matrices based on cross-validation took roughly 12 min, with the main computational burden being the repeated evaluation and inversions of the 550×550 matrices $\mathbf{G}(\boldsymbol{\Sigma})$ and $\mathbf{G}(\boldsymbol{\Lambda})$. Due to the increased computation time, finding the "optimal" $\boldsymbol{\Sigma}$ and $\boldsymbol{\Lambda}$ matrices in each step of the sequential algorithm appeared no longer practically feasible. When adding 450 additional function evaluations (based on the components selected by *iterLap*), computations were considerably slowed down: inversion of the involved 1000×1000 matrix took six times longer compared with the 550×550 matrix, and one might expect an increase in the total computation time by a similar factor. Note that this is roughly in agreement with the fact that matrix inversion is roughly a $O(n^3)$ process (if n denotes the size of the matrix). From these considerations, it becomes clear that the interpolation technique, as presented now, will become quickly infeasible when a large number of function evaluations is required to obtain an adequate approximation, as matrix inversions are the main factor driving computation time.

The essential difference between the two methods is thus in their usage of target density evaluations. If it is cheap to perform a large number of evaluations, it appears *iterLap* (and also well-chosen and efficiently implemented MCMC algorithms) will outperform the interpolation technique, because the interpolation technique will itself get computationally intensive due to the required matrix inversions. If evaluations of the target are extremely expensive, so that only few evaluations are possible anyway, the interpolation technique seems to make better usage of the evaluations performed.

3. FINAL REMARKS

The main computational bottleneck of the proposed procedure is the need to evaluate and invert large-dimensional matrices repeatedly (as in all kriging-type interpolation approaches). This can get quite computationally expensive, but might pay off, for example, when the posterior density is extremely time-consuming to evaluate or when it is of great interest to obtain a high-accuracy approximation of the posterior density itself. However, an improvement of the procedure in this regard seems possible and would certainly be of high interest.

In summary, I would like to congratulate Professor Joseph for an interesting article that provides an innovative approach on how to apply kriging-type techniques for interpolation of positive functions, and I hope Professor Joseph's article stimulates further research in the application of these methods for Bayesian computational problems. I would like to end my discussion with the wish that an implementation of the method will be made publicly available, with concrete recommendations for default or automated choices that have been tested on a variety of example posteriors. The chance that that the methods gets more widely adopted by applied statisticians will be increased if an efficient and easy-to-use implementation is provided.

ADDITIONAL REFERENCES

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2003), *Bayesian Data Analysis* (2nd ed.), London: Chapman and Hall. [225]

Kracker, H. (2011), "Modeling and Calibration of Computer Models with Applications to Sheet Metal Forming," Ph.D. dissertation, Department of Statistics, Dortmund University of Technology. [226]

Comment: DoIt and Do It Well

Tirthankar DASGUPTA and Xiao-Li MENG

Department of Statistics
Harvard University
Cambridge, MA 02138
(*dasgupta@stat.harvard.edu*;
meng@stat.harvard.edu)

1. IS DOIT JUST FOR QUASI-MONTE CARLO?

The key idea in this article is to approximate a complex posterior density by a weighted average of normal densities, where the weights are chosen by fitting a kriging model that interpolates the unnormalized posterior. The accuracy of approximation depends on the choice of evaluation points, and can be improved by augmenting additional points. The method therefore is a generalization of the standard Laplace approximation based on a single design point, namely a posterior mode. Mathematically speaking, the proposed DoIt is a case of quasi-Monte Carlo (QMC), which has an extensive literature on how to strategically place (deterministic) design points for efficient numerical integration; for example, see Niederreiter (1978, 1992), Caflisch (1998), L'Ecuyer and Owen (2009), Dick and Pillichshammer (2010), and particularly, Stein (1987) and Owen (1998) regarding the use of Latin hypercube design—as used in the article for the initial space-filling design—for QMC.

