Bayesian Computation Using Design of Experiments-Based Interpolation Technique

V. Roshan JOSEPH

H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332
(*roshan@gatech.edu*)

In this article, a new deterministic approximation method for Bayesian computation, known as design of experiments-based interpolation technique (DoIt), is proposed. The method works by sampling points from the parameter space using an experimental design and then fitting a kriging model to interpolate the unnormalized posterior. The approximated posterior density is a weighted average of normal densities, and therefore, most of the posterior quantities can be easily computed. DoIt is a general computing technique that is easy to implement and can be applied to many complex Bayesian problems. Moreover, it does not suffer from the curse of dimensionality as much as some quadrature methods. It can work using fewer posterior evaluations, which is a great advantage over the Monte Carlo and Markov chain Monte Carlo methods, especially when dealing with computationally expensive posteriors. This article has supplementary material that is available online.

KEY WORDS: Computer experiments; Experimental design; Kriging; Laplace's method.

1. INTRODUCTION

Computation of posterior quantities is a fundamental problem in the application of Bayesian methods. Earlier work in this field includes approximating the posterior distribution by a normal distribution using the posterior mode (also known as Laplace's approximation) and the use of numerical integration tools, such as Gaussian quadrature; for example, see Naylor and Smith (1982) and Tierney and Kadane (1986). These methods are considered inadequate for high-dimensional and complex Bayesian models. Monte Carlo (MC) methods and the advent of Markov chain Monte Carlo (MCMC) methods have revolutionized the field during the last two decades. The amount of literature on MC/MCMC methods is vast, where some of the landmark articles include Metropolis et al. (1953), Hastings (1970), Geman and Geman (1984), Tanner and Wong (1987), and Gelfand and Smith (1990). A recent review of the methods can be found in Brooks et al. (2011). These methods suffer less from the curse of dimensionality and can obtain the results with arbitrary precision. However, the convergence of the methods and the high computational cost when dealing with computationally expensive posteriors are still a concern.

In this work, a new deterministic method for approximating continuous posterior distributions using normal-like basis functions is introduced. The method draws ideas from the design and analysis of computer experiments (see Santner, Williams, and Notz 2003) and builds on the earlier work of O'Hagan (1991), Kennedy (1998), and Rasmussen and Ghahramani (2003). It is different from the other deterministic approximation methods such as variational Bayes (VB) (e.g., see Bishop 2006), expectation propagation (EP) (Minka 2001), and integrated nested Laplace approximation (INLA) (Rue, Martino, and Chopin 2009) in that it is capable of computing the quantities at a desired accuracy. The proposed method is general and easy to implement, and can be applied to many Bayesian problems. It is shown that the method does not suffer from the curse of dimensionality to the same extent as lattice-based quadrature methods. With a proper use of experimental design techniques, the method can be made to work faster than the MC/MCMC methods, which is quite advantageous in dealing with computationally expensive posteriors or when the posterior needs to be evaluated many times within external algorithms.

The remainder of this article is organized as follows. In Section 2, this new method, called *design of experiments-based interpolation technique* (DoIt), is explained. Experimental designs that are critical for the success of DoIt are discussed in Section 3. Applications to hierarchical models and computationally expensive posteriors are discussed in Section 4, and Section 5 concludes with some remarks and future research directions.

2. DESIGN OF EXPERIMENTS-BASED INTERPOLATION TECHNIQUE

In this section, first, the basic idea of DoIt is introduced. It is presented as an extension of the Laplace approximation, which requires knowledge of the posterior mode. The method is then generalized in Section 2.2 to deal with the case of an unknown posterior mode and nondifferentiable densities. A limitation of the DoIt, as presented, is that estimated densities could be negative. After illustrating this issue, further enhancements are developed in Section 2.3. The resultant method is then used in Section 2.4 for quick approximation of integrals, and comparisons with other posterior approximation methods are made in Section 2.5.

It is worthwhile to mention that although the focus of this article is on approximating posterior densities, DoIt can also

DOI: 10.1080/00401706.2012.680399

be used for approximating arbitrary multivariate densities of continuous random variables.

2.1 The Basic Idea: Weighted Normal Approximation

Let $\boldsymbol{y} = (y_1, \ldots, y_n)'$ be the data generated from a sampling model $p(\boldsymbol{y}|\boldsymbol{\theta})$, where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)'$ denotes the unknown parameters. Assume that after suitable transformation, $\boldsymbol{\theta} \in \mathbb{R}^d$, and let $p(\boldsymbol{\theta})$ be its prior distribution. Let $h(\boldsymbol{\theta}) \propto p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$ be the unnormalized posterior. Then, by the Taylor series expansion of $\log(h(\boldsymbol{\theta}))$ at the posterior mode $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} h(\boldsymbol{\theta})$, one obtains

$$h(\boldsymbol{\theta}) \approx h(\hat{\boldsymbol{\theta}}) \exp\left\{-\frac{1}{2}\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\right\}, \tag{1}$$

where $\boldsymbol{\Sigma} = [-\nabla^2 \log(h(\hat{\boldsymbol{\theta}}))]^{-1}$ is the inverse of the Hessian matrix of $-\log(h(\boldsymbol{\theta}))$ evaluated at the posterior mode. This leads to the Laplace approximation of the posterior distribution, given by $\boldsymbol{\theta}|\boldsymbol{y} \sim^a N(\hat{\boldsymbol{\theta}}, \boldsymbol{\Sigma})$. This can be a reasonable approximation when the posterior is symmetric and unimodal. Below is proposed a method to improve this approximation.

Let $\phi(\boldsymbol{\theta}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denote the normal density function and $g(\boldsymbol{\theta}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \exp\{-\frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta} - \boldsymbol{\mu})\}$, the unnormalized density. Consider a generalization of (1) as follows:

$$h(\boldsymbol{\theta}) \approx \sum_{i=1}^{m} c_i g(\boldsymbol{\theta}; \boldsymbol{\nu}_i, \boldsymbol{\Sigma}), \tag{2}$$

where $\boldsymbol{D} = \{\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_m\}$ is a set of evaluation points chosen based on an experimental design and $\boldsymbol{c} = (c_1, \ldots, c_m)'$ is a vector of real-valued constants. If the posterior mode is known, then without loss of generality, one can take $\boldsymbol{\nu}_1 = \hat{\boldsymbol{\theta}}$, and therefore, for $m = 1$, Equation (2) reduces to (1), with $c_1 = h(\hat{\boldsymbol{\theta}})$. The expansion in (2) is similar to a simple kriging predictor or a radial basis function predictor with a Gaussian correlation function (see Santner, Williams, and Notz 2003, pp. 63–64, or Rasmussen and Williams 2006, p. 17). Kriging is widely applied in computer experiments for approximating expensive deterministic functions, which is why this method can be expected to work well in approximating expensive posteriors. The unknown constants c_i's are obtained as follows. Evaluate $h(\boldsymbol{\theta})$ at the m points in $\boldsymbol{D}$, giving rise to $\boldsymbol{h} = (h_1, \ldots, h_m)'$, where $h_i = h(\boldsymbol{\nu}_i)$. Now, $\boldsymbol{c}$ can be chosen so that the prediction from the right side of (2) at the points in $\boldsymbol{D}$ is as close to $\boldsymbol{h}$ as possible. In fact, it is possible to obtain interpolation. Then, one must have $\boldsymbol{G}\boldsymbol{c} = \boldsymbol{h}$, where $\boldsymbol{G}$ is an $m \times m$ matrix with ijth element $g(\boldsymbol{\nu}_i; \boldsymbol{\nu}_j, \boldsymbol{\Sigma})$. Since $g(\boldsymbol{\theta}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a positive definite function (Santner, Williams, and Notz 2003, sec. 2.3.3), $\boldsymbol{G}^{-1}$ exists, provided $\boldsymbol{\nu}_i \neq \boldsymbol{\nu}_j$ for all i and j. Thus, one obtains the unique solution $\tilde{\boldsymbol{c}} = \boldsymbol{G}^{-1}\boldsymbol{h}$. Let $\boldsymbol{g}(\boldsymbol{\theta}) = (g(\boldsymbol{\theta}; \boldsymbol{\nu}_1, \boldsymbol{\Sigma}), \ldots, g(\boldsymbol{\theta}; \boldsymbol{\nu}_m, \boldsymbol{\Sigma}))'$. Then,

$$\tilde{h}(\boldsymbol{\theta}) = \tilde{\boldsymbol{c}}'\boldsymbol{g}(\boldsymbol{\theta}). \tag{3}$$

Integrating from $-\infty$ to ∞ with respect to each θ_i, one obtains the marginal likelihood

$$\int \tilde{h}(\boldsymbol{\theta})d\boldsymbol{\theta} = \tilde{\boldsymbol{c}}' \int \boldsymbol{g}(\boldsymbol{\theta})d\boldsymbol{\theta} = (2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}\tilde{\boldsymbol{c}}'\boldsymbol{1}, \tag{4}$$

where $\boldsymbol{1}$ is a column of 1's having length m. Thus, an approximation to the posterior distribution is given by

$$\tilde{p}(\boldsymbol{\theta}|\boldsymbol{y}) \approx \frac{\tilde{\boldsymbol{c}}'\boldsymbol{g}(\boldsymbol{\theta})}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}\tilde{\boldsymbol{c}}'\boldsymbol{1}} = \frac{\tilde{\boldsymbol{c}}'\boldsymbol{\phi}(\boldsymbol{\theta})}{\tilde{\boldsymbol{c}}'\boldsymbol{1}}, \tag{5}$$

where $\boldsymbol{\phi}(\boldsymbol{\theta}) = \boldsymbol{g}(\boldsymbol{\theta})/((2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}) = (\phi(\boldsymbol{\theta}; \boldsymbol{\nu}_1, \boldsymbol{\Sigma}), \ldots, \phi(\boldsymbol{\theta}; \boldsymbol{\nu}_m, \boldsymbol{\Sigma}))'$. Thus, the approximation is a weighted average of the normal density functions evaluated at $\boldsymbol{D}$. Note, however, that this is not a mixture normal approximation because the $\tilde{c}_i$'s can be negative. The fact that $\tilde{c}_i$'s can become negative immediately raises the concern that the approximation $\tilde{p}(\boldsymbol{\theta}|\boldsymbol{y})$ itself can be negative. This concern is genuine, but as will be seen later in Section 2.3, the error is not too serious and one can develop methods to overcome it.

Consider the following illustrative example. Suppose a single data value $y = 0$ is observed from Poisson(θ). Under the improper prior distribution, $p(\theta) \propto 1$, the posterior distribution is an exponential distribution (with rate parameter 1). Because θ is nonnegative, first, transform it to $\gamma = \log(\theta)$. Now, one obtains $\hat{\gamma} = 0$ and $\boldsymbol{\Sigma} = \sigma^2 = 1$. Transforming back, the Laplace approximation is given by $\phi(\log(\theta); 0, 1)/\theta$, which is shown in Figure 1. One can see that it is a poor approximation to the exact posterior. Now, consider DoIt with three points taken as: $\nu_1 = \hat{\gamma}$, $\nu_2 = \hat{\gamma} - 1.5\sigma$, and $\nu_2 = \hat{\gamma} + 1.5\sigma$. The approximated density, which is a weighted average of lognormal densities, is shown in Figure 1. One can see that even though the density has two tails, the approximation is much better. The DoIt approximation with 10 equally spaced points taken from $\hat{\gamma} - 3\sigma$ to $\hat{\gamma} + 1.5\sigma$ is also shown in Figure 1. One can see that the approximation is almost indistinguishable from the exact density, clearly showing that the method is promising.

As is evident from the example, a nice feature of the DoIt is that the approximation can be improved by adding more points

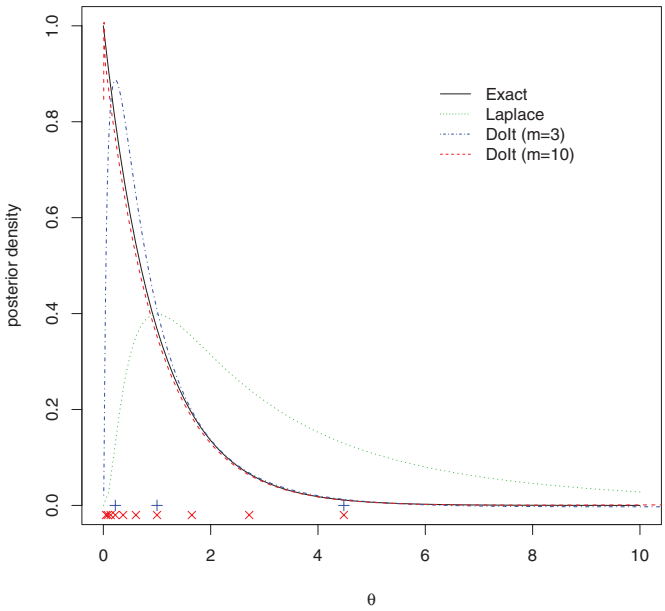

Figure 1. Comparison of the Laplace approximation and the DoIt approximation in the Poisson data example. The online version of this figure is in color.

to the design. This property is formally stated in the following theorem and is proved in the Appendix.

Theorem 1. If $h(\boldsymbol{\theta})$ is continuous, then for any $\alpha \in (0, 1)$ and any $\epsilon > 0$, there exists a finite number of points $\boldsymbol{D} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$ in $\boldsymbol{\Theta}$ such that

$$\left| \frac{\hat{h}(\boldsymbol{\theta}) / \int_{\boldsymbol{\Theta}} \hat{h}(\boldsymbol{\theta}) d\boldsymbol{\theta}}{h(\boldsymbol{\theta}) / \int_{\boldsymbol{\Theta}} h(\boldsymbol{\theta}) d\boldsymbol{\theta}} - 1 \right| < \epsilon \tag{6}$$

for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, where $\hat{h}(\boldsymbol{\theta})$ is any continuous and uniformly convergent interpolator of $h(\boldsymbol{\theta})$ on $\boldsymbol{D}$ and $\boldsymbol{\Theta}$ is the $(1 - \alpha)$ highest posterior density (HPD) credible set.

Since $\tilde{h}(\boldsymbol{\theta})$ in (3) is a continuous and uniformly convergent interpolator (Buhmann 2003) of $h(\boldsymbol{\theta})$, Theorem 1 holds true for the proposed DoIt. One can make α arbitrarily small so that the ratio in (6) is close to $\tilde{p}(\boldsymbol{\theta}|\boldsymbol{y}) / p(\boldsymbol{\theta}|\boldsymbol{y})$, provided the support of the posterior distribution is $\mathbb{R}^d$. Because ϵ can also be made arbitrarily small, one can make this ratio as close to 1 as possible.

2.2 Unknown Posterior Mode

If it is difficult to obtain the posterior mode by maximizing the unnormalized posterior $h(\boldsymbol{\theta})$, then one can proceed as follows. Assuming that a set of points $\boldsymbol{D} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$ can be sampled from a region based on the prior information about $\boldsymbol{\theta}$. Since the posterior mode is unknown, one cannot estimate $\boldsymbol{\Sigma}$ using the curvature of $\log(h(\boldsymbol{\theta}))$ at the mode. A popular approach to estimate $\boldsymbol{\Sigma}$ in the kriging and radial basis functions' literature is cross-validation. First, assume that $\boldsymbol{\Sigma}$ is a diagonal matrix with diagonal elements $\boldsymbol{\sigma}^2 = (\sigma_1^2, \ldots, \sigma_d^2)'$. Note that the off-diagonal elements are set to zero to remove the computational burden of estimating them.

The leave-one-out cross-validation error is defined as $e_i = h_i - \tilde{h}_{(i)i}$, where $\tilde{h}_{(i)i}$ is the predicted value after removing the ith point $(\boldsymbol{v}_i, h_i)$ from the dataset. The computation of the cross-validation errors can be simplified as follows. It is well known from the kriging literature that

$$e_i = \frac{(\boldsymbol{G}^{-1})_i}{(\boldsymbol{G}^{-1})_{ii}} \boldsymbol{h},$$

where $(\boldsymbol{G}^{-1})_i$ is the ith row and $(\boldsymbol{G}^{-1})_{ii}$ is the ith diagonal element of $\boldsymbol{G}^{-1}$. Let $\boldsymbol{e} = (e_1, \ldots, e_m)'$. Then, $\boldsymbol{e} = \{\text{diag}(\boldsymbol{G}^{-1})\}^{-1} \boldsymbol{G}^{-1} \boldsymbol{h}$, where $\text{diag}(\boldsymbol{G}^{-1})$ is a diagonal matrix containing the diagonal elements of $\boldsymbol{G}^{-1}$. Now, $\boldsymbol{\sigma}^2$ can be estimated by minimizing the mean squared cross-validation error $\text{MSCV} = \boldsymbol{e}'\boldsymbol{e}/m$. In my experience, I found that it is better to use a weighted version of the mean squared cross-validation error given by

$$\text{WMSCV} = \frac{1}{m} \boldsymbol{e}' \text{diag}(\boldsymbol{G}^{-1}) \boldsymbol{e}. \tag{7}$$

A justification to this modification follows from the fact that under the kriging model assumptions, $(\boldsymbol{G}^{-1})_{ii}$ is proportional to the inverse of the leave-one-out prediction variance at $\boldsymbol{v}_i$ (e.g., see Rasmussen and Williams 2006, sec. 5.4.2). Note that the minimization of WMSCV can be accomplished using a general-purpose optimization algorithm, such as the Nelder–Mead algorithm, which can become computationally challenging as d increases.

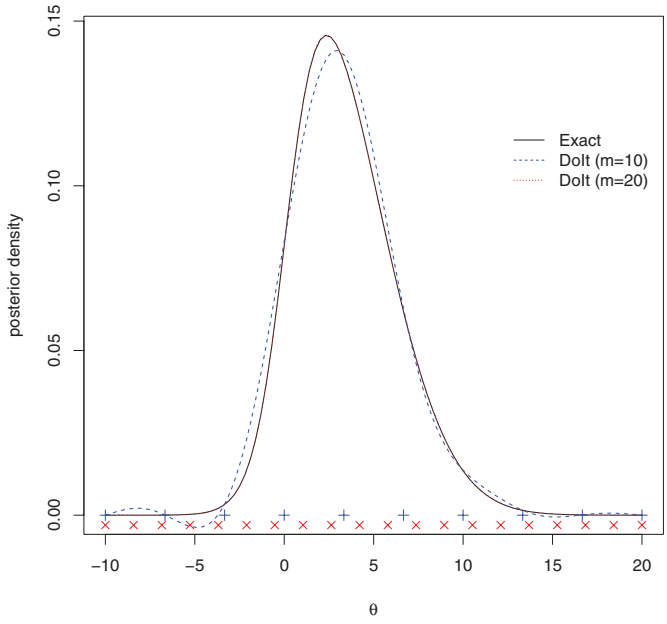

Figure 2. The DoIt approximation with $m = 10$ and $m = 20$ in the binary data example. The online version of this figure is in color.

Consider, for example, the following problem of estimating θ from a binary observation:

$$y|\theta \sim \text{Bernoulli}(\{1 + \exp(-\theta)\}^{-1}),$$
$$\theta \sim N(\mu, \tau^2).$$

Suppose $y = 1$ was observed. Choose $\mu = 1$ and $\tau = 4$. Suppose one samples 10 equally spaced points from -10 to 20. Minimizing WMSCV in (7), one obtains $\sigma^2 = 9.30$ (which can be compared with $\sigma^2 = 7.11$ obtained using the curvature information at the posterior mode). The DoIt approximation and the exact posterior density obtained using numerical integration are shown in Figure 2. One can see that DoIt gives a reasonable approximation. Better approximation can be obtained by adding more points to $\boldsymbol{D}$. The DoIt approximation with $m = 20$ points is also shown in Figure 2, which is almost identical to the exact density.

In summary, DoIt can be applied without knowledge of the posterior mode or modes and can be used even when the likelihood or prior is nondifferentiable. This overcomes some of the limitations of the Laplace method. However, it is preferable to find the mode(s), whenever possible, so that a good approximation can be obtained with fewer points.

