is no point in worrying about speed

Charles J. Geyer is Assistant Professor, School of Statistics, University of Minnesota, Minneapolis, Minnesota 55455.

before one has a program that produces correct results.

In MCMC, of course, speed and correctness cannot be kept completely separate, since a sampler that is perfectly correct in the sense that the computer code correctly implements a Markov chain with a specified stationary distribution can mix so slowly that astronomical computing times would be required before the samples were representative of the stationary distribution. So a millionfold increase in speed might be the difference between a useful sampler and a useless one. A 10-fold or even a 100-fold increase will usually not make such a difference, however much it may affect ease of use. Thus speed and correctness are concerns that can usually but not always be separated.

This notion of "separation of concerns" can be extended beyond computing. We have scientific concerns about how well our statistical models and methods mesh with the scientific facts and theories that apply to the data at hand. We have concerns about the philosophy of statistics, whether to apply Bayesian, likelihood, decision-theoretic and so on theories and methods, and we have purely technical statistical concerns about details of procedures. These concerns should also be kept clearly separated, from each other and from the correctness and efficiency concerns, although they often are not.

The authors deserve high marks for dealing with scientific concerns. The analysis of gamma-camera images in Section 6 and the even more impressive analysis of SPECT images in Weir and Green (1994) fully incorporate the relevant physics. There seem to be no places where computational or mathematical statistical convenience is permitted to interfere with analyzing what is the scientifically correct model.

I am less happy about the separation of philosophical and computational concerns. Indeed, the first two words of the title "Bayesian computation" confuse the two. Although no one seems to have exactly said "MCMC is a strong reason to become Bayesian," many people seem to have picked up this message somewhere. Some of the statements in this paper could be interpreted to say something like this, whether or not this is what the authors intend. Although commonplace, it bears repeating that there is nothing Bayesian about MCMC. It is potentially useful anywhere in statistics where there are technical difficulties in computing probabilities, expectations and distributions. As this paper and many others show, MCMC has brought tremendous progress in Bayesian statistics. As is shown by Geyer and Thompson (1992) and other papers cited in the Introduction, to which I would like to add Gelfand and Carlin (1993) and Geyer and Møller (1994), similar progress has been made in likelihood inference. Complex dependence, missing data, conditional likelihood inference, inequality constraints on the parameters are all easily handled. It seems likely that this pattern would be repeated if MCMC were applied to other areas. Computational convenience is a poor substitute for philosophy.

I realize that Besag, Green, Higdon and Mengersen probably did not intend what they said to be read with the meaning I am criticizing. The point about Bayesian methods being most useful for ranking and selection, for example, is philosophical rather than computational. I say this only to forestall a very common reading of such language.

I am also somewhat unhappy with the emphasis on "full conditionals" as a basis for MCMC, explicitly stated in the first sentence of Section 2.3.1. This shows inadequate separation of concerns. Strictly speaking, conditional probability has nothing whatsoever to do with MCMC. It plays no role, for example, in a "random walk Metropolis" sampler. I realize the tremendous role that the local Markov property has played in spatial statistics, following Besag (1974), and in many other areas, such as graphical models. However, this is a philosophical concern relating to what distribution to simulate—what is the statistical model? It should have no effect on our computational concerns. We should start writing code with a clean slate. If Gibbs-like samplers using full conditionals are most efficient, well and good. If not, they should be avoided. Besag, Green, Higdon and Mengersen realize this, since they always avoid Gibbs whenever it becomes difficult. But why *any* preference for Gibbs?

CHOICE AMONG SAMPLING SCHEMES

Separation of concerns tells us to keep apart choices of sampling schemes made to avoid slow mixing or nonconvergence and choices that make minor improvements in efficiency. Mode jumping, mentioned in Section 4.1, is a remedy for slow mixing in some problems, but it requires a great deal of problem-specific knowledge. The Swenden–Wang algorithm and similar algorithms (grouped under the name "cluster algorithms" by the physicists) provide tremendous improvement over single-site updating but are not applicable to all problems. No cluster algorithms have been proposed for large graphical models in genetics and expert systems. Simulated tempering is a general solution potentially applicable to all problems. It may not provide convergence if the wrong form of "heating" is chosen, but if a good form is found, it will force convergence. Whenever there are worries about convergence, and no better problem-specific acceleration scheme comes to mind, simulated tempering should be tried.

Curiously, the existence of one possibly important acceleration scheme seems to be denied in the last paragraph of Section 2.4.5. It is not true that block updating is "rarely practicable," unless by "small and discrete" state space the authors refer to the state space at a single site. It is practicable, although difficult. Jensen, Kong and Kjærulff (1993) use block Gibbs sampling with very large blocks to sample a genetics problem on a pedigree with 20,000 individuals. The secret is that sampling the large blocks can only be done using so-called peeling methods (Cannings, Thompson and Skolnick, 1978; Lauritzen and Spiegelhalter, 1988). This entails much computational complexity and theory going far beyond ordinary Gibbs sampling, but it does work, at least for some large problems.

The other choices among sampling schemes discussed here seem to help only with efficiency, not with convergence. There the standard should be computing time necessary to get a specified Monte Carlo error (as used to select c in Section 6.2). Analogy with computer science says that there are two important strategies for improving efficiency: (1) radically change the algorithm and (2) speed up the inner loop. The first really applies more to methods such as mode jumping, cluster algorithms and simulated tempering. In regard to the second, a very good suggestion is the simple Hastings update with a uniform proposal used in Section 6.2. It may not be as efficient in terms of number of iterations for a fixed precision as more complicated samplers, but the inner loop runs as fast as possible. This may not always turn out to be the best, but it should always be one of the samplers under consideration.

From a somewhat different angle, it may be that another simple sampler should always be a strong candidate, at least for continuous state spaces. This is the single "random walk" Metropolis or Hastings update that updates all variables at once using a Gaussian proposal. The reason here is not so much computational efficiency (although because of its extreme simplicity it may win here too), but because of its theoretical simplicity. Roberts and Tweedie (1994b) give a geometric ergodicity theorem for this algorithm that depends only on the stationary distribution having exponential tails and asymptotically round contours. It does not depend in any way on the proposal distribution. Such a result seems unlikely for more complicated samplers composed of many elementary update steps. Even if the complicated samplers are slightly more efficient, something rarely investigated, the theoretical simplicity obtained when all variables are updated simultaneously may be worth some loss of efficiency. I am not sure I agree with this point myself, but it is worth thinking about.

That having been said, I should like to propose a reversible scan to add to those in Section 2.4.2. Choose a variable uniformly at random, excluding the one last updated. Then scan forward or backward in numerical order, choosing the directions with equal probabilities. This consumes only one or two uniform random variates per scan, has little other overhead, never updates the same variable twice in succession, updates each variable once per scan and is reversible.

SENSITIVITY ANALYSIS

I should like to point out Geyer (1991b) as another independent proposal of sensitivity analysis via importance sampling besides those of Besag (1992) and Smith (1992) mentioned in Appendix 3. Of course the real credit goes to those who actually implement the proposals, as Besag, Green, Higdon and Mengersen have done. Some other nice work along the same lines has been done by Doss and Narasimhan (1994).