A well-known and critical challenge for QMC is the curse of dimension. DoIt, when used directly for approximating an

TECHNOMETRICS, AUGUST 2012, VOL. 54, NO. 3
DOI: 10.1080/00401706.2012.697248

integration, faces the same challenge, as discussed in Section 5 of the article in the context of hierarchical models. A known strategy for making a QMC method as generally useful as a genuine Monte Carlo (MC) is to reintroduce randomization into the QMC method (i.e., the so-called "randomized QMC") and, more promisingly, to combine it with an MC method, as discussed and explored in Owen (1998). However, despite the extensive literature on both QMC and MC and their shared overall goal, the overlap of the two literatures is surprisingly small, as noted in Meng (2005). We therefore thank Joseph for promoting the use of experiment design principles and techniques in Bayesian computation, with a method that has good potential to form a basis for an effective hybrid MC because of its clear statistical construction. In particular, the normal mixture nature of DoIt makes it a rather convenient and potentially effective proposal for a Metropolis–Hasting algorithm, especially if it can be extended further to the t-mixture type of approximations as investigated by West (1993). Even if there is no need to use the DoIt approximation as a proposal, it can still provide an independent (partial) validation of a Markov chain Monte Carlo (MCMC) method.

With our goal of exploring the possibility of a happy marriage between QMC and MCMC, we touch upon two main issues in this discussion. First, as pointed out by Joseph in the last two paragraphs of his section 1, a line of research in Bayesian computation from computationally expensive black-box posterior distributions is based on the idea of approximating the logarithm of the posterior distribution by a Gaussian process (GP) model, and using the GP-based surrogate model as an approximate target density for MCMC or hybrid-MCMC sampling (Rasmussen 2003; Fielding et al. 2011). A comparison of the proposed DoIt algorithm with the GP-based approach, which will be referred to as GP-MCMC henceforth, is presented in Section 2 of the article. We feel, however, that this comparison might have created an unintended impression that the effectiveness of GP-MCMC, *as a general strategy*, is rather limited. We therefore probe this comparison a little further in Section 1 of our discussion. Next, we address an important aspect of the sequential design discussed in section 3.2 of Joseph's article: judging the accuracy of approximation. We propose a Hellinger distance-based criterion for judging the accuracy of approximation in GP-MCMC and conduct a preliminary exploration with the example used to compare GP-MCMC and DoIt.

2. CAN GP-MCMC DO WELL WITH FEWER EVALUATIONS?

In section 2 of Joseph's article, DoIt and a particular GP-MCMC algorithm are used to study the following two-dimensional posterior density with banana-shaped contours (Haario, Saksman, and Taaminen 2001):

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \phi\big(\big(\theta_1, \theta_2 + 0.03\theta_1^2 - 3\big)'; (0,0)',\ \mathrm{diag}\{100, 1\}\big).$$

As observed from Joseph's figure 9(a), the DoIt approximation obtained from a 100-run maximin Latin hypercube design (MmLHD) chosen from the region $[-20, 20] \times [-10, 5]$ does not give a good fit to the exact distribution. However, after adding 75 more points, the DoIt approximation captures the support and the shape of the distribution quite well. For the hybrid MCMC algorithm proposed by Fielding et al. (2011), the same 100-run MmLHD is used as the initial design, and 500 and 1500 samples are generated from the exploratory phase and the sampling phase, respectively. Although the sampling is very good, as evident from Joseph's figure 11(b), it is reported to have taken almost 90 min as compared with 3 min taken by DoIt. Consequently, it is concluded that although both methods perform well, GP-MCMC is computationally much more expensive than DoIt.

The comparison raises two important questions. First, is the complex hybrid MCMC algorithm with parallel tempering proposed by Fielding et al. (2011) really needed for this two-dimensional example? A simpler MCMC algorithm that uses the basic idea of sampling from a GP-based surrogate may be appropriate. Second, assuming that by a "sample" in the exploratory phase, Joseph means one representative point (typically the last) point of an MCMC chain, is it necessary to generate a total of 500 samples (which also means potentially prohibitively large 500 evaluations of the expensive posterior) in the exploratory phase to adequately capture the contours of the distribution? This also raises a related question: what should be a reasonable guideline to judge whether the surrogate GP model approximates the posterior distribution well? We will discuss the second point elaborately in the next section.