2.3 Mixture Normal Approximation and an Improvement to DoIt

As noted before, the coefficients $\tilde{c}_i$'s can be negative and can result in regions of $\boldsymbol{\theta}$ where the approximation of the posterior distribution is negative. An example can be seen in Figure 2 for the case of $m = 10$, where there are some negative values in the lower and upper tails of the approximated posterior density. This problem will not occur if one restricts c_i's to be nonnegative. Such a solution can be obtained by minimizing

$$(\boldsymbol{h} - \boldsymbol{G}\boldsymbol{c})' \boldsymbol{G}^{-1} (\boldsymbol{h} - \boldsymbol{G}\boldsymbol{c}), \tag{8}$$

subject to the constraints $c_i \geq 0$ for all $i = 1, \ldots, m$. This is a quadratic program and can be easily solved. Note that if one removes the nonnegativity constraints, one obtains the earlier solution $\tilde{\boldsymbol{c}} = \boldsymbol{G}^{-1}\boldsymbol{h}$. Let $\hat{\boldsymbol{c}}$ denote the solution from the quadratic program. Then, the DoIt becomes exactly a mixture normal approximation given by

$$\hat{p}(\boldsymbol{\theta}|\boldsymbol{y}) \approx \frac{\hat{\boldsymbol{c}}'\boldsymbol{\phi}(\boldsymbol{\theta})}{\hat{\boldsymbol{c}}'\mathbf{1}}. \tag{9}$$

The resulting density for the binary data example with $m = 10$ points is plotted in Figure 3 as a dotted line. One can see that although the problem due to negative posterior has disappeared, the overall approximation has deteriorated. A better mixture normal approximation can be obtained using the iterated Laplace (iterLap) approximation in Bornkamp (2011a). The approximate posterior density fitted using the R package `iterLap` (Bornkamp 2011b) is also shown in Figure 3 as a dashed line. One can see that although the approximation has improved, there are still some errors. Moreover, since `iterLap` requires several optimizations of the unnormalized posterior, the method does not seem to be useful for approximating expensive posteriors, and therefore, will not be considered here.

Another approach to overcome the negative posterior density values is as follows. One can see from Figure 2 that the negative values of (5) are observed when the posterior density values are close to zero. Thus, if the DoIt approximation can be pulled toward zero at the low-probability regions, then one might be able to avoid the negative values. At the same time, it should not be pulled toward zero at the high-probability regions; otherwise, the approximation can become poor. In other words, one should multiply the DoIt approximation by a function that closely resembles the posterior distribution. One choice for this function is the mixture normal approximation in (9). Thus, $h(\boldsymbol{\theta})$ is approximated as

$$h(\boldsymbol{\theta}) \approx \sum_{i=1}^{m} \hat{c}_i g(\boldsymbol{\theta}; \boldsymbol{v}_i, \boldsymbol{\Sigma}) \left\{ a + \sum_{i=1}^{m} b_i g(\boldsymbol{\theta}; \boldsymbol{v}_i, \boldsymbol{\Lambda}) \right\},$$

where $\hat{\boldsymbol{c}}$ is the solution of the quadratic program in (8). If the mixture normal approximation is good, that is, if $h(\boldsymbol{\theta}) \approx \sum_{i=1}^{m} \hat{c}_i g(\boldsymbol{\theta}; \boldsymbol{v}_i, \boldsymbol{\Sigma})$, then a will be close to 1 and b_i will be close to 0 for all $i = 1, \ldots, m$. The optimal choices of a, $\boldsymbol{b} = (b_1, \ldots, b_m)'$, and $\boldsymbol{\Lambda}$ will be discussed later. In vector notation, $h(\boldsymbol{\theta}) \approx \hat{\boldsymbol{c}}'\boldsymbol{g}(\boldsymbol{\theta}; \boldsymbol{\Sigma})\{a + \boldsymbol{b}'\boldsymbol{g}(\boldsymbol{\theta}; \boldsymbol{\Lambda})\}$, where the notation from the previous section has been slightly changed to emphasize the use of two different variance-covariance matrices. In the same way, $\boldsymbol{G}(\boldsymbol{\Sigma})$ and $\boldsymbol{G}(\boldsymbol{\Lambda})$ are used to denote the two $\boldsymbol{G}$ matrices. Let $\boldsymbol{z} = \boldsymbol{h}/(\boldsymbol{G}(\boldsymbol{\Sigma})\hat{\boldsymbol{c}})$, where the division of the two vectors indicate an element-wise division, that is, $z_i = h(\boldsymbol{v}_i)/\hat{\boldsymbol{c}}'\boldsymbol{g}(\boldsymbol{v}_i; \boldsymbol{\Sigma})$ for $i = 1, \ldots, m$. For the moment, assume that a is given. Then, to have interpolation, one must choose $\boldsymbol{b}$ to be

$$\hat{\boldsymbol{b}} = \boldsymbol{G}(\boldsymbol{\Lambda})^{-1}(\boldsymbol{z} - a\mathbf{1}). \tag{10}$$

This approach is equivalent to using a simple kriging with a known mean equal to a. Thus, one obtains the new approximation: $\hat{h}(\boldsymbol{\theta}) = \hat{\boldsymbol{c}}'\boldsymbol{g}(\boldsymbol{\theta}; \boldsymbol{\Sigma})\{a + \hat{\boldsymbol{b}}'\boldsymbol{g}(\boldsymbol{\theta}; \boldsymbol{\Lambda})\}$. As in the development of (4), one obtains

$$\begin{aligned} \int \hat{h}(\boldsymbol{\theta})d\boldsymbol{\theta} &= a\hat{\boldsymbol{c}}' \int \boldsymbol{g}(\boldsymbol{\theta}; \boldsymbol{\Sigma})d\boldsymbol{\theta} + \hat{\boldsymbol{c}}' \int \boldsymbol{g}(\boldsymbol{\theta}; \boldsymbol{\Sigma})\boldsymbol{g}(\boldsymbol{\theta}; \boldsymbol{\Lambda})'d\boldsymbol{\theta}\ \hat{\boldsymbol{b}} \\ &= a\hat{\boldsymbol{c}}'(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}\mathbf{1} + \hat{\boldsymbol{c}}'(2\pi)^{d/2}\frac{|\boldsymbol{\Sigma}\boldsymbol{\Lambda}|^{1/2}}{|\boldsymbol{\Sigma} + \boldsymbol{\Lambda}|^{1/2}} \\ &\quad \times \boldsymbol{G}(\boldsymbol{\Sigma} + \boldsymbol{\Lambda})\hat{\boldsymbol{b}}. \end{aligned} \tag{11}$$

Now, consider the choice of a. In the kriging literature (e.g., Joseph 2006), a is taken as the generalized mean $\mathbf{1}'\boldsymbol{G}(\boldsymbol{\Lambda})^{-1}\boldsymbol{z}/\mathbf{1}'\boldsymbol{G}(\boldsymbol{\Lambda})^{-1}\mathbf{1}$. However, to get a better approximation in the high-probability regions, a different choice is used. Here, a is taken as the mean of $\hat{z}(\boldsymbol{\theta}) = \hat{h}(\boldsymbol{\theta})/\hat{\boldsymbol{c}}'\boldsymbol{g}(\boldsymbol{\theta}; \boldsymbol{\Sigma})$ with respect to the mixture normal approximation. Thus,

$$a = \int \hat{z}(\boldsymbol{\theta})\frac{\hat{\boldsymbol{c}}'\boldsymbol{\phi}(\boldsymbol{\theta}; \boldsymbol{\Sigma})}{\hat{\boldsymbol{c}}'\mathbf{1}}\,d\boldsymbol{\theta} = \frac{\int \hat{h}(\boldsymbol{\theta})d\boldsymbol{\theta}}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}\hat{\boldsymbol{c}}'\mathbf{1}}.$$

Substituting in (11) and solving for a, one obtains

$$a = \frac{\hat{\boldsymbol{c}}'\boldsymbol{G}(\boldsymbol{\Sigma} + \boldsymbol{\Lambda})\boldsymbol{G}(\boldsymbol{\Lambda})^{-1}\boldsymbol{z}}{\hat{\boldsymbol{c}}'\boldsymbol{G}(\boldsymbol{\Sigma} + \boldsymbol{\Lambda})\boldsymbol{G}(\boldsymbol{\Lambda})^{-1}\mathbf{1}}. \tag{12}$$

For this choice, the marginal likelihood in (11) takes the simple form:

$$\int \hat{h}(\boldsymbol{\theta})d\boldsymbol{\theta} = a(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}\hat{\boldsymbol{c}}'\mathbf{1}.$$

Thus, the new DoIt approximation of the posterior distribution is given by

$$\hat{p}(\boldsymbol{\theta}|\boldsymbol{y}) \approx \frac{\hat{\boldsymbol{c}}'\boldsymbol{\phi}(\boldsymbol{\theta}; \boldsymbol{\Sigma})}{\hat{\boldsymbol{c}}'\mathbf{1}}\{1 + \hat{\boldsymbol{b}}'\boldsymbol{g}(\boldsymbol{\theta}; \boldsymbol{\Lambda})/a\}. \tag{13}$$

Let $\boldsymbol{V} = \boldsymbol{\Sigma}(\boldsymbol{\Sigma} + \boldsymbol{\Lambda})^{-1}\boldsymbol{\Lambda}$ and $\boldsymbol{\mu}_{ij} = \boldsymbol{V}(\boldsymbol{\Sigma}^{-1}\boldsymbol{v}_i + \boldsymbol{\Lambda}^{-1}\boldsymbol{v}_j)$. Then, using the identity

$$g(\boldsymbol{\theta}; \boldsymbol{v}_i, \boldsymbol{\Sigma})g(\boldsymbol{\theta}; \boldsymbol{v}_j, \boldsymbol{\Lambda}) = g(\boldsymbol{v}_i; \boldsymbol{v}_j, \boldsymbol{\Sigma} + \boldsymbol{\Lambda})g(\boldsymbol{\theta}, \boldsymbol{\mu}_{ij}, \boldsymbol{V}),$$

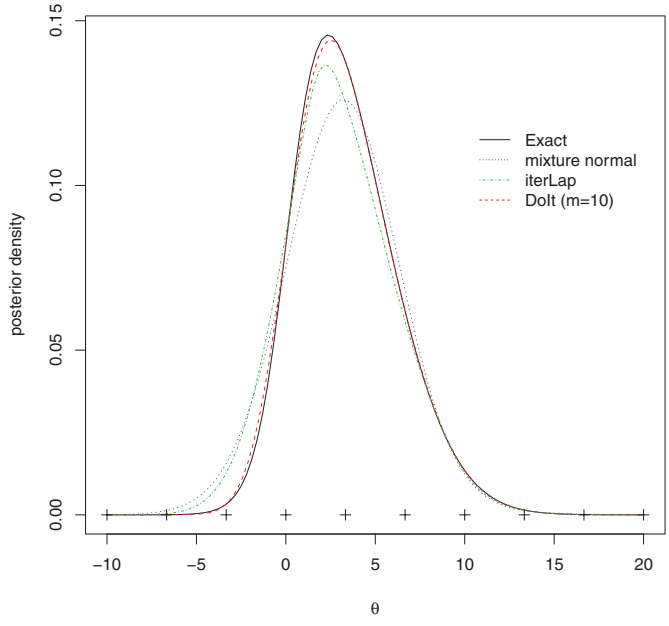

Figure 3. Comparison of mixture normal approximation with $m = 10$, iterated Laplace approximation, and improved DoIt approximation with $m = 10$ in the binary data example. The online version of this figure is in color.

Equation (13) can also be written as

$$\hat{p}(\boldsymbol{\theta}|\boldsymbol{y}) \approx \frac{\sum_{i=1}^{m} \hat{c}_i \phi(\boldsymbol{\theta}; \boldsymbol{v}_i, \boldsymbol{\Sigma}) + \sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij}\phi(\boldsymbol{\theta}; \boldsymbol{\mu}_{ij}, \boldsymbol{V})}{\sum_{i=1}^{m} \hat{c}_i}, \tag{14}$$

where

$$d_{ij} = \frac{\hat{c}_i \hat{b}_j |\boldsymbol{\Lambda}|^{1/2}}{a|\boldsymbol{\Sigma} + \boldsymbol{\Lambda}|^{1/2}} g(\boldsymbol{v}_i; \boldsymbol{v}_j, \boldsymbol{\Sigma} + \boldsymbol{\Lambda}).$$

Note that $\sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij} = 0$. Thus, the new DoIt approximation is also a weighted average of normal distributions. Hereafter, the new DoIt in (13) or (14) is referred simply as DoIt.

The matrix $\boldsymbol{\Lambda}$ can be estimated using cross-validation, as was done for $\boldsymbol{\Sigma}$ in the previous section. To be more specific, let $\boldsymbol{\Lambda} = \text{diag}(\boldsymbol{\lambda})\boldsymbol{\Sigma}\text{diag}(\boldsymbol{\lambda})$, where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_d)'$. Now, estimate $\boldsymbol{\lambda}$ by minimizing $\hat{\boldsymbol{b}}'\{\text{diag}(\boldsymbol{G}(\boldsymbol{\Lambda})^{-1})\}^{-1}\hat{\boldsymbol{b}}/m$ as in (7), where $\hat{\boldsymbol{b}}$ and a are computed using (10) and (12), respectively. Another approach to estimate $\boldsymbol{\lambda}$ is to make a Gaussian process (GP) assumption on the simple kriging model and use likelihood-based methods (e.g., see Santner, Williams, and Notz 2003, p. 66). In this work, the cross-validation-based methods have been used for estimation.

The DoIt approximation for the posterior distribution in the binary data example with $m = 10$ points is shown in Figure 3 as a dashed line. One can see that the approximation is almost indistinguishable from the true density and there are no visible negative posterior values. Clearly, the improvement obtained over the mixture normal and the earlier DoIt approximation is quite substantial. DoIt again does not guarantee the approximated posterior density to be positive, but it seems to mitigate the negative-value problems to an extent that one need not worry about it anymore. However, in applications where the interest is in calculating tail probabilities, DoIt should be used with caution.

Consider a more challenging example from Marin and Robert (2007, example 2.1, p. 26). Suppose two observations $y_1 = -4.3$ and $y_2 = 3.2$ are generated from a Cauchy distribution Cauchy$(\theta, 1)$. The objective is to estimate θ using the prior distribution $\theta \sim N(0, (\sqrt{10})^2)$. The unnormalized posterior is given by

$$h(\theta) = \frac{\exp(-\theta^2/20)}{\prod_{i=1}^{2}(1 + (y_i - \theta)^2)}.$$

Suppose one samples 10 equally spaced points from -10 to 10. The DoIt approximation in (13) and the exact posterior density obtained using numerical integration are shown in Figure 4. The DoIt approximation with 20 equally spaced points from -10 to 10 is also shown in Figure 4, which gives a better fit to the exact posterior. One can see that DoIt has no problem in capturing the bimodal nature of the posterior distribution. However, improved approximations can be obtained by using the curvature information at each mode, such as by fitting a better mixture normal approximation. Because it can complicate the formulas and their implementation, this extension will be considered in a future work. To continue with the framework introduced here, if multiple modes are encountered (or if the Laplace approximation is extremely poor), $\boldsymbol{\Sigma}$ is replaced with $\text{diag}(\boldsymbol{w})\boldsymbol{\Sigma}\text{diag}(\boldsymbol{w})$, where $\boldsymbol{\Sigma}$ is based on the curvature information at the mode

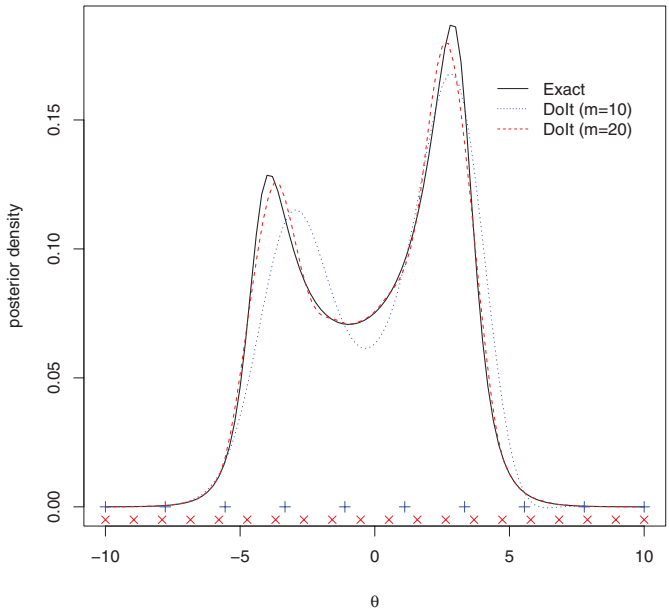

Figure 4. The DoIt approximation in the Cauchy data example. The online version of this figure is in color.

having highest posterior density and $\boldsymbol{w} = (w_1, \ldots, w_m)'$ is estimated using cross-validation methods.

2.4 Marginal Distributions and Posterior Quantities

Marginal posterior distributions can be computed from (14) using properties of the multivariate normal distribution. For instance, the marginal posterior distribution of θ_k is given by

$$\hat{p}(\theta_k|\boldsymbol{y}) \approx \frac{\sum_{i=1}^{m} \hat{c}_i \phi(\theta_k; \nu_{ik}, \boldsymbol{\Sigma}_{kk}) + \sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij}\phi(\theta_k; \mu_{ijk}, \boldsymbol{V}_{kk})}{\sum_{i=1}^{m} \hat{c}_i}, \tag{15}$$

where ν_{ik}, ν_{jk}, and μ_{ijk} are the kth components of $\boldsymbol{v}_i$, $\boldsymbol{v}_j$, and $\boldsymbol{\mu}_{ij}$, respectively.

Many of the required posterior quantities, such as mean and variance, can also be easily calculated. For example,

$$E(\boldsymbol{\theta}|\boldsymbol{y}) = \bar{\boldsymbol{\theta}} \approx \frac{\sum_{i=1}^{m} \hat{c}_i \boldsymbol{v}_i + \sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij}\boldsymbol{\mu}_{ij}}{\sum_{i=1}^{m} \hat{c}_i}$$

and

$$\text{var}(\boldsymbol{\theta}|\boldsymbol{y}) \approx \frac{\sum_{i=1}^{m} \hat{c}_i(\boldsymbol{v}_i\boldsymbol{v}_i' + \boldsymbol{\Sigma}) + \sum_{i=1}^{m}\sum_{j=1}^{m} d_{ij}(\boldsymbol{\mu}_{ij}\boldsymbol{\mu}_{ij}' + \boldsymbol{V})}{\sum_{i=1}^{m} \hat{c}_i} - \bar{\boldsymbol{\theta}}\bar{\boldsymbol{\theta}}'.$$

More generally, one may be interested in the computation of

$$\xi = E\{f(\boldsymbol{\theta})|\boldsymbol{y}\} \approx \int f(\boldsymbol{\theta}) \frac{\hat{\boldsymbol{c}}'\boldsymbol{\phi}(\boldsymbol{\theta}; \boldsymbol{\Sigma})}{a\hat{\boldsymbol{c}}'\boldsymbol{1}}\{a + \hat{\boldsymbol{b}}'\boldsymbol{g}(\boldsymbol{\theta}; \boldsymbol{\Lambda})\}\, d\boldsymbol{\theta}$$

for some continuous function $f(\boldsymbol{\theta})$. An explicit calculation of this integral can be difficult, except for a few simple functions, and therefore, approximation is resorted to. First, let $f^*(\boldsymbol{\theta}) =$

$f(\boldsymbol{\theta})z(\boldsymbol{\theta})$, where $z(\boldsymbol{\theta}) = a + \hat{\boldsymbol{b}}'\boldsymbol{g}(\boldsymbol{\theta}; \boldsymbol{\Lambda})$. Then,

$$\xi \approx \frac{1}{a\hat{\boldsymbol{c}}'\mathbf{1}} \sum_{i=1}^{m} \hat{c}_i \int f^*(\boldsymbol{\theta})\phi(\boldsymbol{\theta}; \boldsymbol{v}_i, \boldsymbol{\Sigma})\, d\boldsymbol{\theta}. \tag{16}$$

Now, one can approximate $f^*(\boldsymbol{\theta})$ using kriging. Let $\boldsymbol{f} = (f(\boldsymbol{v}_1), \ldots, f(\boldsymbol{v}_m))'$ and $\boldsymbol{z} = a\mathbf{1} + \boldsymbol{G}(\boldsymbol{\Lambda})\hat{\boldsymbol{b}}$. Then, $\boldsymbol{f}^* = (f^*(\boldsymbol{v}_1), \ldots, f^*(\boldsymbol{v}_m))' = \boldsymbol{f} \odot \boldsymbol{z}$, where $\odot$ denotes element-wise multiplication.

A key idea in this approximation is to use the following kriging predictor:

$$f^*(\boldsymbol{\theta}) = \alpha z(\boldsymbol{\theta}) + \boldsymbol{g}(\boldsymbol{\theta}; \boldsymbol{\Omega})'\boldsymbol{G}(\boldsymbol{\Omega})^{-1}(\boldsymbol{f}^* - \alpha\boldsymbol{z}), \tag{17}$$

where α is a constant that needs to be specified. Then, the integral in (16) becomes

$$\begin{aligned}\int f^*(\boldsymbol{\theta})\phi(\boldsymbol{\theta}; \boldsymbol{v}_i, \boldsymbol{\Sigma})\, d\boldsymbol{\theta} &= \alpha \int z(\boldsymbol{\theta})\phi(\boldsymbol{\theta}; \boldsymbol{v}_i, \boldsymbol{\Sigma})\, d\boldsymbol{\theta} \\ &\quad + \int \boldsymbol{g}(\boldsymbol{\theta}; \boldsymbol{\Omega})'\phi(\boldsymbol{\theta}; \boldsymbol{v}_i, \boldsymbol{\Sigma})\, d\boldsymbol{\theta} \\ &\quad \times \boldsymbol{G}(\boldsymbol{\Omega})^{-1}(\boldsymbol{f}^* - \alpha\boldsymbol{z}).\end{aligned}$$

It is easy to show that

$$\int g(\boldsymbol{\theta}; \boldsymbol{v}_j, \boldsymbol{\Omega})\phi(\boldsymbol{\theta}; \boldsymbol{v}_i, \boldsymbol{\Sigma})\, d\boldsymbol{\theta} = \frac{|\boldsymbol{\Omega}|^{1/2}}{|\boldsymbol{\Omega} + \boldsymbol{\Sigma}|^{1/2}} g(\boldsymbol{v}_i; \boldsymbol{v}_j, \boldsymbol{\Omega} + \boldsymbol{\Sigma}).$$

Thus,

$$\begin{aligned}\int f^*(\boldsymbol{\theta})\phi(\boldsymbol{\theta}; \boldsymbol{v}_i, \boldsymbol{\Sigma})\, d\boldsymbol{\theta} &= \alpha\left(a + \frac{|\boldsymbol{\Lambda}|^{1/2}}{|\boldsymbol{\Sigma} + \boldsymbol{\Lambda}|^{1/2}}\boldsymbol{G}_i(\boldsymbol{\Sigma} + \boldsymbol{\Lambda})\hat{\boldsymbol{b}}\right) \\ &\quad + \frac{|\boldsymbol{\Omega}|^{1/2}}{|\boldsymbol{\Omega} + \boldsymbol{\Sigma}|^{1/2}}\boldsymbol{G}_i(\boldsymbol{\Omega} + \boldsymbol{\Sigma}) \\ &\quad \times \boldsymbol{G}(\boldsymbol{\Omega})^{-1}(\boldsymbol{f}^* - \alpha\boldsymbol{z}),\end{aligned}$$

where $\boldsymbol{G}_i(\boldsymbol{\Omega} + \boldsymbol{\Sigma})$ and $\boldsymbol{G}_i(\boldsymbol{\Sigma} + \boldsymbol{\Lambda})$ denote the ith rows of $\boldsymbol{G}(\boldsymbol{\Omega} + \boldsymbol{\Sigma})$ and $\boldsymbol{G}(\boldsymbol{\Sigma} + \boldsymbol{\Lambda})$, respectively. Substituting in (16), one obtains

$$\xi \approx \alpha + \frac{|\boldsymbol{\Omega}|^{1/2}}{a\hat{\boldsymbol{c}}'\mathbf{1}|\boldsymbol{\Omega} + \boldsymbol{\Sigma}|^{1/2}}\hat{\boldsymbol{c}}'\boldsymbol{G}(\boldsymbol{\Omega} + \boldsymbol{\Sigma})\boldsymbol{G}(\boldsymbol{\Omega})^{-1}(\boldsymbol{f}^* - \alpha\boldsymbol{z}).$$

Similar to the arguments made in Joseph (2006), choosing $\alpha = \xi$ makes the approximation less sensitive to the choice of the covariance matrix $\boldsymbol{\Omega}$. Then, one obtains

$$\xi \approx \frac{\hat{\boldsymbol{c}}'\boldsymbol{G}(\boldsymbol{\Omega} + \boldsymbol{\Sigma})\boldsymbol{G}(\boldsymbol{\Omega})^{-1}\boldsymbol{f}^*}{\hat{\boldsymbol{c}}'\boldsymbol{G}(\boldsymbol{\Omega} + \boldsymbol{\Sigma})\boldsymbol{G}(\boldsymbol{\Omega})^{-1}\boldsymbol{z}}.$$

As before, $\boldsymbol{\Omega}$ can be estimated using cross-validation. But since the predictions are less sensitive to the choice of $\boldsymbol{\Omega}$, a reasonable approximation can be obtained by taking $\boldsymbol{\Omega} = \boldsymbol{\Lambda}$, which significantly reduces the computations. Thus,

$$\xi \approx \frac{\hat{\boldsymbol{c}}'\boldsymbol{G}(\boldsymbol{\Sigma} + \boldsymbol{\Lambda})\boldsymbol{G}(\boldsymbol{\Lambda})^{-1}\boldsymbol{f}^*}{\hat{\boldsymbol{c}}'\boldsymbol{G}(\boldsymbol{\Sigma} + \boldsymbol{\Lambda})\boldsymbol{G}(\boldsymbol{\Lambda})^{-1}\boldsymbol{z}}. \tag{18}$$

Consider, for example, the computation of the posterior predictive density in the binary data example from Section 2.2. Here, the posterior predictive distribution is Bernoulli, with probability $\xi = E(\{1 + \exp(-\theta)\}^{-1}|\boldsymbol{y})$. Numerical integration gives $\xi = 0.8496$. Now, using (18), one obtains $\xi \approx 0.8478$, which is very close to the true value and much better than the first-order approximation $\xi \approx \{1 + \exp(-\hat{\theta})\}^{-1} = 0.914$.

2.5 Comparison With Other Approximation Methods

Recently, a wealth of deterministic methods have been proposed in the machine learning literature for approximate Bayesian inference, such as VB methods; see the reviews in Bishop (2006) and Ormerod and Wand (2010), and the references therein. Another deterministic approximation method that is popular in machine learning is the EP algorithm of Minka (2001). A more recent development on deterministic methods is the INLA proposed by Rue, Martino, and Chopin (2009), which can be applied to a class of regression problems, known as latent Gaussian models. In general, these methods are much faster than the MCMC algorithms and more accurate than the original Laplace approximation.

For comparison with the DoIt, consider again the binary data problem introduced in Section 2.2. Using the tangent transform variational approximation in Jaakkola and Jordan (2000), one obtains $\theta|\boldsymbol{y} \sim^a N(\mu_{\text{VB}}, \tau^2_{\text{VB}})$, where $\tau^2_{\text{VB}} = (1/\tau^2 + 0.5/\xi \tanh(\xi/2))^{-1}$, $\mu_{\text{VB}} = \tau^2_{\text{VB}}(\mu/\tau^2 + 1/2)$, and ξ is solved from $\xi^2 = \tau^2_{\text{VB}} + \mu^2_{\text{VB}}$. Figure 5(a) shows the VB approximation to the posterior density, which can be compared with the DoIt approximation in Figure 3. One can see that the variational method underestimates the posterior variance, leading to a poor approximation of the density. This underestimation of variance has been observed by Jaakkola and Jordan (2000) as well as by other researchers (Rue, Martino, and Chopin 2009). On the other hand, the EP algorithm works through moment matching of approximate marginal posterior distributions. Figure 5(a) also shows a normal distribution approximation obtained by matching the posterior mean and variance, which can be considered as the solution of the EP algorithm using Gaussian distributions. One can see that although this approximation is better than that obtained by the tangent transform variational method, it is still not satisfactory due to the skewness of the true posterior density. Of course, this example is a bit unfair to the EP algorithm because only one binary observation has been used. With more data, the marginal posterior distributions get closer to normal distributions and the EP algorithm will become more accurate (see Kuss and Rasmussen 2005). A main advantage of DoIt over these methods is that its accuracy can be improved by adding more evaluation points. Another advantage of DoIt is its ease of implementation. As can be seen in the foregoing binary data example, VB methods and the EP algorithm require problem-specific developments, whereas DoIt can be implemented almost as a black box method where the user needs to update only the likelihood and prior information.

The idea of using interpolation techniques for Bayesian computation is not new and can be traced back to at least O'Hagan (1991), where he used GP models for the integrand in a Bayesian integration problem. He derived Bayes–Hermite quadrature rules for integration similar in spirit to the widely used Gauss–Hermite quadrature rules. Extensions of these quadrature methods to nonnormal distributions were considered by Kennedy (1998). On the other hand, Rasmussen and Ghahramani (2003) used importance sampling techniques to address the nonnormal distributions. The DoIt proposed here is much more

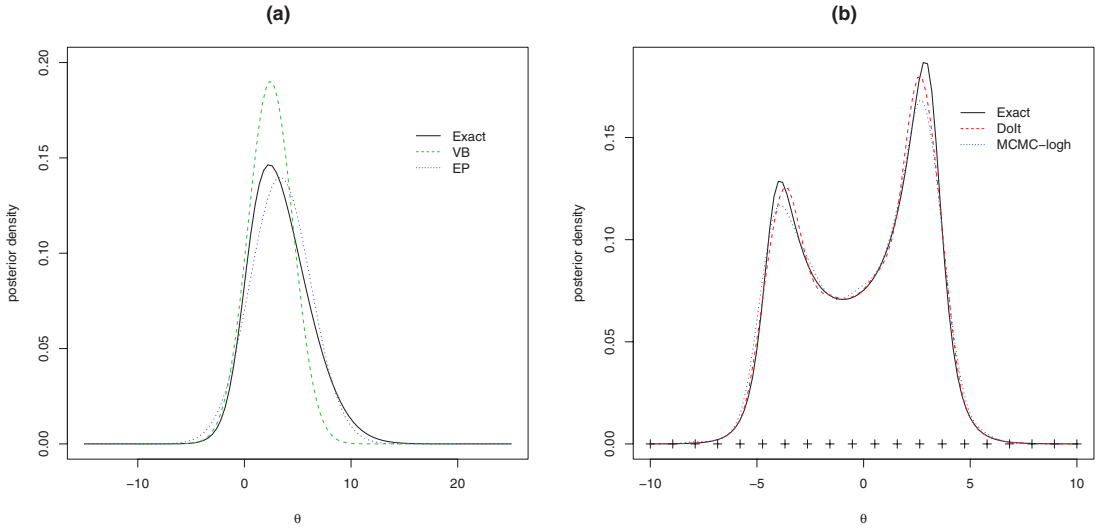

Figure 5. Posterior densities obtained using (a) VB and EP in the binary data example, and (b) DoIt and MCMC using log-posterior approximation in the Cauchy data example. The online version of this figure is in color.

general than these earlier works. There is no need to guess the shape of the posterior distribution a priori as in Kennedy (1998), and thus, DoIt can be applied to a wider class of Bayesian problems. Moreover, closed-form approximations of posterior distributions, marginal likelihoods, and marginal posterior distributions are derived, which makes the implementation of DoIt extremely easy. Furthermore, reasonable approximations to any continuous functionals of the parameters can also be obtained without the need of fitting new GP models. As will be shown in the next section, efficient designs for the evaluation points can be generated using some of the ideas in computer experiments' literature, and therefore, tedious derivations of quadrature points are also not necessary for implementation.

Another line of research is in using GP to approximate computationally expensive posterior densities to speed up MC/MCMC sampling (see the hybrid MC implementation in Rasmussen 2003). An extension of this approach was recently proposed by Fielding, Nott, and Liong (2011). A distinguishing feature in their approach is that the logarithm of the unnormalized posterior is approximated using the GP model. This avoids the negative-value problems encountered in directly approximating the unnormalized posterior, which is a great advantage. However, the drawback is that the integrals become analytically untractable, and therefore, one has to resort to MC/MCMC techniques. Similar approaches to the Bayesian calibration of computationally expensive models can be found in Bliznyuk et al. (2008) and Henderson et al. (2009). Bliznyuk et al. (2008) used radial basis functions to approximate the log-posterior, whereas Henderson et al. (2009) used a GP to approximate the expensive simulation model instead of using the posterior. The comparison of DoIt with some of these methods is postponed until Section 3.2.

However, it is of immediate interest to see how the log-posterior approximation used in Bliznyuk et al. (2008) and Fielding, Nott, and Liong (2011) compares with that of the direct posterior approximation employed in DoIt. For this purpose, an ordinary kriging model is fitted with a Gaussian correlation function using $m = 20$ points to the log-unnormalized posterior $\log(h(\theta))$ in the Cauchy example. The density of 20,000 draws obtained using a Metropolis–Hastings algorithm is shown in Figure 5(b), along with the DoIt approximation constructed using the same set of 20 points. One can see that DoIt performs slightly better in this example. In my experience, I found that the log-posterior approximation works better than the DoIt approximation for unimodal distributions but not for multimodal distributions. A simple explanation to this observation is that most multimodal distributions are finite mixtures and thus are additive in density scale and not in log-density scale. Another advantage of DoIt is that it is a pure deterministic approach and does not require generating random samples, which saves additional computations. This advantage over the log-posterior approximation diminishes as the posterior density becomes more and more expensive to calculate.

3. EXPERIMENTAL DESIGN

Clearly, the choice of the evaluation points is critical for the success of DoIt. A general strategy that will be adopted here is to first choose a space-filling design and then to add points sequentially to improve the accuracy of approximation. Below, the space-filling design using a 12-parameter Poisson nonlinear mixed model and the sequential design using a difficult-to-approximate banana-shaped posterior density have been illustrated.

3.1 Initial Space-Filling Design

First, consider the case of a single known posterior mode. Knowing the posterior mode ($\hat{\boldsymbol{\theta}}$) and the variability around it ($\boldsymbol{\Sigma}$) helps to define a region inside the parameter space from which one can select the evaluation points $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$. Arrange the points so that $\boldsymbol{D} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m)'$ is an $m \times d$ design matrix. Because $\theta_i|\boldsymbol{y}$'s may be dependent, transform the parameters to $\boldsymbol{\alpha} = \boldsymbol{\Sigma}^{-1/2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})$. Now, by Laplace's approximation, $\boldsymbol{\alpha}|\boldsymbol{y} \sim^a N(\boldsymbol{0}, \boldsymbol{I})$, where $\boldsymbol{0}$ is a vector of 0's having length d. Thus, α_i's are approximately uncorrelated. Therefore, first, one can choose a

design $\boldsymbol{D}^* = (\boldsymbol{v}_1^*, \ldots, \boldsymbol{v}_m^*)'$ uniformly distributed in $(0, 1)^d$ and then obtain $\boldsymbol{D} = (\hat{\boldsymbol{\theta}} + \boldsymbol{\Sigma}^{1/2}\Phi^{-1}(\boldsymbol{v}_1^*), \ldots, \hat{\boldsymbol{\theta}} + \boldsymbol{\Sigma}^{1/2}\Phi^{-1}(\boldsymbol{v}_m^*))'$, where Φ is the standard normal distribution function.

Because the likelihood evaluations are deterministic, experimental designs for computer experiments, such as latin hypercube design (LHD), are more suitable here (e.g., see Santner, Williams, and Notz 2003). A maximin LHD (MmLHD) can be obtained by maximizing the minimum distance among the points (Morris and Mitchell 1995). However, in the problem discussed in this article, one needs to fix one of the design points at the posterior mode. This can be achieved as follows. Let $\boldsymbol{v}_1^*$ be the center point $\mathbf{0.5} = (0.5, \ldots, 0.5)'$. Then, the remaining $m - 1$ points can be obtained by minimizing

$$\left\{\sum_{i=2}^{m}\sum_{j=2}^{m} 1/d^k(\boldsymbol{v}_i^*, \boldsymbol{v}_j^*)\right\}^{1/k}$$

for some large value of k (such as $k = 15$), where $d(\boldsymbol{v}_i^*, \boldsymbol{v}_j^*)$ denotes the Euclidian distance between $\boldsymbol{v}_i^*$ and $\boldsymbol{v}_j^*$.

Regarding the choice of sample size for the initial space-filling design, a common rule of thumb in the computer experiments' literature is to use $m = 10d$ (see Loeppky, Sacks, and Welch 2009). However, approximation of posterior densities is different from that of computer models in the sense that the domain of approximation is not well defined. Therefore, using a larger sample size, say, $m = 50d$, is recommended.

For posterior distributions with multiple modes, one can take a union of the designs constructed for each mode and then remove some points from the intersecting regions of the designs. If the modes are unknown, then the points should be taken from a region based on the prior information. Many points are likely to be from the low-probability regions, and therefore, a much larger sample size should be used. Moreover, the accuracy of the approximation needs to be assessed using cross-validation and more points should be added, as described later.

For illustrative purposes, consider the problem of predicting the density of nanowires (y) with respect to the thickness of polymer films (x) in a solution-based growth process. Eight experiments were conducted with two replicates (except for one run). The details and the data are given in Dasgupta, Weintraub, and Joseph (2011). The density of nanowires is assumed to follow a Poisson distribution with mean $\mu(x)$, where

$$\mu(x) = \theta_1 \exp(-\theta_2 x^2) + \theta_3\{1 - \exp(-\theta_2 x^2)\}\Phi(-x/\theta_4).$$

Here, their model has been extended by explicitly including the possibility of experimental errors, such as the differences in the preparation of substrates, changes in the machine settings, etc. Thus, for the ith run, let $\mu(x_i) = \left[\theta_1 \exp(-\theta_2 x_i^2) + \theta_3\{1 - \exp(-\theta_2 x_i^2)\}\Phi(-x_i/\theta_4)\right] u_i$ for $i = 1, \ldots, 8$. Note that because both the replicates are obtained from the same experimental setup, only one parameter is introduced for each run. All of the parameters must be positive, and therefore, it makes sense to transform them to log-scale. Let $\gamma_i = \log(\theta_i)$ for $i = 1, \ldots, 4$ and $\alpha_i = \log(u_i)$ for $i = 1, \ldots, 8$. Here, an independent and noninformative prior for $\boldsymbol{\gamma}$: $p(\boldsymbol{\gamma}) \propto 1$ is assumed; however, for identification purposes, an informative prior for $\boldsymbol{\alpha}$ is used. Assuming the experimental errors can be as large as 20%, $\alpha_i \overset{\text{iid}}{\sim} N(0, 0.1^2)$ is chosen.

By maximizing the log-likelihood, $\hat{\boldsymbol{\gamma}} = (4.82, -1.69, 3.32, 2.37)'$ and $\hat{\boldsymbol{\alpha}} = (-0.003, 0.005, -0.008, 0.014, -0.007, -0.007, 0.011, -0.005)'$ are obtained. The $\boldsymbol{\Sigma}$ can now be obtained through numerical differentiation. Suppose $m = 50 \times 12 = 600$ is chosen. To avoid tedious programming, the `lhs` package in R version 2.9.2 (Carnell 2009) is used to obtain an MmLHD in $[0.001, 0.999]^{12}$ and the closest point to $\mathbf{0.5}$ in the design is replaced with $\mathbf{0.5}$. Now, one can transform, rotate, and shift the points to the desired region based on the Laplace approximation. Two two-dimensional projections of the points are shown in Figure 6. One can see that γ_3 and γ_4 are highly correlated, and therefore, the rotation of the points made using the $\boldsymbol{\Sigma}$ matrix was quite effective in obtaining a good design.