At this point, we briefly introduce a rudimentary random-walk MCMC algorithm using the GP approximation. Let $\mathcal{D}$ denote the exploration region (design space) and $\pi(\boldsymbol{x})$ the unnormalized posterior density of interest. Let π^* denote the corresponding normalized density, and assume that the design space is an adequate approximation to its support, that is,

$$\pi^*(\mathcal{D}^c) \approx 0, \tag{1}$$

where $\mathcal{D}^c$ denotes the complementary set of $\mathcal{D}$. As in DoIt, we choose an initial space-filling (e.g., MmLHD) design of N points in $\mathcal{D}$. Let $\hat{y}(\boldsymbol{x})$ denote the ordinary Kriging predictor (Santner, Williams, and Notz 2003) of $\log \pi(\boldsymbol{x})$, and $s^2(\boldsymbol{x})$ denote the mean squared error (MSE) of the predictor. During the exploratory phase, we use a random-walk Metropolis algorithm to sample from the following target distribution:

$$p(\boldsymbol{x}) \propto \begin{cases} \exp(\hat{y}(x) + s(x)), & \boldsymbol{x} \in \mathcal{D} \\ \exp(\hat{y}(x)), & \boldsymbol{x} \in \mathcal{D}^c \end{cases} \tag{2}$$

In the sampling phase, we sample from the target distribution proportional to $\exp(\hat{y}(x))$, as proposed by Fielding et al. (2011). We emphasize here that it is wise to allow our sampling algorithm to go beyond the design space $\mathcal{D}$ no matter how carefully it was chosen in the first place.

Thus, denoting the current state at the $(t-1)$th iteration by $\boldsymbol{x}^{(t-1)}$, we generate the proposal state

$$\boldsymbol{x}' = \boldsymbol{x}^{(t-1)} + \epsilon,$$

where $\epsilon \sim N((0,0)', \sigma^2 \mathrm{diag}(1,1))$ with $\sigma^2 = 1$. The new state is obtained as

$$\boldsymbol{x}^{(t)} = \begin{cases} \boldsymbol{x}' & \text{if } r^{(t)} \le \min\{1, p(\boldsymbol{x}')/p(\boldsymbol{x}^{t-1})\} \\ \boldsymbol{x}^{(t-1)} & \text{otherwise} \end{cases},$$

where $r^{(t)}$ is a random sample drawn from Uniform[0, 1].

To see how well this algorithm works, we choose an MmLHD design with $N = 30$ points and then sequentially generate 20