The marginal densities of θ_k's computed using (15) are plotted in Figure 7 (dashed lines). The marginal densities of u_i's can be obtained similarly but are not shown here. For comparison purposes, three million samples from the posterior using Metropolis algorithm have been drawn. Note that such a large

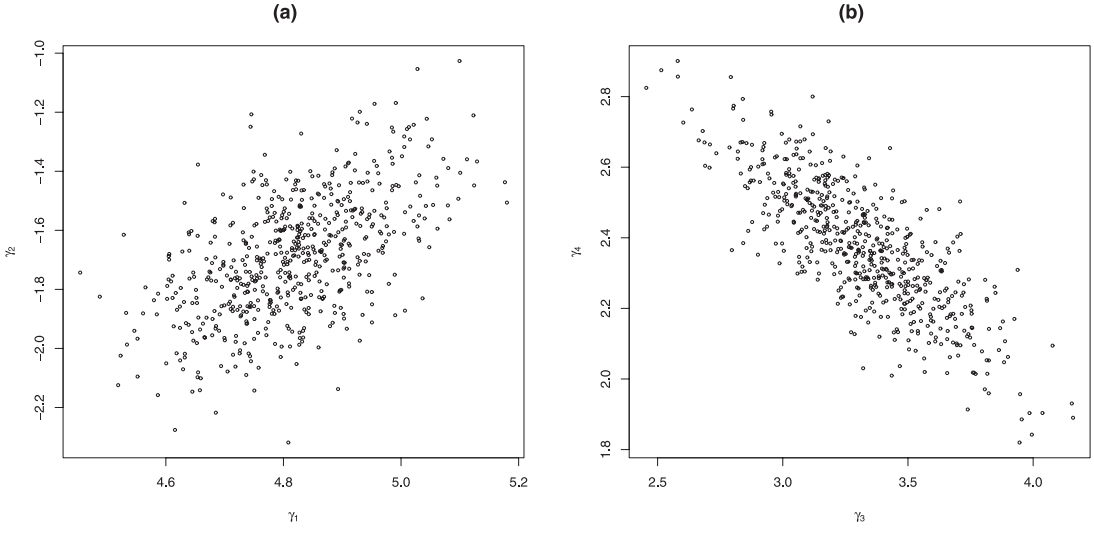

Figure 6. Two-dimensional projection of the design points in the nanowire example: (a) γ_1 versus γ_2 and (b) γ_3 versus γ_4.

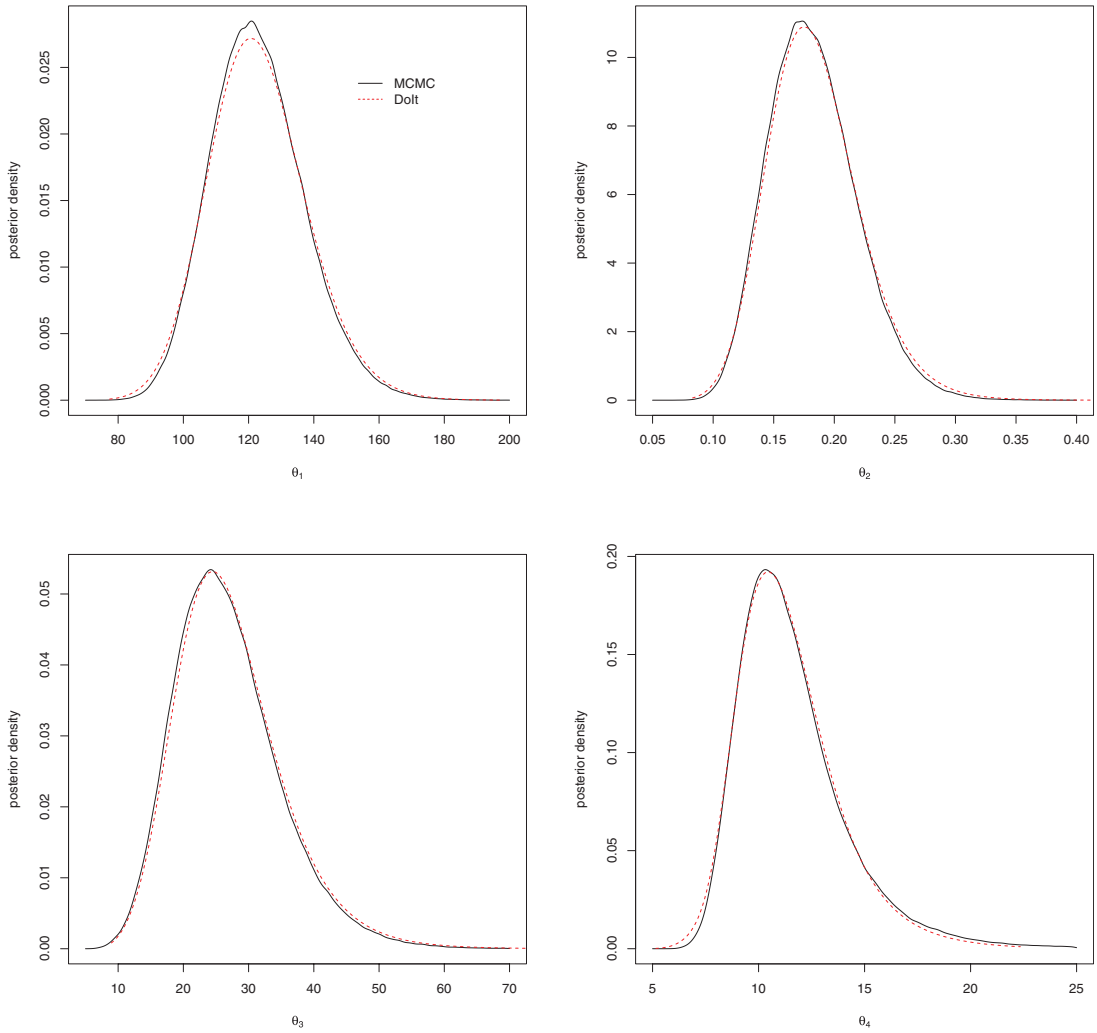

Figure 7. Marginal posterior densities of θ_i's in the nanowire example. The online version of this figure is in color.

sample for the Metropolis algorithm has been chosen only for the purpose of getting a gold standard, which may not be needed in its day-to-day use. The density plots of the samples after a burn-in of 10,000 are also shown in Figure 7 as solid lines. One can see that the DoIt gives a reasonably good approximation to the posterior density. For comparison with the other deterministic approximation methods, the quadrature method was implemented using the `cubature` package in R (Johnson and Narasimhan 2009). However, the method failed to produce even the normalizing constant after waiting for one whole day, whereas the DoIt took only about 5 min for the entire computation on a 3.20-GHz computer. Application of VB, EP, and INLA methods to this problem is not straightforward, owing to the nonlinear model structure. Ormerod and Wand (2012) recently developed Gaussian variational approximation (GVA) for generalized linear mixed models. GVA cannot be directly applied here because the model discussed here is Bayesian and nonlinear. However, assuming some methods can be devised for its implementation, the final marginal densities are going to be Gaussian. Therefore, the best-fitting normal distribution (lognormal after transformation), with the mean and variance estimated from the MCMC samples, is plotted in Figure 8 for the case of θ_4. One can see that it does not give a good fit to the true posterior because of the skewness of the distribution. The Laplace approximation, also plotted for reference, obviously gives a poor approximation.

3.2 Sequential Design

If the accuracy of the approximation based on the initial space-filling design is not adequate, then more points need to be added to improve the accuracy. Consider a sequential strategy of adding one point to the design at a time. It is well known in the literature of optimal design of experiments that the information gain can be maximized by adding the new point at the location with maximum prediction variance (see Fedorov 1972). Similar ideas have been used for active learning by MacKay (1992) and Cohn (1994); see also Gramacy and Lee (2009) for the adoption of these ideas in computer experiments. Since the DoIt approximation in (13) is not based on any stochastic model, it is not easy to obtain the prediction variance. However, it is easy to obtain a conditional prediction variance because the DoIt approximation can be viewed as a simple kriging predictor, given $\hat{\boldsymbol{c}}' \boldsymbol{g}(\boldsymbol{\theta}; \boldsymbol{\Sigma})$. This conditional variance is proportional to

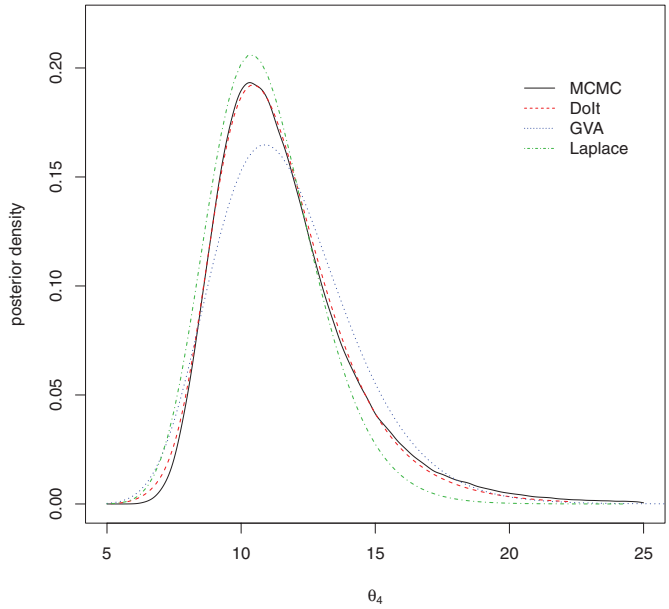

Figure 8. Comparison of DoIt with the Laplace approximation and GVA (computed using posterior mean and variance) in the nanowire example for the marginal posterior distribution of θ_4. The online version of this figure is in color.

$(\hat{\boldsymbol{c}}'\boldsymbol{g}(\boldsymbol{\theta};\boldsymbol{\Sigma}))^2\{1-\boldsymbol{g}(\boldsymbol{\theta};\boldsymbol{\Lambda})'\boldsymbol{G}^{-1}(\boldsymbol{\Lambda})\boldsymbol{g}(\boldsymbol{\theta};\boldsymbol{\Lambda})\}$. Thus, the new point is chosen as

$$\boldsymbol{v}_{m+1} = \arg\max_{\boldsymbol{\theta}} (\hat{\boldsymbol{c}}'\boldsymbol{g}(\boldsymbol{\theta};\boldsymbol{\Sigma}))^2\{1-\boldsymbol{g}(\boldsymbol{\theta};\boldsymbol{\Lambda})'\boldsymbol{G}^{-1}(\boldsymbol{\Lambda})\boldsymbol{g}(\boldsymbol{\theta};\boldsymbol{\Lambda})\}. \tag{19}$$

The foregoing criterion makes sense intuitively because the second term is 0 at the already observed locations $\{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_m\}$. Thus, by maximizing the variance, one moves away from those locations. Moreover, since the mixture normal approximation [the first bracketed term in (19)] roughly captures the shape of the posterior, one moves toward the regions with large probability mass. This is desirable. However, the objective function in (19) can be multimodal and hard to optimize. To circumvent this problem, a local optimization in a region where the variance is expected to be large can be performed. One approach to identify this region is to use a leave-one-out estimation strategy. Let $v_{(i)}$ be the estimate of prediction variance at $\boldsymbol{v}_i$ after removing point $\boldsymbol{v}_i$ from the design. Computation of $v_{(i)}$'s is complicated because there is no explicit expression for $\hat{\boldsymbol{c}}$, the solution to a quadratic program. Repeating this m times makes the computations quite demanding. To reduce the computational time, an approximation for the $v_{(i)}$'s is used. As will be shown in the Appendix,

$$v_{(i)} \approx \left(h_i + l_i - \frac{\boldsymbol{G}_i^{-1}(\boldsymbol{\Sigma})}{\boldsymbol{G}_{ii}^{-1}(\boldsymbol{\Sigma})}(\boldsymbol{h}+\boldsymbol{l})\right)^2 \frac{1}{\boldsymbol{G}_{ii}^{-1}(\boldsymbol{\Lambda})} \tag{20}$$

for $i=1,\ldots,m$, where $\boldsymbol{l} = \boldsymbol{G}(\boldsymbol{\Sigma})\hat{\boldsymbol{c}} - \boldsymbol{h}$ and $\boldsymbol{G}_i^{-1}(\boldsymbol{\Sigma})$ is the ith row of $\boldsymbol{G}^{-1}(\boldsymbol{\Sigma})$. This can be computed more efficiently because one needs to solve the quadratic program and invert $\boldsymbol{G}(\boldsymbol{\Sigma})$ and $\boldsymbol{G}(\boldsymbol{\Lambda})$ only once. Let $i^* = \arg\max_i v_{(i)}$. Now, the optimization in (19) can be performed in the neighborhood of $\boldsymbol{v}_{i^*}$. In other words, one only needs to find a local maxima of the prediction variance near $\boldsymbol{v}_{i^*}$, which is easy to do.

For illustrative purposes, consider the two-dimensional posterior density with banana-shaped contours discussed in Haario, Saksman, and Tamminen (2001):

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

Suppose a 100-run MmLHD from the region $[-20,20]\times[-10,5]$ is chosen as the initial space-filling design (shown as circles in Figure 9). The $\boldsymbol{\Sigma}$ and $\boldsymbol{\Lambda}$ can be estimated using cross-validation, as described before. The DoIt approximation of the posterior distribution is shown in Figure 9(a), which does not give a good fit to the exact distribution. The maximum value of $v_{(i)}$ happens at $\boldsymbol{v}_{25} = (2.56, 2.92)'$. Now, the new point to add is found using (19), where $\boldsymbol{v}_{25}$ is used as the starting point in the optimization algorithm. The algorithm converged to $(7.34, 1.44)'$, which could be a local optimum near $\boldsymbol{v}_{25}$. This is taken as the new point $\boldsymbol{v}_{101}$. This procedure can be continued. Figure 9(b), (c), and (d) shows the posterior distribution after adding 25, 50, and 75 points, respectively. One can clearly see the improvement in the approximation. Typically, the exact density will not be known, and therefore, the improvement should be monitored using some measures that can be computed. Here, it is proposed to monitor the leave-one-out cross-validation errors: $cv_i = h_i - \hat{h}_{(i)i}$. Note that cv_i is defined with respect to the DoIt approximation in (13) and is different from $e_i = h_i - \tilde{h}_{(i)i}$, which is defined using (5). Similar to (20), a shortcut formula for computing cv_i's can be obtained as (see the Appendix for details)

$$cv_i = h_i - \left(h_i + l_i - \frac{\boldsymbol{G}_i^{-1}(\boldsymbol{\Sigma})}{\boldsymbol{G}_{ii}^{-1}(\boldsymbol{\Sigma})}(\boldsymbol{h}+\boldsymbol{l})\right) \times \left(\frac{h_i}{h_i+l_i} - \frac{\boldsymbol{G}_i^{-1}(\boldsymbol{\Lambda})}{\boldsymbol{G}_{ii}^{-1}(\boldsymbol{\Lambda})}\left(\frac{\boldsymbol{h}}{\boldsymbol{h}+\boldsymbol{l}} - a\mathbf{1}\right)\right). \tag{21}$$

Define the percentage relative error to be

$$\%\text{RE} = \frac{\overline{|cv|}}{\bar{h}} \times 100,$$

where $\overline{|cv|} = E(|cv(\boldsymbol{\theta})|\,|\boldsymbol{y})$ and $\bar{h} = E(h(\boldsymbol{\theta})|\boldsymbol{y})$ are the average absolute cross-validation error and average height of the unnormalized posterior with respect to the posterior distribution, respectively. These quantities can be easily computed using (18). The %RE is plotted in Figure 10. At $m=100$, the relative error was 52%, which is reduced to 4% after adding 75 points. One can stop adding points when the relative error is less than an acceptable level. To check the effectiveness of the sequential design, a 175-run MmLHD is generated. The relative error of the corresponding DoIt approximation is found to be 27%, which is much larger than that of the $100+75$-run sequential design.

For comparison with the other deterministic approximation methods, the VB approximation of the posterior using the product density transform approach is computed. It is given by

$$\hat{p}_{\text{VB}}(\boldsymbol{\theta}|\boldsymbol{y}) = q_1(\theta_1)\phi\big(\theta_2;\mu_2,\sigma_2^2\big),$$

where $q_1(\theta_1) \propto \exp\{-0.5[\theta_1^2/100 + (\mu_2 + 0.03\theta_1^2 - 3)^2]\}$, $\mu_2 = -0.03(\mu_1^2+\sigma_1^2)+3$, and $\sigma_2^2 = 1$. The posterior mean (μ_1) and variance (σ_1^2) of θ_1 can be obtained through numerical

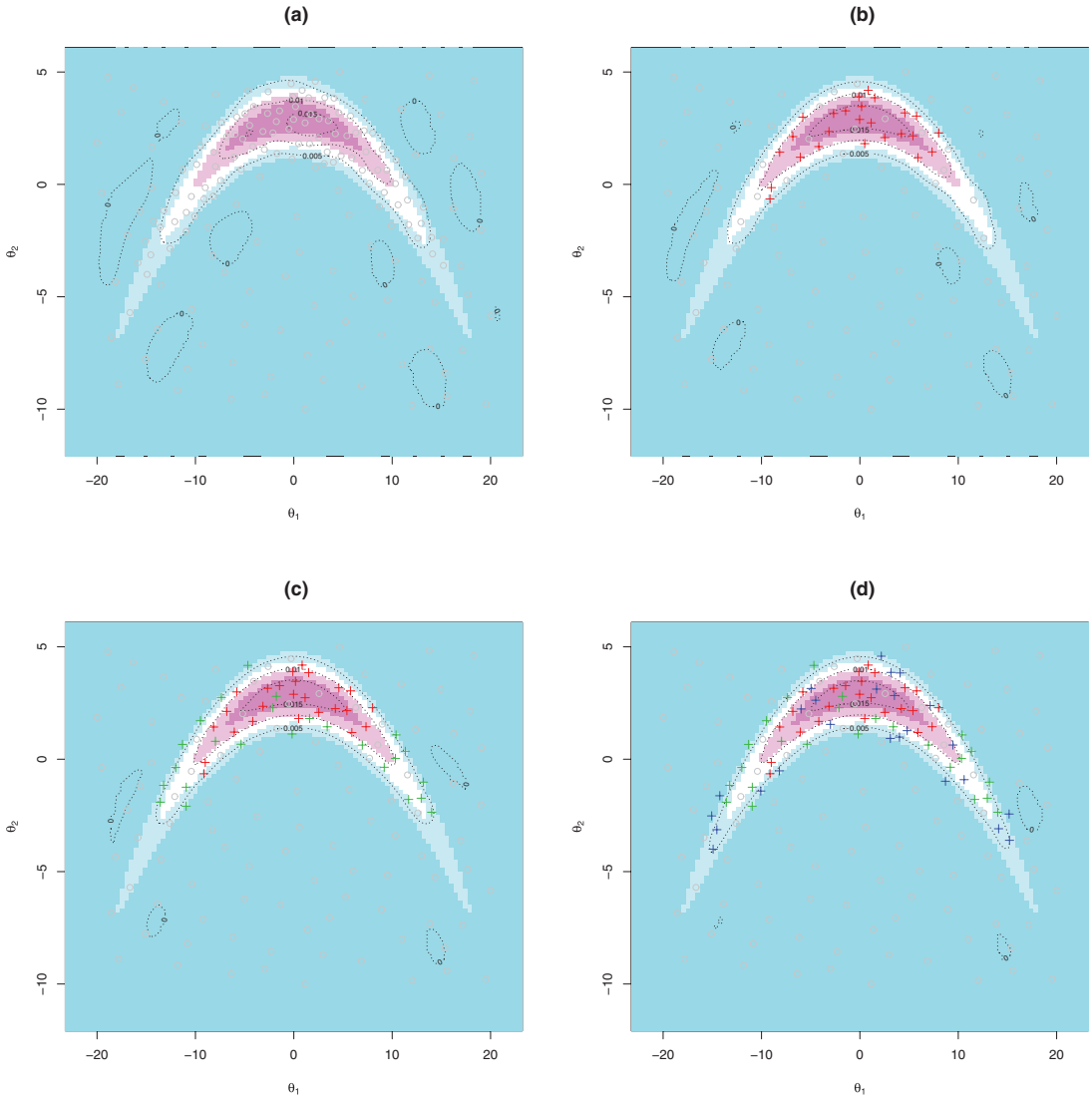

Figure 9. DoIt approximation (contour lines) superimposed over the image of the true posterior with (a) initial space-filling design ($m = 100$), (b) after adding 25 points ($m = 125$), (c) after adding 50 points ($m = 150$), and (d) after adding 75 points ($m = 175$). Added points are denoted with a "+." The online version of this figure is in color.

integration. The iterations quickly converges to the distribution shown in Figure 11(a). One can see that it is not a good approximation. This is due to the high correlation between the two parameters, which is ignored in the factorized solution of the VB method. I also ran, the hybrid MCMC algorithm of Fielding, Nott, and Liong (2011) using the R package `MCMChybridGP` (Fielding 2010). The same 100-run MmLHD was used as the initial sample and 500 samples were generated from the exploratory phase of the algorithm and another 1500 samples from the sampling phase of the algorithm. Figure 11(b) shows that the hybrid MCMC sampling is very good. However, it took about 90 min for this sampling, whereas DoIt took only about 3 min for the entire computation.

4. HIERARCHICAL MODELS

Hierarchical models create challenges in Bayesian computation due to the sheer number of parameters they may contain. Quadrature methods break down in solving them due to the curse of dimensionality. MCMC on the other hand, and in particular Gibbs sampling, is surprisingly efficient in solving such problems (Gelfand et al. 1990). DoIt is less affected by the curse of dimensionality because the evaluation points need not have to be on a regular grid, as in the lattice-based quadrature methods. However, finding a good space-filling design in high dimensions can still be a difficult task. Here, a method to efficiently sample the points and obtain the DoIt approximation by making use of a special probability structure of hierarchical models has been proposed.

Consider a hierarchical model $\boldsymbol{y}|\boldsymbol{\theta} \sim p(\boldsymbol{y}|\boldsymbol{\theta})$, $\boldsymbol{\theta}|\boldsymbol{\eta} \sim p(\boldsymbol{\theta}|\boldsymbol{\eta})$, and $\boldsymbol{\eta} \sim p(\boldsymbol{\eta})$. Suppose that one can obtain an explicit expression of

$$p(\boldsymbol{y}|\boldsymbol{\eta}) = \int p(\boldsymbol{y}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\boldsymbol{\eta})\, d\boldsymbol{\theta},$$

and that the conditional distribution

$$p(\boldsymbol{\theta}|\boldsymbol{\eta}, \boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\boldsymbol{\eta})$$

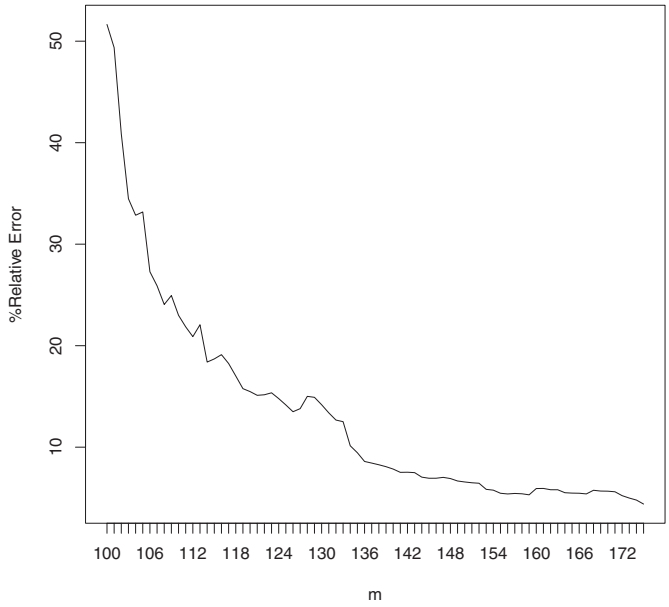

Figure 10. Percent relative error against the number of points added sequentially in the banana-shaped posterior example.

is known (i.e., it has a standard form). Let $h(\boldsymbol{\eta}) \propto p(\boldsymbol{y}|\boldsymbol{\eta})p(\boldsymbol{\eta})$. Now, using DoIt, one can approximate the posterior distribution of $\boldsymbol{\eta}$ as

$$\hat{p}(\boldsymbol{\eta}|\boldsymbol{y}) \approx \frac{\hat{\boldsymbol{c}}'\boldsymbol{\phi}(\boldsymbol{\eta};\boldsymbol{\Sigma})}{\hat{\boldsymbol{c}}'\mathbf{1}}\{1+\hat{\boldsymbol{b}}'\boldsymbol{g}(\boldsymbol{\eta};\boldsymbol{\Lambda})/a\}, \tag{22}$$

where the notations are defined as before. Now, the posterior distribution of $\boldsymbol{\theta}$ can be obtained using the formula in (18):

$$\hat{p}(\boldsymbol{\theta}|\boldsymbol{y}) \approx \frac{\hat{\boldsymbol{c}}'\boldsymbol{G}(\boldsymbol{\Sigma}+\boldsymbol{\Lambda})\boldsymbol{G}(\boldsymbol{\Lambda})^{-1}\boldsymbol{p}^*(\boldsymbol{\theta})}{\hat{\boldsymbol{c}}'\boldsymbol{G}(\boldsymbol{\Sigma}+\boldsymbol{\Lambda})\boldsymbol{G}(\boldsymbol{\Lambda})^{-1}\boldsymbol{z}}, \tag{23}$$

where $\boldsymbol{p}^*(\boldsymbol{\theta}) = (p(\boldsymbol{\theta}|\boldsymbol{v}_1,\boldsymbol{y}),\ldots,p(\boldsymbol{\theta}|\boldsymbol{v}_m,\boldsymbol{y}))' \odot \boldsymbol{z}$ and $\boldsymbol{z} = a\mathbf{1} + \boldsymbol{G}(\boldsymbol{\Lambda})\hat{\boldsymbol{b}}$. Although (18) is only an approximate formula, it gives a valid density here because $\int \boldsymbol{p}^*(\boldsymbol{\theta})d\boldsymbol{\theta} = \boldsymbol{z}$. Moreover, the posterior density is a weighted average of $p(\boldsymbol{\theta}|\boldsymbol{v}_i,\boldsymbol{y})$'s, and since this conditional distribution has a standard form, the required posterior summaries of $\boldsymbol{\theta}$ can be easily computed.

The advantage of the foregoing method is that one only needs to create a design $\boldsymbol{D} = \{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_m\}$ in the space of $\boldsymbol{\eta}$. The vector $\boldsymbol{\theta}$ may contain thousands of parameters, which cause no difficulty in the computation. In a more general setting of the hierarchical models, suppose one can group the parameters (and the hyperparameters) as $(\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_q)$ and that one can integrate out $\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_{q-1}$. Then, to apply DoIt, one needs to create a design in the space of $\boldsymbol{\theta}_q$. Therefore, DoIt works efficiently if the size of $\boldsymbol{\theta}_q$ is small.

4.1 A Longitudinal Data Analysis

As an example, consider the longitudinal study of orthodontic measurements on 27 children, reported by Pinheiro and Bates (2000), which was recently reanalyzed by Ormerod and Wand (2010) using VB methods. The study concerns the modeling of an orthodontic distance (y) measured on the children with respect to their age and sex. Consider the following random intercept model:

$$\begin{aligned} y_{ij}|\boldsymbol{\beta},u_i,\sigma_\epsilon^2 &\overset{\text{iid}}{\sim} N\big(\beta_0+u_i+\beta_1\text{age}_{ij}+\beta_2\text{sex}_i,\sigma_\epsilon^2\big),\\ u_i|\sigma_u^2 &\overset{\text{iid}}{\sim} N\big(0,\sigma_u^2\big), \end{aligned}$$

for $i=1,\ldots,27$ and $j=1,\ldots,4$. The prior specifications for the parameters are made as in Ormerod and Wand (2010): $\boldsymbol{\beta} \sim N(\mathbf{0},10^8\boldsymbol{I}_3)$ and $\sigma_\epsilon^2,\sigma_u^2 \overset{\text{ind.}}{\sim} IG(.01,.01)$. In this analysis, the sex variable is coded as 1 for male and -1 for female, and the age variable is centered to have mean 0.

There are a total of 32 parameters in this Bayesian model, including the random effects u_i's, the regression parameters β_i's, and the two variance components. A direct fitting of DoIt for such a high-dimensional problem can be challenging. Fortunately, one can integrate out the random effects, thereby reducing this to a five-dimensional problem. First, write the model in matrix notation: $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{u} + \boldsymbol{\epsilon}$, where $\boldsymbol{X}$ is the 108×3 regression model matrix and $\boldsymbol{Z}$ is the 108×27 indicator matrix for the random effects $\boldsymbol{u} = (u_1,\ldots,u_{27})'$. Integrating out $\boldsymbol{u}$, one obtains

$$\boldsymbol{y}|\boldsymbol{\beta},\sigma_\epsilon^2,\sigma_u^2 \sim N\big(\boldsymbol{X}\boldsymbol{\beta},\sigma_\epsilon^2\boldsymbol{I}_{108}+\sigma_u^2\boldsymbol{Z}\boldsymbol{Z}'\big).$$

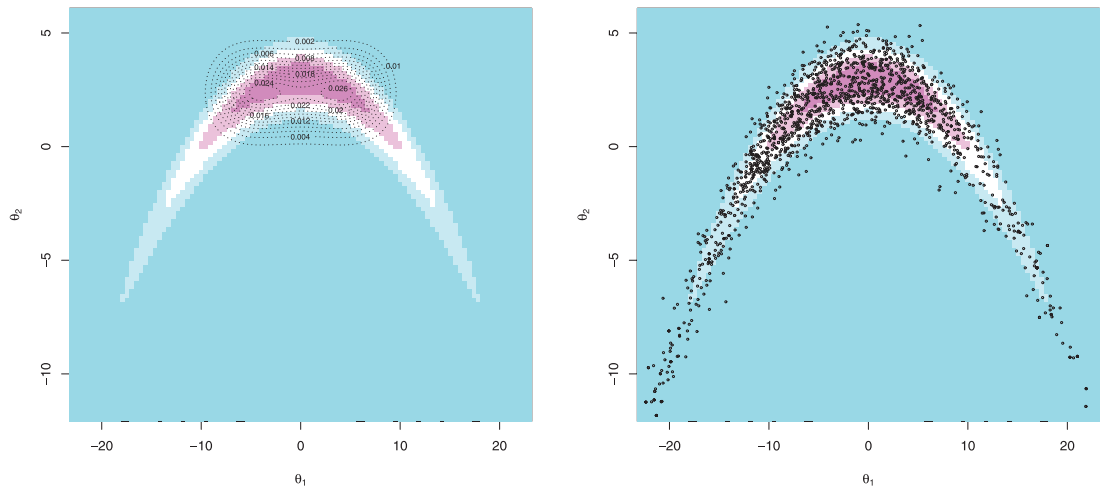

Figure 11. Comparison of different methods in the banana-shaped posterior example: (a) VB and (b) hybrid MCMC of Fielding, Nott, and Liong (2011). The online version of this figure is in color.

Also, one has

$$\boldsymbol{u}|\boldsymbol{\beta}, \sigma_\epsilon^2, \sigma_u^2, \boldsymbol{y} \sim N\left(\left(\boldsymbol{Z}'\boldsymbol{Z} + \frac{\sigma_\epsilon^2}{\sigma_u^2}\boldsymbol{I}_{27}\right)^{-1}\boldsymbol{Z}'(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}), \left(\boldsymbol{Z}'\boldsymbol{Z} + \frac{\sigma_\epsilon^2}{\sigma_u^2}\boldsymbol{I}_{27}\right)^{-1}\right).$$

This fits into the earlier discussion of applying DoIt to hierarchical models with $\boldsymbol{\eta} = (\boldsymbol{\beta}', \sigma_\epsilon^2, \sigma_u^2)'$ and $\boldsymbol{\theta} = \boldsymbol{u}$. Thus, one can obtain the posterior distributions using (22) and (23). DoIt was fitted using a 250-run space-filling design, and the marginal posterior distributions of one of the u_i's, β_i's, σ_ϵ^2, and σ_u^2 are shown in Figure 12. The density plots of 300,000 samples obtained using Gibbs sampling are also shown in the same plot. One can see that DoIt and Gibbs sampling are in good agreement. Now, consider the VB analysis. As shown in Ormerod and Wand (2010), the posterior density can be factorized into a product of a 30-dimensional multivariate normal density for the fixed and random effects, and two inverse gamma densities for the two variance components. The parameters of these densities can be obtained using the algorithm given in Ormerod and Wand. The resulting marginal posterior densities are plotted in Figure 12. One can see that the VB approximation is quite good for the regression model parameters β_i's and the random effects u_i's, but it gives a poor approximation for the two variance components.

4.2 A Computationally Expensive Posterior

As another example of hierarchical models, consider a computer experiment conducted to optimize a laser-assisted mechanical micromachining (LAMM) process. Four process variables, depth of cut (x_1), cutting speed (x_2), laser power (x_3), and distance between the laser and the cutting tool (x_4), are varied in the experiment using a $4 \times 2 \times 3 \times 2$ full factorial design (all the variables are scaled in -1 to 1). Many outputs are obtained in the experiment, but for illustrative purposes, here only the cutting force (y) has been analyzed. The details about the process and the experiment can be found in Singh, Joseph, and Melkote (2011). The computer model ($\theta(\boldsymbol{x})$) is computationally expensive, with each experimental run taking more than 12 hr of computer time. Thus, an easy-to-evaluate approximation of the computer model is required for predicting and optimizing

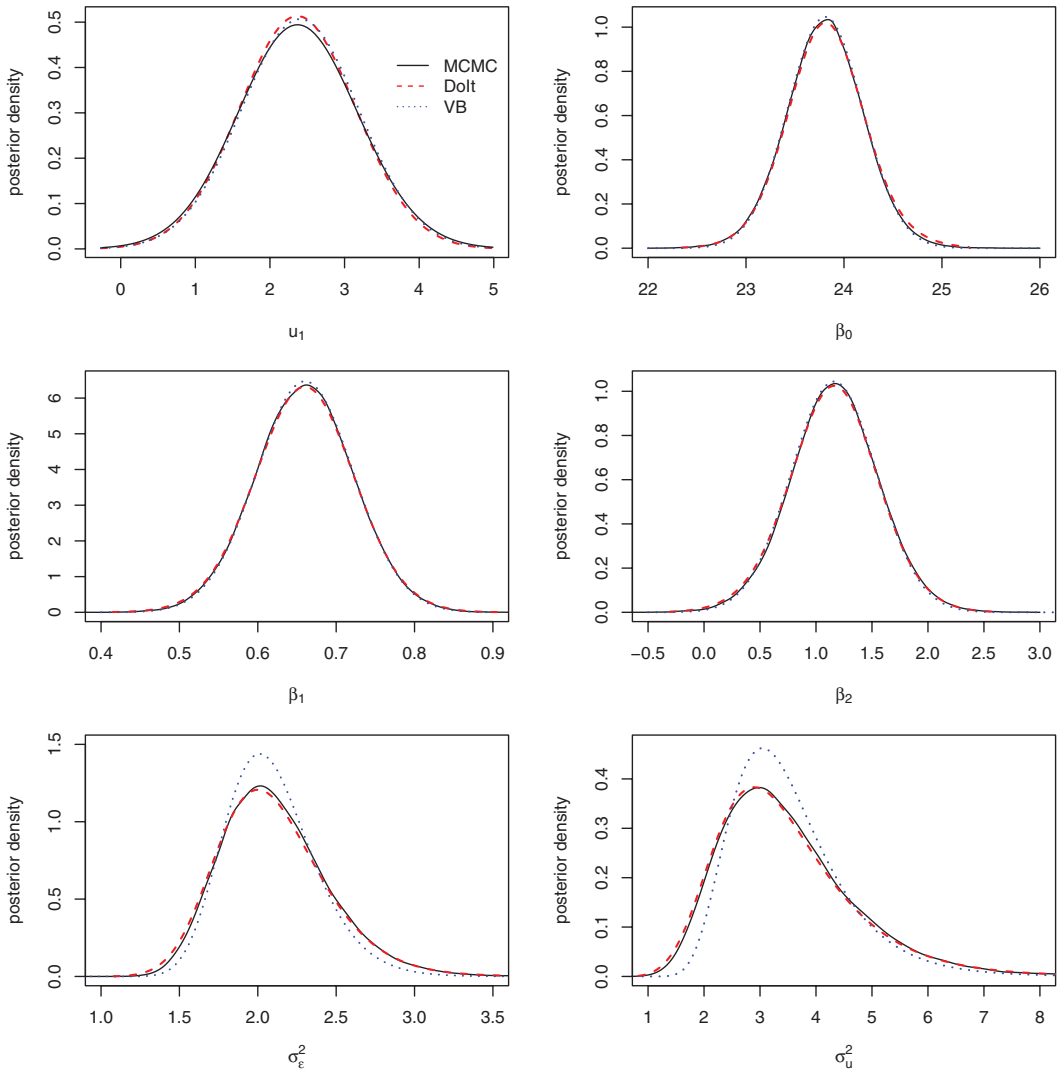

Figure 12. Comparison of DoIt with Gibbs sampling and VB in the orthodontic example. The online version of this figure is in color.

the cutting force. This is usually done using a GP model (Santner, Williams, and Notz 2003): $\theta(\boldsymbol{x}) \sim \text{GP}(\mu, \tau^2 r)$, which can be viewed as a prior on the underlying true computer model. Here, $r(\boldsymbol{x}_i, \boldsymbol{x}_j) = \text{cor}\{\theta(\boldsymbol{x}_i), \theta(\boldsymbol{x}_j)\}$ is the correlation function, which is taken as the Gaussian correlation function given by $\exp\{-\sum_{k=1}^{4} \alpha_k (x_{ik} - x_{jk})^2\}$. Assume a noninformative prior for the hyperparameters μ and τ^2: $p(\mu, \tau^2) \propto 1/\tau^2$. The priors on the correlation parameters are chosen as $\gamma_i = \log(\alpha_i) \overset{\text{iid}}{\sim} N(0, 1)$ for $i = 1, \ldots, 4$.