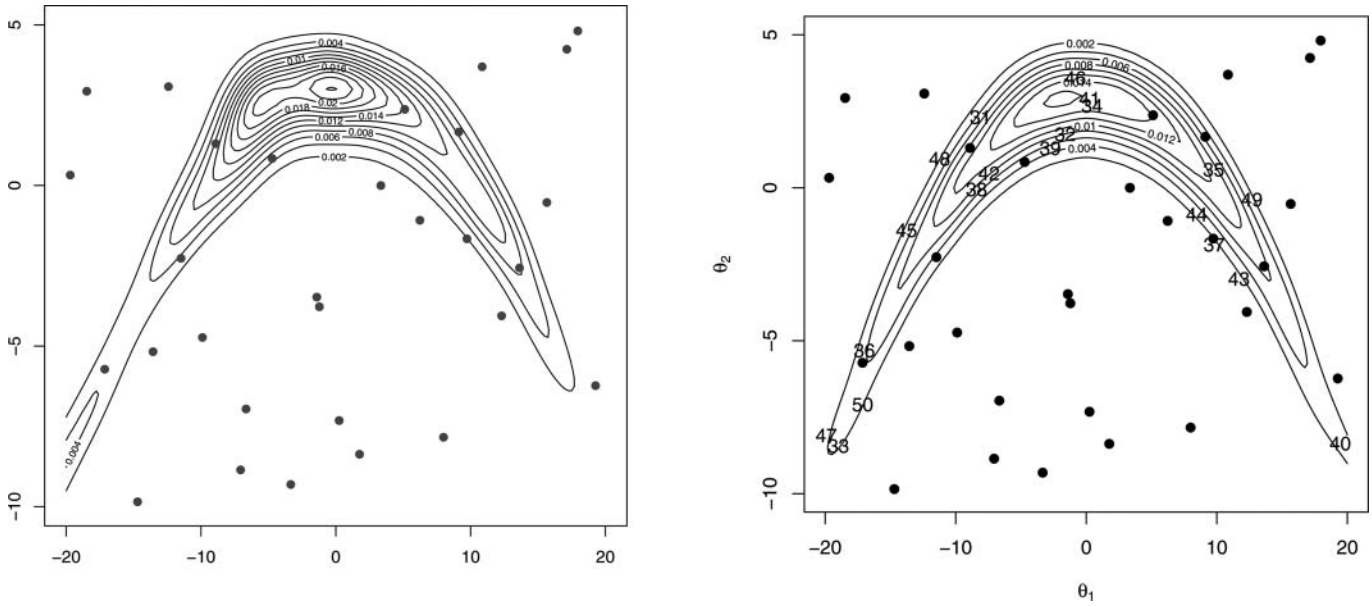

Figure 1. Contours approximated from the initial 30-run design (left panel) and after adding 20 points sequentially (right panel). The online version of this figure is in color.

additional points from the exploratory phase of the aforementioned algorithm, where each point is the last point of an MCMC chain of length 2000. The left panel in Figure 1 shows the contour plot generated from the kriging predictor obtained from the 30 initial design points, and the right panel shows the contour plot after sequentially adding 20 points from the exploratory phase of the algorithm. In both the panels, the dots represent the initial 30 points and the numbers in the right panel indicate the order of points generated sequentially. The left panel in Figure 2 shows an MCMC chain of 10,000 points drawn from the sampling phase using the surrogate density obtained from the 50 sampled points.

We observe that the initial 30-point design approximates the contour pretty well—in fact, substantially better than the DoIt approximation based on 100 design points. The approximation appears to be very good after adding only 20 points using our rudimentary Metropolis algorithm based on the GP approximation. The time taken for this entire task was about 7 min, most of which (about 6 min) was spent on adding the 20 points during the exploratory phase. Generating 10,000 points during the sampling phase barely took 1 min. Thus, the total time taken by our GP-MCMC to approximate the posterior as good as one obtained by using DoIt was found to be more (7 min vs. 3 min, as reported by Joesph). But our GP-MCMC required far less evaluations (50 vs. 175), which can be a substantial advantage for computationally expensive functions. In fact, if one follows Joseph's guideline of selecting $50d$ initial points (where d denotes the dimension), then a 100-run initial design provides an excellent GP

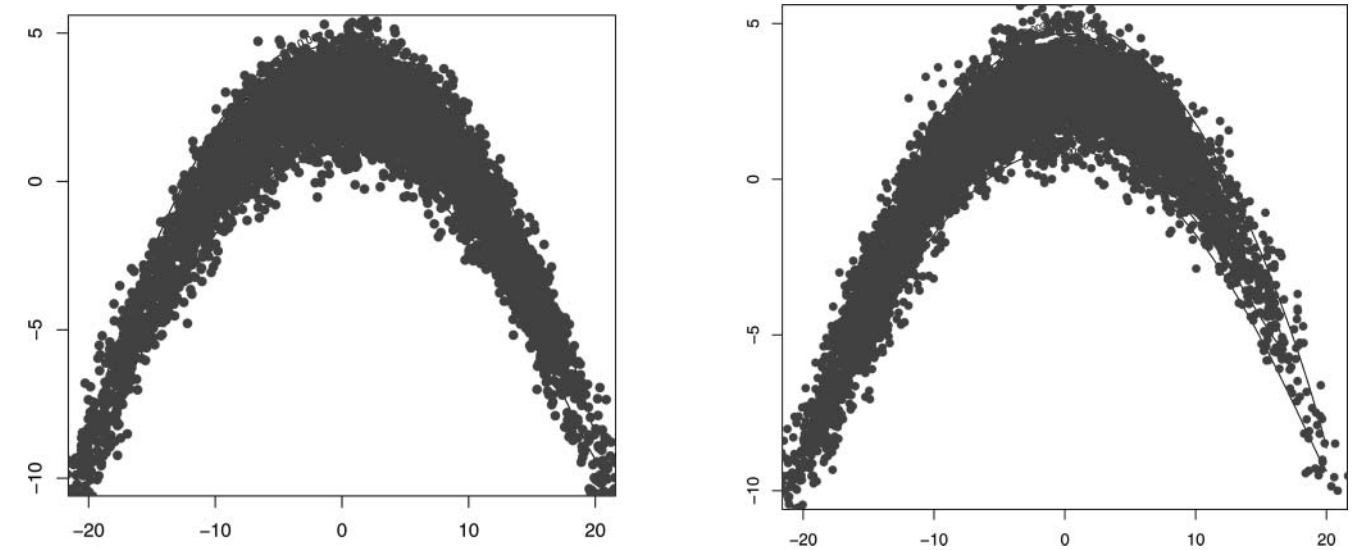

Figure 2. A total of 10,000 points generated using 50 (= 30 + 20) design points (left panel) and 100 (= 100 + 0) design points (right panel). The online version of this figure is in color.

approximation of the density, and one can immediately proceed to the sampling phase, completely bypassing the exploratory phase. An MCMC chain of 10,000 points generated from the sampling phase using the GP approximation based on 100 initial points is shown in the right panel in Figure 2. This entire task, starting from the generation of the 100 design points to the generation of 10,000 samples took about 1 min, about one-third of the reported time taken by DoIt. It is worth noting that our MCMC scheme here is most inefficient, being a simple random walk without any tuning of, for example, the variance of a step size ϵ.

3. IS THE APPROXIMATION ADEQUATE?

The foregoing example reinforces the importance of the question raised in Section 1: in GP-MCMC, when should we switch to the sampling phase from the exploratory phase? In other words, when do we have enough confidence in the surrogate model as an emulator of the true posterior? To the best of our knowledge, this particular aspect has not been adequately addressed in the literature. Clearly, to make a decision, we need a criterion that is able to judge the "goodness of fit" of the surrogate density. Establishing such a criterion may also be helpful to judge when a DoIt approximation of a computationally expensive posterior is good enough.

We now propose a criterion based on the Hellinger distance between two densities f and g, which is defined as

$$H(f,g)=\left[\frac{1}{2}\int\left(\sqrt{f(\boldsymbol{x})}-\sqrt{g(\boldsymbol{x})}\right)^2 d\boldsymbol{x}\right]^{1/2}. \tag{3}$$

It is well known that $H(f,g)$ defined by (3) is related to the Bhattacharya coefficient $BC(f,g)$ given by $\int\sqrt{f(\boldsymbol{x})g(\boldsymbol{x})}d\boldsymbol{x}$ through the following identity:

$$H(f,g)=\sqrt{1-BC(f,g)}. \tag{4}$$

In the current problem, the two densities that need to be compared are the true density π^* proportional to π, and our sampling target density p^* proportional to p, where p is defined by (2). Let their supports be, respectively, $\mathcal{S}_\pi$ and $\mathcal{S}_p$. Then, the Bhattacharya coefficient between π^* and p^* can be written as

$$\begin{aligned}BC(\pi^*,p^*)&=\frac{\int_{\mathcal{S}_\pi\cap\mathcal{S}_p}\sqrt{\pi(\boldsymbol{x})p(\boldsymbol{x})}d\boldsymbol{x}}{\sqrt{\int_{\mathcal{S}_\pi}\pi(\boldsymbol{x})d\boldsymbol{x}}\sqrt{\int_{\mathcal{S}_p}p(\boldsymbol{x})d\boldsymbol{x}}}\\&=\frac{\int_{\mathcal{S}_p}\sqrt{[\pi(\boldsymbol{x})/p(\boldsymbol{x})]}p^*(\boldsymbol{x})d\boldsymbol{x}}{\sqrt{\int_{\mathcal{S}_p}[\pi(\boldsymbol{x})/p(\boldsymbol{x})]p^*(\boldsymbol{x})d\boldsymbol{x}+\Delta}},\end{aligned} \tag{5}$$

where

$$\Delta=\frac{\int_{\mathcal{S}_\pi\cap\mathcal{S}_p^c}\pi(\boldsymbol{x})d\boldsymbol{x}}{\int_{\mathcal{S}_p}p(\boldsymbol{x})d\boldsymbol{x}}.$$