In this study, there are 48 observations from the computer model $\boldsymbol{y} = (y_1, \ldots, y_n)'$ at the locations specified by the full factorial design ($n = 48$). Thus, the joint likelihood is given by

$$\begin{aligned} h(\theta(\boldsymbol{x}), \mu, \tau^2, \boldsymbol{\gamma}) &\propto p(\boldsymbol{y}|\theta(\boldsymbol{x}), \mu, \tau^2, \boldsymbol{\gamma}) p(\theta(\boldsymbol{x})|\mu, \tau^2, \boldsymbol{\gamma}) \\ &\quad \times p(\mu, \tau^2) p(\boldsymbol{\gamma}), \\ &= p(\theta(\boldsymbol{x})|\mu, \tau^2, \boldsymbol{\gamma}, \boldsymbol{y}) p(\boldsymbol{y}|\mu, \tau^2, \boldsymbol{\gamma}) \\ &\quad \times p(\mu, \tau^2) p(\boldsymbol{\gamma}), \end{aligned} \tag{24}$$

where $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_4)'$. It is well known that

$$\begin{aligned} \theta(\boldsymbol{x})|\mu, \tau^2, \boldsymbol{\gamma}, \boldsymbol{y} \sim N\big(&\mu + \boldsymbol{r}(\boldsymbol{x})' \boldsymbol{R}^{-1}(\boldsymbol{y} - \mu \mathbf{1}), \tau^2\{1 - \boldsymbol{r}(\boldsymbol{x})' \\ &\times \boldsymbol{R}^{-1} \boldsymbol{r}(\boldsymbol{x})\}\big), \end{aligned}$$

where $\boldsymbol{r}(\boldsymbol{x})' = (r(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, r(\boldsymbol{x}, \boldsymbol{x}_n))$ and $\boldsymbol{R}$ is the $n \times n$ correlation matrix with ijth element $r(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Thus, integrating out $\theta(\boldsymbol{x})$ from (24), one obtains

$$h(\mu, \tau^2, \boldsymbol{\gamma}) = p(\boldsymbol{y}|\mu, \tau^2, \boldsymbol{\gamma}) p(\mu, \tau^2) p(\boldsymbol{\gamma}).$$

Now, one can fit DoIt. In fact, in this particular case, it is easy to integrate out μ and τ^2 as well. This reduction to a smaller space makes the choice of a design easier. Thus, one obtains

$$h(\boldsymbol{\gamma}) = |\boldsymbol{R}|^{-1/2} (\mathbf{1}' \boldsymbol{R}^{-1} \mathbf{1})^{-1/2} [(\boldsymbol{y} - \hat{\mu} \mathbf{1})' \boldsymbol{R}^{-1} (\boldsymbol{y} - \hat{\mu} \mathbf{1})]^{-(n-1)/2},$$

where the proportionality constant is omitted. One also needs the conditional distribution $p(\theta(\boldsymbol{x})|\boldsymbol{\gamma}, \boldsymbol{y})$, which can be obtained as (e.g., see Santner, Williams, and Notz 2003, p. 95)

$$\frac{\theta(\boldsymbol{x}) - \hat{\theta}(\boldsymbol{x})}{\sqrt{V(\boldsymbol{x})}} | \boldsymbol{\gamma}, \boldsymbol{y} \sim t_{n-1}, \tag{25}$$

where

$$\begin{aligned} \hat{\theta}(\boldsymbol{x}) &= \hat{\mu} + \boldsymbol{r}(\boldsymbol{x})' \boldsymbol{R}^{-1} (\boldsymbol{y} - \hat{\mu} \mathbf{1}), \\ V(\boldsymbol{x}) &= \hat{\tau}^2 \left(1 - \boldsymbol{r}(\boldsymbol{x})' \boldsymbol{R}^{-1} \boldsymbol{r}(\boldsymbol{x}) + \frac{\{1 - \boldsymbol{r}(\boldsymbol{x})' \boldsymbol{R}^{-1} \mathbf{1}\}^2}{\mathbf{1}' \boldsymbol{R}^{-1} \mathbf{1}} \right), \\ \hat{\mu} &= \frac{\mathbf{1}' \boldsymbol{R}^{-1} \boldsymbol{y}}{\mathbf{1}' \boldsymbol{R}^{-1} \mathbf{1}}, \text{ and} \\ \hat{\tau}^2 &= \frac{1}{n-1} (\boldsymbol{y} - \hat{\mu} \mathbf{1})' \boldsymbol{R}^{-1} (\boldsymbol{y} - \hat{\mu} \mathbf{1}). \end{aligned}$$

Now, one can fit DoIt to obtain $\hat{p}(\boldsymbol{\gamma}|\boldsymbol{y})$ and then use (23) to obtain $\hat{p}(\theta(\boldsymbol{x})|\boldsymbol{y})$.

As described in Section 3.1, an MmLHD of $m = 100$ points in the space of $\boldsymbol{\gamma}$ was chosen and the DoIt was fitted. The posterior distribution of $\theta(\boldsymbol{x})$ obtained from (23) is a weighted average of t-distributions given in (25). This can be used for predicting the cutting force at any $\boldsymbol{x}$ and quantifying its uncertainty. As an example, predictions at the three locations $\boldsymbol{x} = (-0.5, -0.5, -0.5, -0.5)'$, $\boldsymbol{x} = (0, 0, 0, 0)'$, and $\boldsymbol{x} = (0.5, 0.5, 0.5, 0.5)'$ are shown in Figure 13 (dashed lines). For comparison purposes, a Metropolis algorithm is run to obtain 100,000 posterior samples. The resulting posterior densities of the predictions are also plotted in Figure 13 (solid lines), which are in good agreement with the densities obtained using DoIt.

In this example, DoIt took about 3 sec on a 3.20-GHz computer, whereas the MCMC took about 10 min on the same computer. This is an example of a computationally intensive posterior because its calculation requires the inverse of the matrix $\boldsymbol{R}$, whose computational complexity is $O(n^3)$. Here, n was only 48. In many other computer experiments and spatial statistics problems, n can be much larger ($> 10,000$), where it is impractical to apply MCMC. Because of this computational hindrance, it is a common practice to avoid a fully Bayesian analysis by ignoring the variability in $\boldsymbol{\gamma}$. For example, the posterior densities of the three predictions after plugging in the posterior mode of $\boldsymbol{\gamma}$ in (25) are shown in Figure 13 (dotted lines). One can see that the plug-in approach underestimates the prediction uncertainties. This example clearly demonstrates the advantages of using DoIt, because it can easily incorporate these uncertainties even in problems with large datasets.

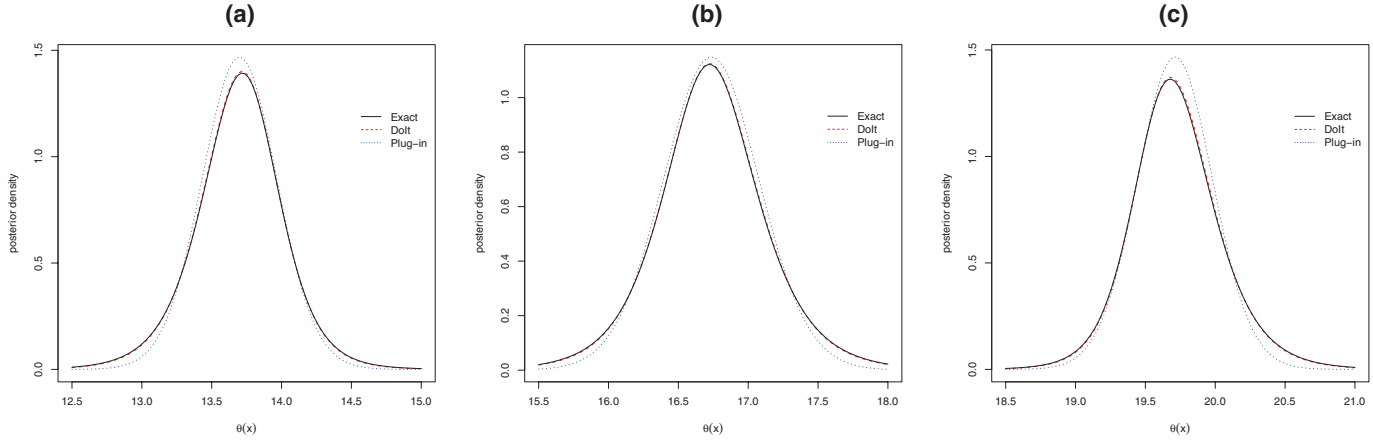

Figure 13. Posterior distribution of the cutting force predictions at (a) $\boldsymbol{x} = (-0.5, -0.5, -0.5, -0.5)'$, (b) $\boldsymbol{x} = (0, 0, 0, 0)'$, and (c) $\boldsymbol{x} = (0.5, 0.5, 0.5, 0.5)'$ using DoIt and MCMC, and with $\boldsymbol{\gamma} = \hat{\boldsymbol{\gamma}}$ (plug-in). The online version of this figure is in color.

5. CONCLUSIONS

In this article, a method known as design of experiments-based interpolation technique (DoIt) has been described for approximating continuous posterior distributions using normal-like basis functions. Thus, the method can be considered as an extension of the Laplace method. The method is much more general and flexible in the sense that it does not require the posterior to be unimodal or differentiable. Moreover, as the number of basis functions increases, the approximation becomes better, and therefore, unlike the Laplace method, DoIt is capable of approximating the posterior with any desired precision. This is a great advantage over other deterministic approximation methods, such as VB, EP, and INLA. Moreover, DoIt can be implemented almost as a black box method and can be easily adapted to many Bayesian problems. Here, it is shown through many examples that DoIt, with fewer posterior evaluations, can produce comparable accuracy to those produced by the MC/MCMC methods. This is especially useful when the posterior is expensive to evaluate or when it needs to be evaluated many times as part of some external algorithms.

However, DoIt does not seem to be as flexible as the MCMC methods, particularly in fitting hierarchical models. As alluded before, when the parameters are grouped as $(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_q)$, DoIt can be efficient only if most parameters can be integrated out, leaving only a small set of parameters (say, $\boldsymbol{\theta}_q$). The method also requires the conditional distributions $p(\boldsymbol{\theta}_i|\boldsymbol{\theta}_q, \boldsymbol{y})$ to be available, whereas a Gibbs sampling algorithm requires only the full conditional distributions $p(\boldsymbol{\theta}_i|\{\boldsymbol{\theta}_j\}_{j\neq i}, \boldsymbol{y})$ to be available, which in most hierarchical models are much easier to obtain. Nevertheless, DoIt can solve many of the hierarchical model problems when conjugate prior distributions are used. Moreover, its applicability to a wider class of problems is possible if more efficient methods for design construction that make use of the hierarchical model structure can be developed. Furthermore, in this article, the focus has been on normal-like basis functions. However, as with kriging, it is possible to use other basis functions. If the method can be further extended to incorporate some of the conditional distributions as bases, then many of the hierarchical model problems can be solved even more efficiently.

The quality of kriging approximation depends on the fill distance of the space-filling design (Haaland and Qian 2011). As the dimensions increase, the fill distance increases unless the number of designs points are also increased. However, finding a large space-filling design in a higher-dimensional space is a difficult task. Furthermore, computational difficulties and numerical errors also increase as the the number of points increases. Therefore, the DoIt approximation tends to deteriorate as the dimension increases. As a result, apart from some of the hierarchical models, DoIt seems to be capable of handling only small-to-moderately large number of dimensions, whereas the other methods, such as VB, EP, and INLA, have shown to be capable of handling hundreds or thousands of dimensions. The strength of DoIt relative to such approximation methods is in delivering fast and accurate approximations for moderate-dimensional posterior distributions. Moreover, VB and EP methods work well only under some restrictive assumptions about the form of the posterior such as that it can be factorized into a product of marginals. It will be interesting to see if DoIt can also be extended to deal with high dimensions by invoking similar assumptions. This is left as a topic for future research.

APPENDIX: PROOFS

Proof of Theorem 1

Let $I = \int h(\boldsymbol{\theta})d\boldsymbol{\theta}$ and $\boldsymbol{\Theta}$ the $(1-\alpha)$ HPD credible set of the posterior distribution for $\alpha \in (0, 1)$. Then, there exists $\kappa_\alpha > 0$ such that $h(\boldsymbol{\theta}) \geq \kappa_\alpha I$ for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, and $\int_{\boldsymbol{\Theta}} h(\boldsymbol{\theta})d\boldsymbol{\theta} = (1-\alpha)I$. Because $\boldsymbol{\Theta}$ is a closed set, for any $r > 0$, there exists a finite number of balls (m) with radius r that can cover $\boldsymbol{\Theta}$. Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$ be the center of these m balls. Consider another continuous function $\hat{h}(\boldsymbol{\theta})$ that interpolates $h(\boldsymbol{\theta})$ on the m points. Because $h(\boldsymbol{\theta}) \geq \kappa_\alpha I$ for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, $\hat{h}(\boldsymbol{\theta})/h(\boldsymbol{\theta})$ is also a continuous function on $\boldsymbol{\Theta}$. Moreover, $\hat{h}(\boldsymbol{v}_i)/h(\boldsymbol{v}_i) = 1$ for all $i = 1, \ldots, m$. Therefore, for any radius $r > 0$ one can find an $\epsilon' > 0$ such that $|\hat{h}(\boldsymbol{\theta})/h(\boldsymbol{\theta}) - 1| < \epsilon'$ for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. Since $\hat{h}(\boldsymbol{\theta})$ is uniformly convergent, one can choose a radius $r > 0$ small enough (and m large enough) so that $\epsilon' < 1$. Thus, $0 < (1-\epsilon')h(\boldsymbol{\theta}) < \hat{h}(\boldsymbol{\theta}) < (1+\epsilon')h(\boldsymbol{\theta})$ for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, which implies

$$0 < (1-\epsilon')(1-\alpha)I < \int_{\boldsymbol{\Theta}} \hat{h}(\boldsymbol{\theta})d\boldsymbol{\theta} < (1+\epsilon')(1-\alpha)I.$$

Thus,

$$\frac{\hat{h}(\boldsymbol{\theta})\int_{\boldsymbol{\Theta}} h(\boldsymbol{\theta})d\boldsymbol{\theta}}{h(\boldsymbol{\theta})\int_{\boldsymbol{\Theta}} \hat{h}(\boldsymbol{\theta})d\boldsymbol{\theta}} < \frac{(1+\epsilon')h(\boldsymbol{\theta})(1-\alpha)I}{h(\boldsymbol{\theta})(1-\epsilon')(1-\alpha)I} = \frac{1+\epsilon'}{1-\epsilon'}.$$

Similarly,

$$\frac{\hat{h}(\boldsymbol{\theta})\int_{\boldsymbol{\Theta}} h(\boldsymbol{\theta})d\boldsymbol{\theta}}{h(\boldsymbol{\theta})\int_{\boldsymbol{\Theta}} \hat{h}(\boldsymbol{\theta})d\boldsymbol{\theta}} > \frac{(1-\epsilon')h(\boldsymbol{\theta})(1-\alpha)I}{h(\boldsymbol{\theta})(1+\epsilon')(1-\alpha)I} = \frac{1-\epsilon'}{1+\epsilon'}.$$

Thus,

$$\frac{-2\epsilon'}{1+\epsilon'} < \frac{\hat{h}(\boldsymbol{\theta})\int_{\boldsymbol{\Theta}} h(\boldsymbol{\theta})d\boldsymbol{\theta}}{h(\boldsymbol{\theta})\int_{\boldsymbol{\Theta}} \hat{h}(\boldsymbol{\theta})d\boldsymbol{\theta}} - 1 < \frac{2\epsilon'}{1-\epsilon'},$$

which implies

$$\left|\frac{\hat{h}(\boldsymbol{\theta})\int_{\boldsymbol{\Theta}} h(\boldsymbol{\theta})d\boldsymbol{\theta}}{h(\boldsymbol{\theta})\int_{\boldsymbol{\Theta}} \hat{h}(\boldsymbol{\theta})d\boldsymbol{\theta}} - 1\right| < \frac{2\epsilon'}{1-\epsilon'},$$

for all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. Now, the theorem is proved by letting $\epsilon = 2\epsilon'/(1-\epsilon')$.

Proof of Equation (20)

Let $h^+(\boldsymbol{\theta}) = \boldsymbol{g}(\boldsymbol{\theta}; \boldsymbol{\Sigma})'\hat{\boldsymbol{c}}$, where $\hat{\boldsymbol{c}}$ is obtained by minimizing $(\boldsymbol{h} - \boldsymbol{G}(\boldsymbol{\Sigma})\boldsymbol{c})'\boldsymbol{G}^{-1}(\boldsymbol{\Sigma})(\boldsymbol{h} - \boldsymbol{G}(\boldsymbol{\Sigma})\boldsymbol{c})$, subject to $\boldsymbol{c} \geq 0$. By Kuhn–Tucker conditions: $\boldsymbol{G}(\boldsymbol{\Sigma})\boldsymbol{c} = \boldsymbol{h} + \boldsymbol{l}$, $\boldsymbol{c} \geq 0$, $\boldsymbol{l} \geq 0$, and $c_i l_i = 0$ for $i = 1, \ldots, m$, where $\boldsymbol{l} = (l_1, \ldots, l_m)'$ are the Lagrangian multipliers. Thus, if $\hat{\boldsymbol{c}}$ is the solution of the quadratic program, then $\boldsymbol{l} = \boldsymbol{G}(\boldsymbol{\Sigma})\hat{\boldsymbol{c}} - \boldsymbol{h}$. Suppose the ith point is removed. Then, $\hat{\boldsymbol{c}}_{(i)} = \boldsymbol{G}^{-1}_{(i)}(\boldsymbol{\Sigma})(\boldsymbol{h}_{(i)} + \hat{\boldsymbol{l}}_{(i)})$. Assume that the change in $\boldsymbol{l}$ due to the removal of the ith point is negligible. Then, one can approximate $\hat{\boldsymbol{l}}_{(i)} \approx \boldsymbol{l}_{(i)}$, which gives $\hat{\boldsymbol{c}}_{(i)} \approx \boldsymbol{G}^{-1}_{(i)}(\boldsymbol{\Sigma})(\boldsymbol{h}_{(i)} + \boldsymbol{l}_{(i)})$. Also assume that $\hat{\boldsymbol{c}}_{(i)} \geq 0$ and $\hat{c}_{(i)j} l_j = 0$ for

all $j \neq i$. Thus,

$$h^{+}_{(i)}(\boldsymbol{v}_i) \approx \boldsymbol{g}_{(i)}(\boldsymbol{v}_i;\boldsymbol{\Sigma})'\hat{\boldsymbol{c}_{(i)}} = \boldsymbol{g}_{(i)}(\boldsymbol{v}_i;\boldsymbol{\Sigma})'\boldsymbol{G}^{-1}_{(i)}(\boldsymbol{\Sigma})(\boldsymbol{h}_{(i)}+\boldsymbol{l}_{(i)})$$
$$= -\frac{\boldsymbol{G}^{-1}_{(i)i}(\boldsymbol{\Sigma})}{\boldsymbol{G}^{-1}_{ii}(\boldsymbol{\Sigma})}(\boldsymbol{h}_{(i)}+\boldsymbol{l}_{(i)}) = h_i + l_i - \frac{\boldsymbol{G}^{-1}_{i}(\boldsymbol{\Sigma})}{\boldsymbol{G}^{-1}_{ii}(\boldsymbol{\Sigma})}(\boldsymbol{h}+\boldsymbol{l}).$$

Also, $1 - \boldsymbol{g}_{(i)}(\boldsymbol{v}_i;\boldsymbol{\Lambda})'\boldsymbol{G}^{-1}_{(i)}(\boldsymbol{\Lambda})\boldsymbol{g}_{(i)}(\boldsymbol{v}_i;\boldsymbol{\Lambda}) = 1/\boldsymbol{G}^{-1}_{ii}(\boldsymbol{\Lambda})$. Substituting them in $v_{(i)} = (h^{+}_{(i)}(\boldsymbol{v}_i))^2\{1 - \boldsymbol{g}_{(i)}(\boldsymbol{v}_i;\boldsymbol{\Lambda})'\boldsymbol{G}^{-1}_{(i)}(\boldsymbol{\Lambda})\boldsymbol{g}_{(i)}(\boldsymbol{v}_i;\boldsymbol{\Lambda})\}$, one obtains the desired result.

Proof of Equation (21)

Following the proof of (20), one has for $j \neq i$

$$h^{+}_{(i)}(\boldsymbol{v}_j) \approx \boldsymbol{g}_{(i)}(\boldsymbol{v}_j;\boldsymbol{\Sigma})'\hat{\boldsymbol{c}_{(i)}} = \boldsymbol{g}_{(i)}(\boldsymbol{v}_j;\boldsymbol{\Sigma})'\boldsymbol{G}^{-1}_{(i)}(\boldsymbol{\Sigma})(\boldsymbol{h}_{(i)}+\boldsymbol{l}_{(i)})$$
$$= h_j + l_j.$$

Assume that $a_{(i)} \approx a$. Then,

$$\hat{h}_{(i)}(\boldsymbol{v}_i) = h^{+}_{(i)}(\boldsymbol{v}_i)\left\{a + \boldsymbol{G}^{-1}_{(i)}(\boldsymbol{\Lambda})\left(\frac{\boldsymbol{h}_{(i)}}{h^{+}_{(i)}} - a\boldsymbol{1}\right)\right\}$$
$$\approx h^{+}_{(i)}(\boldsymbol{v}_i)\left\{a - \frac{\boldsymbol{G}^{-1}_{(i)i}(\boldsymbol{\Lambda})}{\boldsymbol{G}^{-1}_{ii}(\boldsymbol{\Lambda})}\left(\frac{\boldsymbol{h}_{(i)}}{\boldsymbol{h}_{(i)}+\boldsymbol{l}_{(i)}} - a\boldsymbol{1}\right)\right\}$$
$$= \left(h_i + l_i - \frac{\boldsymbol{G}^{-1}_{i}(\boldsymbol{\Sigma})}{\boldsymbol{G}^{-1}_{ii}(\boldsymbol{\Sigma})}(\boldsymbol{h}+\boldsymbol{l})\right)$$
$$\times \left\{\frac{h_i}{h_i+l_i} - \frac{\boldsymbol{G}^{-1}_{i}(\boldsymbol{\Lambda})}{\boldsymbol{G}^{-1}_{ii}(\boldsymbol{\Lambda})}\left(\frac{\boldsymbol{h}}{\boldsymbol{h}+\boldsymbol{l}} - a\boldsymbol{1}\right)\right\}.$$

Substituting this in $cv_i = h_i - \hat{h}_{(i)}(\boldsymbol{v}_i)$, one obtains (21).

SUPPLEMENTARY MATERIAL

R codes and data files: The R codes and data files can be downloaded as a .zip file.

ACKNOWLEDGMENTS

The author thanks the editor, four referees, and Mr. Rui Tuo for their valuable comments and suggestions. This research was supported by the U.S. National Science Foundation grant CMMI-1030125.

[Received September 2010. Revised February 2012.]

REFERENCES

Bishop, C. M. (2006), *Pattern Recognition and Machine Learning*, New York: Springer. [209,214]

Bliznyuk, N., Ruppert, D., Shoemaker, C., Regis, R., Wild, S., and Mugunthan, P. (2008), "Bayesian Calibration and Uncertainty Analysis for Computationally Expensive Models Using Optimization and Radial Basis Function Approximation," *Journal of Computational and Graphical Statistics*, 17, 270–294. [215]

Bornkamp, B. (2011a), "Approximating Probability Densities by Iterated Laplace Approximations," *Journal of Computational and Graphical Statistics*, 20, 656–669. [212]

——— (2011b), "iterLap: Iterated Laplace approximations, (R package version 1.0-2)," Available at *http://cran.r-project.org/src/contrib/Archive/iterLap/* [212]

Brooks, S., Gelman, A., Jones, G. L., and Meng, X.-L. (2011), *Handbook of Markov Chain Monte Carlo*, Boca Raton, FL: CRC Press. [209]

Buhmann, M. D. (2003), *Radial Basis Functions: Theory and Implementations*, Cambridge: Cambridge University Press. [211]

Carnell, R. (2009), "lhs: Latin hypercube samples (R package version 0.5)," Available at *http://cran.r-project.org/src/contrib/Archive/lhs/* [216]

Cohn, D. A. (1996), "Neural Network Exploration Using Optimal Experimental Design," *Advances in Neural Information Processing Systems*, 6, 679–686. [217]

Dasgupta, T., Weintraub, B., and Joseph, V. R. (2011), "A Physical-Statistical Model for Density Control of Zinc Oxide Nanowires," *IIE Transactions on Quality and Reliability Engineering*, 43, 233–241. [216]

Fedorov, V. V. (1972), *Theory of Optimal Experiments*, New York: Academic Press. [217]

Fielding, M. (2010), "MCMChybridGP: Hybrid Markov chain Monte Carlo using Gaussian processes (R package version 3.1)," Available at *http://cran.r-project.org/* [218]

Fielding, M., Nott, D. J., and Liong, S-Y (2011), "Efficient MCMC Schemes for Computationally Expensive Posterior Distributions," *Technometrics*, 53, 16–28. [215,218,219]

Gelfand, A. E., Hills, S. E., Racine-Poon, A., and Smith, A. F. M. (1990), "Illustration of Bayesian Inference in Normal Data Models Using Gibbs Sampling," *Journal of the American Statistical Association*, 85, 972–985. [209]

Gelfand, A. E., and Smith, A. F. M. (1990), "Sampling-Based Approaches to Calculating Marginal Densities," *Journal of the American Statistical Association*, 85, 398–409. [219]

Geman, S., and Geman, D. (1984), "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721–741. [209]

Gramacy, R. B., and Lee, H. K. H. (2009), "Adaptive Design and Analysis of Supercomputer Experiments," *Technometrics*, 51, 130–145. [217]

Haaland, B., and Qian, P. Z. G. (2011), "Accurate Emulators for Large-Scale Computer Experiments," *The Annals of Statistics*, 39, 2974–3002. [223]

Haario, H., Saksman, E., and Tamminen, J. (2001), "An Adaptive Metropolis Algorithm," *Bernoulli*, 7, 223–242. [218]

Hastings, W. K. (1970), "Monte Carlo Sampling Methods Using Markov Chain and Their Applications," *Biometrika*, 87, 97–109. [209]

Henderson, D. A., Boys, R. J., Krishnan, K. J., Lawless, C., and Wilkinson, D. J. (2009), "Bayesian Emulation and Calibration of a Stochastic Computer Model of Mitochondrial DNA Deletions in Substantia Nigra Neurons," *Journal of the American Statistical Association*, 104, 76–87. [215]

Jaakkola, T. S., and Jordan, M. I. (2000), "Bayesian Parameter Estimation via Variational Methods," *Statistics and Computing*, 10, 25–37. [214]

Johnson, S. G., and Narasimhan, B. (2009), "cubature: Adaptive multivariate integration over hypercubes (R package version 1.0)," Available at *http://cran.r-project.org/* [216]

Joseph, V. R. (2006), "Limit Kriging," *Technometrics*, 48, 458–466. [212,214]

Kennedy, M. (1998), "Bayesian Quadrature With Non-Normal Approximating Functions," *Statistics and Computing*, 8, 365–375. [209,214]

Kuss, M., and Rasmussen, C. E. (2005), "Assessing Approximate Inference for Binary Gaussian Process Classification," *Journal of Machine Learning Research*, 6, 1679–1704. [214]

Loeppky, J. L., Sacks, J., and Welch, W. J. (2009), "Choosing the Sample Size of a Computer Experiment: A Practical Guide," *Technometrics*, 51, 366–376. [216]

MacKay, D. J. C. (1992), "Information-Based Objective Functions for Active Data Selection," *Neural Computation*, 4, 590–604. [217]

Marin, J-M, and Robert, C. P. (2007), *Bayesian Core: A Practical Approach to Computational Bayesian Statistics*, New York: Springer. [213]

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953), "Equation of State Calculation by Fast Computing Machines," *Journal of Chemical Physics*, 21, 1087–1092. [209]

Minka, P. (2001), "Expectation Propagation for Approximate Bayesian Inference," *Uncertainty in Artificial Intelligence*, 17, 362–369. [209,214]

Morris, M. D., and Mitchell, T. J. (1995), "Exploratory Designs for Computer Experiments," *Journal of Statistical Planning and Inference*, 43, 381–402. [216]

Naylor, J. C., and Smith, A. F. M. (1982), "Applications of a Method for the Efficient Computation of Posterior Distributions," *Applied Statistics*, 31, 214–225. [209]

O'Hagan, A. (1991), "Bayes-Hermite Quadrature," *Journal of Statistical Planning and Inference*, 29, 245–260. [209,214]

Ormerod, J. T., and Wand, M. P. (2010), "Explaining Variational Approximations," *The American Statistician*, 64, 140–153. [214,220,221]

——— (2012), "Gaussian Variational Approximate Inference for Generalized Linear Mixed Models," *Journal of Computational and Graphical Statistics*, 21(1), 2–17. [216]

Pinheiro, J. C., and Bates, D. M. (2000), *Mixed-Effects Models in S and S-PLUS*, New York: Springer. [220]

Rasmussen, C. E. (2003), "Gaussian Processes to Speed Up Hybrid Monte Carlo for Expensive Bayesian Integrals," in *Bayesian Statistics 7*, eds. J. M.

Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West, pp. 651–659. [215]

Rasmussen, C. E., and Ghahramani, Z. (2003), "Bayesian Monte Carlo," in *Advances in Neural Information Processing Systems* (Vol. 15), eds. S. T. S. Becker, and K. Obermayer, Cambridge, MA: MIT Press, pp. 489–496. [209,214]

Rasmussen, C. E., and Williams, C. K. I. (2006), *Gaussian Processes for Machine Learning*, Cambridge, MA: MIT Press. [210,211]

Rue, H., Martino, S., and Chopin, N. (2009), "Approximate Bayesian Inference for Latent Gaussian Models by Using Integrated Nested Laplace Approximations" (with discussion), *Journal of the Royal Statistical Society,* Series B, 71, 319–392. [209,214]

Santner, T. J., Williams, B. J., and Notz, W. I. (2003), *The Design and Analysis of Computer Experiments*, New York: Springer. [209,210,213,216,221,222]

Singh, R. K., Joseph, V. R., and Melkote, S. N. (2011), "A Statistical Approach to the Optimization of a Laser-Assisted Micromachining Process," *International Journal of Advanced Manufacturing Technology*, 53, 221–230. [221]

Tanner, M., and Wong, W. (1987), "The Calculation of Posterior Distributions by Data Augmentation" (with discussion), *Journal of the American Statistical Association*, 82, 528–550. [209]

Tierney, L., and Kadane, J. B. (1986), "Accurate Approximations for Posterior Moments and Marginal Densities," *Journal of the American Statistical Association*, 81, 82–86. [209]

Comment: Comparison With Iterated Laplace Approximation

Björn BORNKAMP

Novartis Pharma AG Lichtstraße 35
CH-4056 Basel, Switzerland
(*bjoern.bornkamp@novartis.com*)

1. INTRODUCTION

When using standard kriging for interpolation of positive functions (such as a nonnormalized posterior density), one either ends up with an interpolant that can get negative but can be integrated analytically, or, when working on log scale, a positive interpolant that cannot be integrated analytically anymore. The proposed interpolation technique provides a solution to this dilemma as the interpolant is (practically) positive *and* one can calculate a number of integrals analytically (normalization constant, marginal densities, mean, covariance matrix). This is probably the most important contribution of this article.

One area where the procedure appears to be promising in terms of approximation of Bayesian posterior densities is nonlinear modeling in low-to-moderate dimensional situations. Posterior distributions for these models have little general structure to exploit, so it makes sense to use flexible algorithms that treat the posterior density like a black box. The procedure seems to be particularly attractive for computationally intensive nonlinear models (e.g., defined implicitly as solution of a differential equation that needs to be solved numerically): a clever choice of evaluation points allows that few evaluations evaluations seem to be required to obtain an adequate approximation. For example, the sequential algorithm illustrated in the banana example in section 3.2 of Professor Joseph's paper obtains a good approximation with spectacularly few target density evaluations.

I would like to focus my discussion on a comparison with the iterated Laplace approximation (*iterLap*) (Bornkamp 2011a,b), which is an alternative deterministic approximation technique.

2. COMPARISON WITH ITERATED LAPLACE APPROXIMATION

Although derived from different perspectives, the two methods share a few similarities: the iterated Laplace approximation (*iterLap*) also uses linear combinations of multivariate normals (to approximate the *nonnormalized* posterior), low-discrepancy point sets (aka space-filling designs), quadratic programming, and sequential exploration of the target density. In a nutshell, the *iterLap* starts with a (multiple mode) Laplace approximation and then sequentially improves this approximation by adding additional normal distribution components, where the current approximation is worst. A more detailed description is as follows:

Iterated Laplace Approximation (*iterLap*)
Iteration 0:

1. *Multiple-mode Laplace approximation*: Fit a Laplace approximation to each mode of $h(\theta)$, resulting in an approximation based on a linear combination of multivariate normals: $\tilde{h}_0(\theta) = \sum_{j=1}^{J^{(0)}} c_j \phi(\theta, \nu_j, \Sigma_j)$; see (Gelman et al. 2003, chap. 12).
2. *Space-filling design*: Determine for each component in the linear combination a "grid" of size n that encloses most of its probability mass, using a quasi-random sample of the underlying multivariate normal distribution based on the

TECHNOMETRICS, AUGUST 2012, VOL. 54, NO. 3
DOI: 10.1080/00401706.2012.697246

randomized Sobol sequence. This is the same approach as in section 3.1 for the nanowire example but replacing the Latin hypercube design with the Sobol sequence. Let $\mathbf{D}_0$ denote the matrix that contains these points in the rows.

3. *Evaluation of* $h(.)$: Evaluate $h(.)$ at $\mathbf{D}_0$, resulting in the vector $\mathbf{h}_0$. Also, evaluate each of the $J^{(0)}$ component densities in the mixture at $\mathbf{D}_0$ and write those evaluations in the matrix $\mathbf{F}_0$.

Iteration t:

1. *Residual Laplace approximation*: In this step, a Laplace approximation on the residual $r_t(\theta) = (h(\theta) - \tilde{\mathbf{h}}_{t-1}(\theta))_+$ is performed to obtain a new mixture component. Select starting values, where $h(\theta)/\tilde{\mathbf{h}}_{t-1}(\theta)$ for $\theta \in \mathbf{D}_{t-1}$ is largest (i.e., the fit is worst). Start a local optimizer there, resulting in a maximum $\tilde{\theta}$. Add the new mixture component, with $J^{(t)} \leftarrow J^{(t-1)} + 1, \boldsymbol{\nu}_{J^{(t)}} = \tilde{\theta}$ and $\boldsymbol{\Sigma}_{J^{(t)}}$, the negative inverse Hessian matrix of $r_t(\theta)$ at $\tilde{\theta}$.
2. *Space-filling design*: Determine a grid $\mathbf{N}_t$ of size n for the new component (as in Step 2 at iteration 0). Add these points to the current grid $\mathbf{D}_{t-1}$ to form $\mathbf{D}_t = \binom{\mathbf{D}_{t-1}}{\mathbf{N}_t}$.
3. *Evaluation of* $h(.)$: Evaluate $h(.)$ at $\mathbf{N}_t$ and append these evaluations to $\mathbf{h}_{t-1}$ to form $\mathbf{h}_t$. Evaluate all components of the approximation $\tilde{h}_{t-1}(.)$ at $\mathbf{N}_t$ and the new component at the entire grid $\mathbf{D}_t$ to form $\mathbf{F}_t$.
4. *Quadratic programming*: Find the coefficients $c_1, \ldots, c_{J^{(t)}}$ by minimizing $(\mathbf{h}_t - \mathbf{F}_t'\mathbf{c})'(\mathbf{h}_t - \mathbf{F}_t'\mathbf{c})$ subject to $c_j \geq 0$ for $j = 1, \ldots, J^{(t)}$. The current approximation of $h(\theta)$ is $\tilde{\mathbf{h}}_t(\theta) = \sum_{j=1}^{J^{(t)}} c_j \phi(\theta, \boldsymbol{\nu}_j, \boldsymbol{\Sigma}_j)$, and the current approximation of the normalization constant is $\sum_{j=1}^{J^{(t)}} c_j$.

See Bornkamp (2011a) for more details and recommendations on how to choose stopping criteria and other tuning parameters. The procedure has, for example, been successfully applied in Bayesian calibration of computer models using Gaussian process interpolation; see Kracker (2011).

A main difference is that *iterLap* is meant to be used as a proposal distribution for Monte Carlo sampling. The procedure generates a positive linear combination of multivariate normal densities, which can be normalized to form a mixture of multivariate normals and is thus easy to sample. It will in general not approximate the posterior density arbitrarily well for a large number of iterations, as it only adds new mixture components where $h(\theta) - \tilde{\mathbf{h}}_t(\theta)$ is positive (which is important to obtain a good proposal distribution), but not where $\tilde{\mathbf{h}}_t(\theta) - h(\theta)$ is positive. This would require usage of negative coefficients and/or additional alterations of the basic algorithm.

The procedure proposed by Professor Joseph interpolates the posterior evaluations and strives for high-accuracy approximation. This does not come for free: the approximation is not necessarily a mixture of normals any more and sampling from it is no longer trivial. This makes it more difficult to use it for Monte Carlo sampling. Another cost is that a large number of components is needed in the linear combination (as many as there are evaluation points). This becomes computationally expensive as matrices of large size need to be inverted to calculate the approximation. *iterLap* typically uses less than 20 components.

Another difference is that the interpolation technique uses the same matrices $\boldsymbol{\Sigma}$ and $\boldsymbol{\Lambda}$ for all components, while for *iterLap*, each component has its own matrix $\boldsymbol{\Sigma}$. This allows a better adaption to the local behavior of the posterior density, and by this, a smaller number of components is needed in the approximation.

A difficult task for any computational algorithm for Bayesian problems is to identify the regions of high probability. Determining the initial space-filling design by "prior information" is often not possible in practical situations, when the dimension is beyond 1 or 2. The proposal for finding the initial design for the interpolation technique is based on optimization and the Laplace approximation (note that these function evaluations have not been counted in the examples in sections 4.1 and 4.2 of Professor Joseph's article), so here both methods use the same approach. For sequential exploration, *iterLap* adds components where the approximation error is worst by optimizing the residual error, while the interpolation technique adds design points where the conditional prediction variance is largest, which is identified by starting an optimizer at the point with maximum cross-validation error. The advantage of the latter approach is that no target function evaluations are needed to find the next evaluation point.

Hence, the number of target density evaluations will usually be larger for the *iterLap*, due to the repeated optimizations, grid evaluations, and determination of $\boldsymbol{\Sigma}$, which is currently being done by calculating the Hessian matrix using finite differencing. This is a disadvantage if the target density is computationally expensive.

2.1 Comparisons

I thank Professor Joseph for sending me the code he used for the examples in his article, which allowed me to evaluate the computational efficiency of the procedure on an example. All computations were performed on a computer with 2.30 GHz and 4 GB RAM.

First, however, I applied *iterLap* to the banana example of section 3.2 of his article. With the default parameter settings from the R package, the procedure uses 11 mixture components and obtains a good approximation: the effective sample size (ESS) for importance sampling using 10,000 samples from this proposal is ≈ 7200. Building this approximation takes less than 1 sec on my computer, which is on my computer roughly 300 times faster than the sequential procedure proposed in his article. The number of function evaluations is roughly 2000. By fine-tuning *iterLap* to this problem, one can halve this number without suffering in terms of the quality of the approximation.

Then, I applied the interpolation technique to an 11-dimensional nonlinear model, where the posterior distribution contains some nonlinear features; see (Bornkamp 2011a, sec. 3.2) for details. Here, *iterLap* selects 12 mixture components using the default settings and building the approximations takes roughly 6 sec using roughly 25,000 functions evaluations. Monte Carlo sampling based on this proposal is very competitive with adaptive Markov chain Monte Carlo (MCMC) despite using less function evaluations in total.

Applying the interpolation technique with 550 function evaluations (adapting codes used for the banana example) turned out to be computationally intensive, finding the $\boldsymbol{\Sigma}$ and $\boldsymbol{\Lambda}$

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

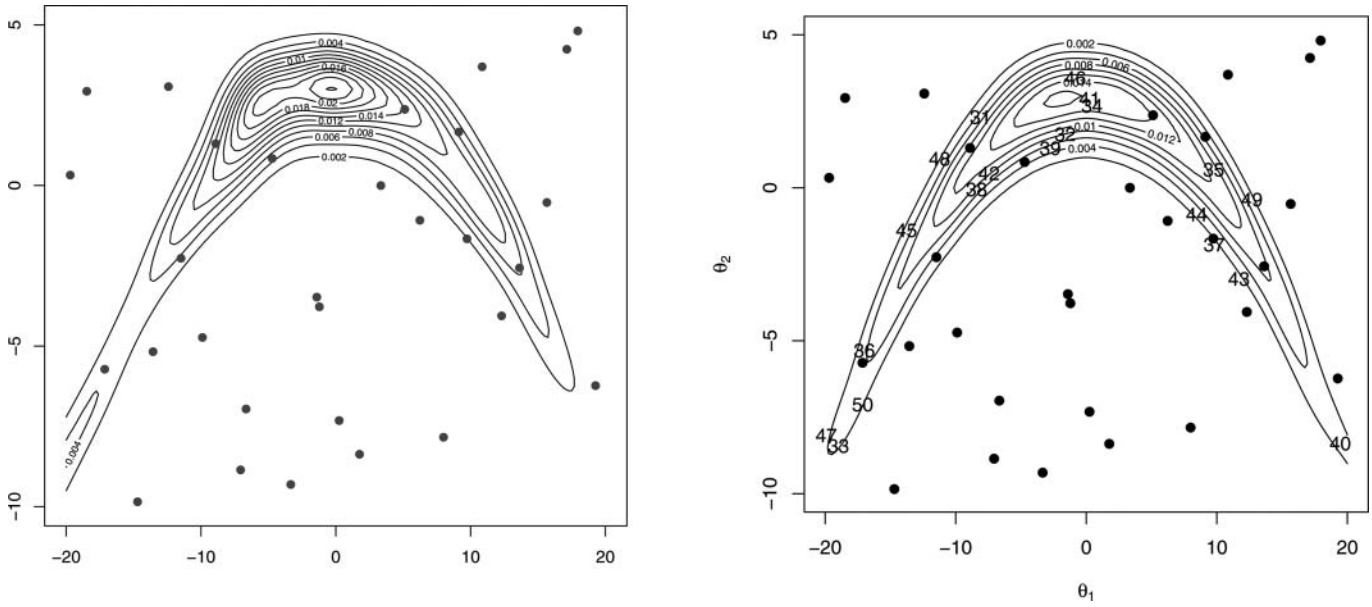

Figure 1. Contours approximated from the initial 30-run design (left panel) and after adding 20 points sequentially (right panel). The online version of this figure is in color.