Consequently, when $\mathcal{S}_\pi\subseteq\mathcal{S}_p$, which implies $\Delta=0$, the Bhattacharya coefficient can be easily estimated—as proposed by Meng and Wong (1996)—by

$$\hat{BC}=\frac{\frac{1}{k}\sum_{j=1}^k\sqrt{\zeta_j}}{\sqrt{\frac{1}{k}\sum_{j=1}^k\zeta_j}}, \tag{6}$$

where

$$\zeta_j=\pi(\omega_j)/p(\omega_j), \tag{7}$$

and $\omega_1,\ldots,\omega_k$ are k draws from p. A beauty of the estimator in (6) is that it is numerically constrained to be inside the unit interval, just as its estimand (5). Of course, we need to be mindful that its computation involves k additional evaluations of the posterior π, so we often will keep k relatively small (compared with the overall number of draws) if evaluating π is expensive. [Meng and Wong (1996) adopted the Hellinger distance because the variance of their bridge sampling estimator is bounded both above and below by simple functions of the Hellinger distance between the two densities for which the ratio of their normalizing constants is the estimand.]

To apply this method, recall that, in the exploratory phase of our algorithm applied to the banana-shaped function in Section 1, we drew 2000 MCMC samples in each iteration and chose the last sample as our next design point. Because these 2000 points were drawn from p, a subset of these points could be used to compute $\hat{BC}$ from (6). Figure 3 shows a plot of the estimated Bhattacharya coefficients for 20 successive iterations during the exploratory phase. The coefficients were estimated using $k=20$ points randomly chosen from 2000 MCMC draws in each iteration. We observe that all the estimated coefficients are greater than 0.9 (suggesting that the approximation from the initial 30-run design is reasonable) and appear to converge to 1.0 after about 16 iterations (the solid horizontal line is 0.99).

At this point, it might be tempting to suggest a switching rule such as: "switch to the sampling phase *if* m consecutive estimated Bhattacharya coefficients are above a certain threshold δ." Cautions are needed, however, for implementing such a rule. The obvious one is that we need to take into account the variability in (6) when we compare it to a threshold. This can be achieved by using a lower confidence bound, say $\hat{BC}-2\hat{\tau}_k$, instead of $\hat{BC}$, in making the comparison. Here, $\hat{\tau}_k$ is an estimate of the standard error of $\hat{BC}$, which can be obtained in various ways, including direct MC replications using existing draws (e.g., a part of the 2000 draws in our example) and taking

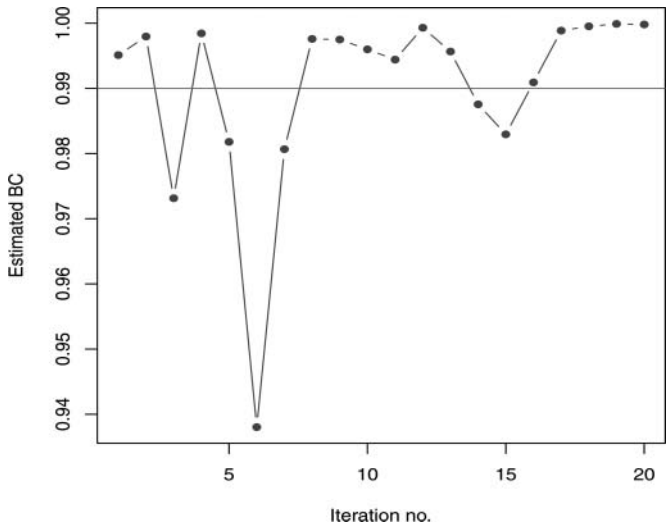

Figure 3. Plot of the Bhattacharya coefficient. The online version of this figure is in color.

advantage of theoretical formulas such as Equation (8.7) in Meng and Wong (1996). Details will be reported in a future work.

The more difficult one is to deal with the positive bias in (6) when our sampling state space $\mathcal{S}_p$ fails to cover the actual state space $\mathcal{S}_\pi$. Such a failure is likely in practice even when in theory we design $\mathcal{S}_p = \mathcal{S}_\pi$ (e.g., as in our random-walk algorithm), because it reflects the very problem we try to resolve, namely our MCMC algorithm may fail to explore all the regions with appreciable masses under the desired π; see Meng and Schilling (1996) for a numerical illustration of this aspect. Such an overestimation, if not taken into account appropriately, would then lead us to prematurely make the switch with a higher probability than we plan.

This bias issue is hard to deal with precisely because it is not possible to use samples inside $\mathcal{S}_p$ to explore masses outside $\mathcal{S}_p$ under π, unless we use the knowledge of how masses outside $\mathcal{S}_p$ are related to those from inside of it. Such knowledge, for example, may provide us with a convenient upper bound on the relative overestimation, which then would allow us to adjust the threshold δ to prevent (statistically) the premature switching. Clearly, a thorough investigation of such issues is needed, and so is an in-depth investigation of the effects of k, m, and δ (in the switching rule) on the computation time and cost under different situations. Furthermore, in our example, we required $20 \times 20 = 400$ evaluations of the posterior π simply to judge the accuracy of approximation. To circumvent this problem, one can, for example, consider estimating the Bhattacharya coefficient at an interval of several iterations during the exploratory phase.

There are, of course, multiple ways to improve both the computational efficiency and the statistical efficiency of such algorithms, leading to a good number of interesting and useful research projects. We therefore want to thank Joseph again, not only for proposing DoIt, but also for inspiring us to search for those hybrid MCMC methods that will do well.

ACKNOWLEDGMENTS

The authors thank the editor for giving them the opportunity of writing this discussion. This work was partially supported by grants from the National Science Foundation.

ADDITIONAL REFERENCES

Caflisch, R. E. (1998), "Monte Carlo and Quasi-Monte Carlo Methods," *Acta Numerica*, 7, 1–49. [227]

Dick, J., and Pillichshammer, F. (2010), *Digital Nets and Sequences. Discrepancy Theory and Quasi-Monte Carlo Integration*, Cambridge: Cambridge University Press. [227]

L'Ecuyer, P., and Owen, A. B.(eds.) (2009), "Monte Carlo and Quasi-Monte Carlo Methods," in *Proceedings of MCQMC 2008*, Berlin: Springer-Verlag. [227]

Meng, X.-L. (2005), "Comment: Computation, Survey and Inference," *Statistical Science*, 20, 21–28. [227]

Meng, X.-L., and Schilling, S. (1996), "Fitting Full-Information Item Factor Models and an Empirical Investigation of Bridge Sampling," *Journal of the American Statistical Association*, 91, 1254–1267. [231]

Meng, X.-L., and Wong, W. H. (1996), "Simulating Ratios of Normalizing Constants via a Simple Identity: A Theoretical Exploration," *Statistica Sinica*, 6, 831–860. [230]

Niederreiter, H. G. (1978), "Quasi-Monte Carlo Methods and Pseudo-Random Numbers," *Bulletin of the American Mathematical Society*, 84, 957–1041. [227]

——— (1992), *Random Number Generation and Quasi-Monte Carlo Methods*, Philadelphia, PA: SIAM (Society for Industrial and Applied Mathematics). [227]

Owen, A. B. (1998), "Monte Carlo Extension of Quasi-Monte Carlo," in *1998 Winter Simulation Conference Proceedings*, eds. D. J. Medieiros, E. F. Watson, M. Manivannan, and J. Carson, pp. 571–577. [227]

Stein, M. (1987), "Large Sample Properties of Simulations Using Latin Hypercube Sampling," *Technometrics*, 29, 143–151. [227]

West, M. (1993), "Approximate Posterior Distributions by Mixture," *Journal of the Royal Statistical Society,* Series B, 55, 409–422. [227]

Comment

Herbert K. H. Lee

Department of Applied Mathematics and Statistics
University of California
Santa Cruz, CA 95060
(*herbie@soe.ucsc.edu*)

1. OVERVIEW

The innovative article by Joseph (2012) incorporates some elements from the computer modeling literature and further develops them for the all-important question in Bayesian statistics of approximating a complex posterior distribution. Markov chain Monte Carlo (MCMC) has made all posterior distributions accessible in theory, but the amount of computing time needed to get a good estimate can easily exceed available resources. The article provides a promising new approach when the posterior is expensive to evaluate.

TECHNOMETRICS, AUGUST 2012, VOL. 54, NO. 3
DOI: 10.1080/00401706.2012.697245