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

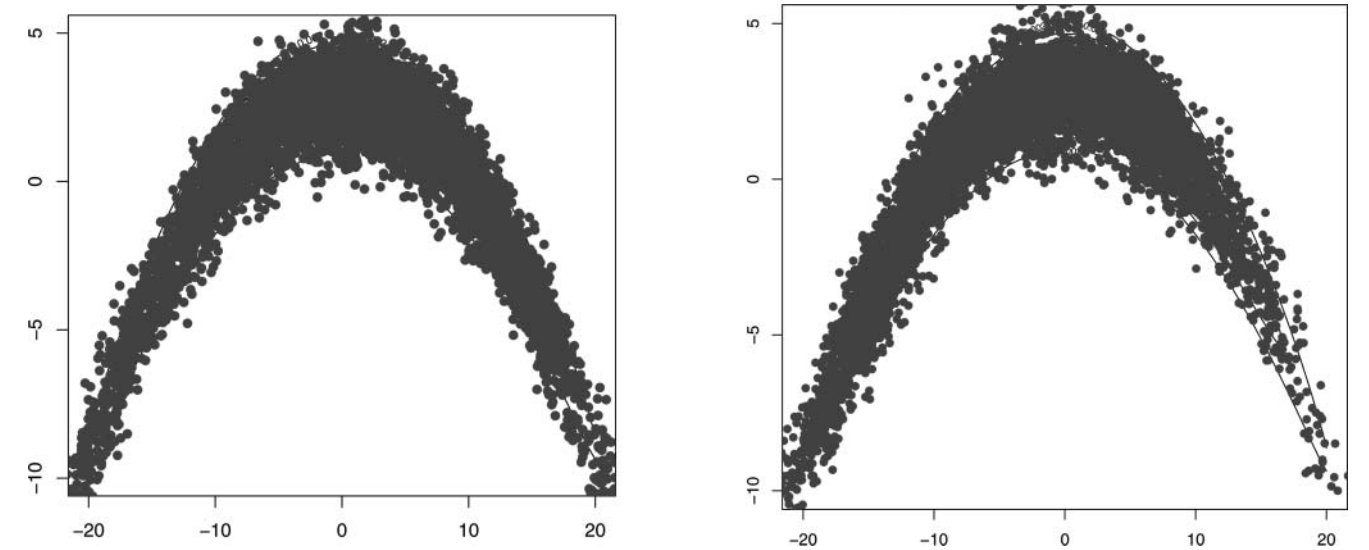

Figure 2. A total of 10,000 points generated using 50 (= 30 + 20) design points (left panel) and 100 (= 100 + 0) design points (right panel). The online version of this figure is in color.

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

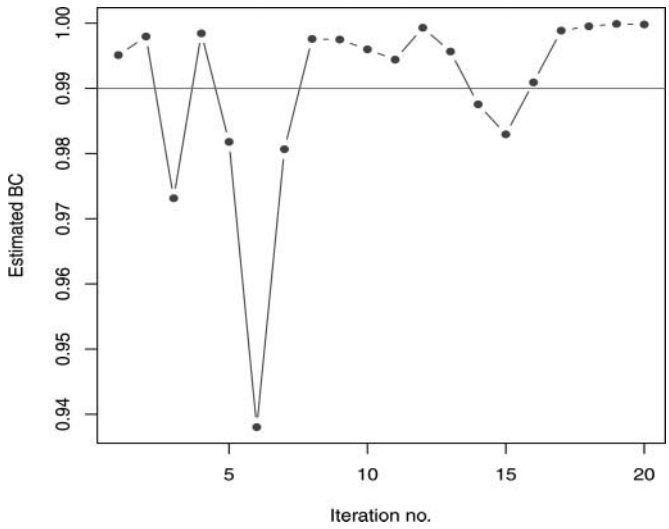

Figure 3. Plot of the Bhattacharya coefficient. The online version of this figure is in color.

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

Computer modeling entails statistical inference through the use of a computer simulator. Typically, the statistician works in collaboration with subject area scientists who have developed a detailed simulation of a process, such asclimate modeling, computer chip design, or social networking. Thus, evaluating the likelihood requires a run of the computer simulator, which could take hours, days, or even months, depending on the simulator. With expensive simulators, the number of likelihood evaluations will necessarily be limited, and thus, posterior estimation becomes difficult. MCMC can be impractical. A common approach is surrogate modeling with a Gaussian process interpolator (Sacks et al. 1989; Santner et al. 2003). Alternatives include importance sampling (Taddy, Lee, and Sansó 2009). The method proposed by Professor Joseph is an intriguing new approach that may allow for more accurate approximations with limited function evaluations. It has great potential to improve Bayesian inference in computer modeling.

2. BAYES

I find it a bit ironic that in approximating a Bayesian posterior density, we end up using a single point estimate. DoIt uses a single value for the coefficients c_i, and a single estimate of the approximating variance $\mathbf{\Sigma}$. Particularly, when the mode is unknown, we know there will be uncertainty present. In a fully Bayesian analysis, should we take that into account? It would be more computational effort, but we could think about a Bayesian approach to estimating $\mathbf{\Sigma}$, instead of using a cross-validation approach. It could be useful to explore how much uncertainty there is, and whether or not it is worth the extra effort to take it into account.

3. UNKNOWN MODES AND SEQUENTIAL DESIGN

When the posterior mode is unknown, or indeed, the problem is multimodal and all modes are unknown, it may be useful to spend a little computing power to actually explore for the modes. The optimization literature contains quite a number of efficient algorithms for derivative-free black-box optimization, methods that could be applied directly here. An example is pattern search (Hough, Kolda, and Torczon 2001), which is an efficient local optimization algorithm. Thus, some likelihood function evaluations could be given to the optimization routine, allowing better information about the mode(s). Pattern search is particularly useful here, because it is easily combined with the general functional exploration in section 3.2 of Joseph's article. One could interweave the evaluations in a style analogous to the hybrid optimization methods in Taddy et al. (2009). This would allow improvement in the approximation by simultaneously reducing uncertainty about the locations of the modes and increasing information about the rest of the function. Pattern search could quickly find the modes, with those function evaluations still being useful for improving the overall fit, while other function evaluations would be directed toward less-explored regions of the input space to address overall fit. Mode hunting will also naturally concentrate more observations in the regions of the modes, which is helpful for the approximation.

4. INTERPOLATION

One common assumption in the computer modeling literature is that when the simulator is deterministic, that is, every time it is run with the same inputs, it will always produce the same outputs, one should always interpolate the results. Joseph's article does not insist on strict interpolation, and I think that is a good thing.

The article acknowledges that basic DoIt can lead to negative estimates of the density, and that one way to ameliorate these results is to shrink the resulting density using a guide density such as a normal density. Since all of the data points will necessarily be nonnegative, a good approximating function would typically also be nonnegative. It is the insistence on interpolation that opens a wide door for negative coefficients c_i. To exactly fit a set of points with a particular smoothness, interpolators can occasionally stray very far from the observed points. By allowing just a little bit of flexibility in the fit, this effect can usually be made to go away. Allowing just a little shrinkage on the data can greatly improve the chances that the fit will be everywhere nonnegative. In practice, this is equivalent to assuming a small amount of error in the observations, or doing a small amount of smoothing in the fit. In the parlance of kriging, this is equivalent to using a small nugget term. In a conceptual sense, instead of the shrinkage to a distribution as proposed in the article, one could do shrinkage directly on the data, which results in the well-understood model of a Gaussian process with measurement error. In any case, the improvement to DoIt method of the article does involve shrinkage, and thus, is no longer a strict interpolator.

In the broader perspective, the common insistence on strict interpolation may be unjustified. When data are plentiful and the function is relatively simple, then interpolation is straightforward. When the assumptions may be questionable, or the data are sparse, is interpolation always best? Gramacy and Lee (2012) have given several examples where allowing just a little shrinkage or smoothing can outperform interpolation on deterministic functions. In practice, interpolation can lead to poor results when something unexpected happens. With nonlinear functions in high dimensions, how often are you sure that there will be nothing unexpected?

In conclusion, I would like to thank Professor Joseph for his innovative article, which may help improve Bayesian inference for complex problems.

ADDITIONAL REFERENCES

Gramacy, R. B., and Lee, H. K. H. (2012), "Cases for the Nugget in Modeling Computer Experiments," *Statistics and Computing*, 22 (3), 713–722. [232]

Hough, P. D., Kolda, T. G., and Torczon, V. (2001), "Asynchronous Parallel Pattern Search for Nonlinear Optimization," *SIAM Journal on Scientific Computing*, 23, 134–156. [232]

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989), "Design and Analysis of Computer Experiments," *Statistical Science*, 4, 409–435. [231]

Taddy, M., Lee, H. K. H., Gray, G. A., and Griffin, J. D. (2009), "Bayesian Guided Pattern Search for Robust Local Optimization," *Technometrics*, 51, 389–401. [232]

Taddy, M., Lee, H. K. H., and Sansó, B. (2009), "Fast Inference for Statistical Inverse Problems," *Inverse Problems*, 25, 085001. [231]

Comment

John T. Ormerod

School of Mathematics and Statistics
University of Sydney
Sydney 2006, Australia
(john.ormerod@sydney.edu.au)

M. P. Wand

School of Mathematical Sciences
University of Technology
Sydney, Broadway 2007, Australia
(Matt.Wand@uts.edu.au)

The author is to be commended on the development of this new piece of methodology, which he names *DoIt*. We believe that the method has the potential to be an important element in the kit-bag of methods for approximate Bayesian inference. Throughout the article, a number of criticisms have been leveled toward variational approximations, of which variational Bayes (VB) is a special case. As much of our recent research has been in this area, we will focus our comments in defense of this methodology.

As a basis for comparison between methods, we adapt the criteria listed in Ruppert, Wand, and Carroll (2003, sec. 3.16), upon which scatterplot smoothers may be judged, to criteria for general statistical methodology:

- *Convenience*: Is it available in a computer package?
- *Implementability*: If not immediately available, how easy is it to implement in the analyst's favorite programming language?
- *Flexibility*: Is the method able to handle a wide range of models?
- *Tractability*: Is it easy to analyze the mathematical properties of the technique?
- *Accuracy*: Does the method solve the problem to sufficient accuracy?
- *Speed*: Are answers obtained sufficiently quickly for the analyst's application?
- *Extendibility*: Is the method easily extended to more complicated settings?

Concerning the convenience criterion, we note that VB is part of the Infer.NET computing framework (Minka et al. 2010). The Infer.NET framework can be used in any of the .NET languages, which includes C#, C++, and Visual Basic, and implements the expectation propagation and Gibb's sampling algorithms in addition to VB. The use of Infer.NET for some simple statistical models is illustrated in Wang and Wand (2011). Although DoIt is a new idea, we look forward to its implementation in a commonly used statistical environment such as R.

The Infer.NET framework is still in its infancy and does not support all models for which VB algorithms can be derived. In such cases, the analyst has to implement VB in his/her favorite programming language.

Under this implementability criteria, VB can also have an advantage over DoIt. The article describes DoIt over several pages. But the algorithm can summarized in the following set of steps, with some notational changes that we believe improve digestibility. Joseph uses $\text{diag}(\boldsymbol{v})$ to denote the diagonal matrix with diagonal entries corresponding to the vector $\boldsymbol{v}$ and $\text{diag}(\boldsymbol{M})$ to denote the diagonal matrix formed when the off-diagonal entries of the square matrix $\boldsymbol{M}$ are set to zero. Following Magnus and Neudecker (1988), we use $\text{dg}(\boldsymbol{M})$ for the latter to avoid having different meanings of "diag." We also use $\boldsymbol{v} > \boldsymbol{0}$ to denote all entries of a vector $\boldsymbol{v}$ being positive:

1. Choose a design $D = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m\}$ within the parameter space (discussed below) and set
$$\boldsymbol{h} = \begin{bmatrix} p(\boldsymbol{y}, \boldsymbol{\theta}_1) \\ \vdots \\ p(\boldsymbol{y}, \boldsymbol{\theta}_m) \end{bmatrix}.$$
2. Define the $m \times m$ matrix $\boldsymbol{G}(\boldsymbol{\sigma})$ to have (i, j)th entry
$$|\text{diag}(\boldsymbol{\sigma})|^{-1} \exp\left[-\frac{1}{2}(\boldsymbol{\theta}_i - \boldsymbol{\theta}_j)^T \{\text{diag}(\boldsymbol{\sigma})\}^{-2}(\boldsymbol{\theta}_i - \boldsymbol{\theta}_j)\right].$$
Solve
$$\widehat{\boldsymbol{\sigma}} = \underset{\sigma > \boldsymbol{0}}{\text{argmin}} \left[\boldsymbol{h}^T \boldsymbol{G}(\boldsymbol{\sigma})^{-1} \{\text{dg}(\boldsymbol{G}(\boldsymbol{\sigma}))\}^{-1} \boldsymbol{G}(\boldsymbol{\sigma})^{-1} \boldsymbol{h}\right].$$
3. Solve
$$\widehat{\boldsymbol{c}} = \underset{c \geq \boldsymbol{0}}{\text{argmin}} \left\{\frac{1}{2} \boldsymbol{c}^T \boldsymbol{G}(\widehat{\boldsymbol{\sigma}})\boldsymbol{c} - \boldsymbol{h}^T \boldsymbol{c}\right\}.$$
4. Define
$$z \equiv \{\text{diag}(\boldsymbol{G}(\widehat{\boldsymbol{\sigma}})\widehat{\boldsymbol{c}})\}^{-1} \boldsymbol{h} \quad \text{and}$$
$$a(\boldsymbol{\lambda}) \equiv \frac{\widehat{\boldsymbol{c}}^T \boldsymbol{G}\left(\sqrt{\widehat{\boldsymbol{\sigma}}^2 + \boldsymbol{\lambda}^2}\right) \boldsymbol{G}(\boldsymbol{\lambda})^{-1} z}{\widehat{\boldsymbol{c}}^T \boldsymbol{G}\left(\sqrt{\widehat{\boldsymbol{\sigma}}^2 + \boldsymbol{\lambda}^2}\right) \boldsymbol{G}(\boldsymbol{\lambda})^{-1} \boldsymbol{1}}$$
for $m \times 1$ vectors $\boldsymbol{\lambda}$. Here, $\sqrt{\widehat{\boldsymbol{\sigma}}^2 + \boldsymbol{\lambda}^2}$ is the $m \times 1$ vector defined by taking element-wise squares and square roots, and $\boldsymbol{1}$ is an $m \times 1$ vector of 1's. Solve
$$\widehat{\boldsymbol{\lambda}} = \underset{\boldsymbol{\lambda} > \boldsymbol{0}}{\text{argmin}} [\{z - a(\boldsymbol{\lambda})\boldsymbol{1}\}^T \boldsymbol{G}(\boldsymbol{\lambda} \odot \widehat{\boldsymbol{\sigma}})^{-1} \{\text{dg}(\boldsymbol{G}(\boldsymbol{\lambda} \odot \widehat{\boldsymbol{\sigma}}))\}^{-1} \times \boldsymbol{G}(\boldsymbol{\lambda} \odot \widehat{\boldsymbol{\sigma}})^{-1} \{z - a(\boldsymbol{\lambda})\boldsymbol{1}\}],$$

TECHNOMETRICS, AUGUST 2012, VOL. 54, NO. 3
DOI: 10.1080/00401706.2012.697244

where $\widehat{\boldsymbol{\lambda}} \odot \boldsymbol{\sigma}$ denotes the element-wise product of $\widehat{\boldsymbol{\lambda}}$ and $\boldsymbol{\sigma}$.

5. The approximation to the posterior density function $p(\boldsymbol{\theta}|\underline{y})$ involves simple calculations involving D, $\widehat{\sigma}$, $\widehat{c}$, and $\widehat{\boldsymbol{\lambda}}$, given by (13) and (14) in the article.

The DoIt algorithm may need to follow steps 1–4 many times to determine a good design set D, which is chosen differently depending on whether the posterior mode is known. If the posterior mode is known, then D is chosen to follow a Latin hypercube design based on the Laplace approximation of the posterior density. If the posterior mode is unknown, or if the Laplace approximation is judged to be inaccurate, then D is built sequentially by solving an additional suite of multidimensional optimization problems. The starting points for these maximization problems are obtained by choosing a point in the neighborhood of the $\boldsymbol{\theta}_i$ with the largest approximate leave-one-out error (specific details for this step are vague on how this neighborhood is chosen). The DoIt algorithm stops adding points to D when an approximate cross-validation criterion-based criterion is judged to be sufficiently accurate. The minimization problems are solved using the Nelder–Mead algorithm, which does not require derivative information. The algorithm contains many subproblems. Each of these subproblems may require some tuning for DoIt to obtain reasonable results. Termination criteria may need to be adjusted, multiple starting points may be required to ensure Steps 2 and 4 do not obtain poor results, and the size of the neighborhood used for sequential updates of the design may need adjusting. Consider the longitudinal data analysis example considered in section 4.1 of the article. The VB algorithm for this analysis, corresponding to algorithm 3 of Ormerod and Wand (2010), requires 10–15 lines of simple R code to implement and no tuning. In comparison, DoIt requires several multidimensional constrained optimizations and, possibly, some tuning.

The DoIt algorithm has been custom-designed for models involving continuous random variables with continuous joint distributions (implied by Theorem 1). Provided that the problem falls into this category, DoIt appears quite flexible. In particular, results for the nonlinear regression in section 3.1 are quite impressive and we do not know of a variational approximation for obtaining suitably accurate approximations for problems of this type. Furthermore, the only other non-MCMC (Markov chain Monte Carlo) method that we are aware of, suitable for this type of problem, is the iterLap method of Bornkamp (2011a). However, VB is applicable in situations for models with both discrete and continuous random variables, and it is not limited to joint distributions that are continuous. For example, the VB method has been successfully applied to Gaussian mixture models (McGrory and Titterington 2007) and hidden Markov models (McGrory and Titterington 2009), and has an advantage over DoIt in this setting. Furthermore, when the prior is discontinuous, for example, if the horseshoe prior of Carvalho, Polson, and Scott (2010) is employed, then VB can be applied (Neville, Ormerod, and Wand 2012). In such a setting, it is unclear whether DoIt needs a prohibitively large number of design points to obtain a sufficiently accurate approximation. In short, for the criteria of flexibility, VB can handle some models DoIt cannot and vice versa.

Both methods are simple and fairly easy to understand how answers are obtained. We admit that few theoretical developments for variational approximations have been made and those that exist are context-specific (Hall, Humphreys and Titterington 2002; Wang and Titterington 2006; Hall, Ormerod and Wand 2011; Hall et al. 2011; Ormerod and Wand 2012). In terms of tractability, Gaussian interpolation is a reasonably well-understood technique (e.g., Fasshauer 2007). As noted in the article, most results for bounding errors for such interpolation methods rely on the fill-distance of the design points. We do not know of results for obtaining good designs in high-dimensional spaces. Thus, we concur that a direct application of DoIt, without using some type of dimension reduction, would be unsuitable for high-dimensional problems. In comparison, VB has been successfully applied in genetic association studies, where the problems can involve parameters numbering in

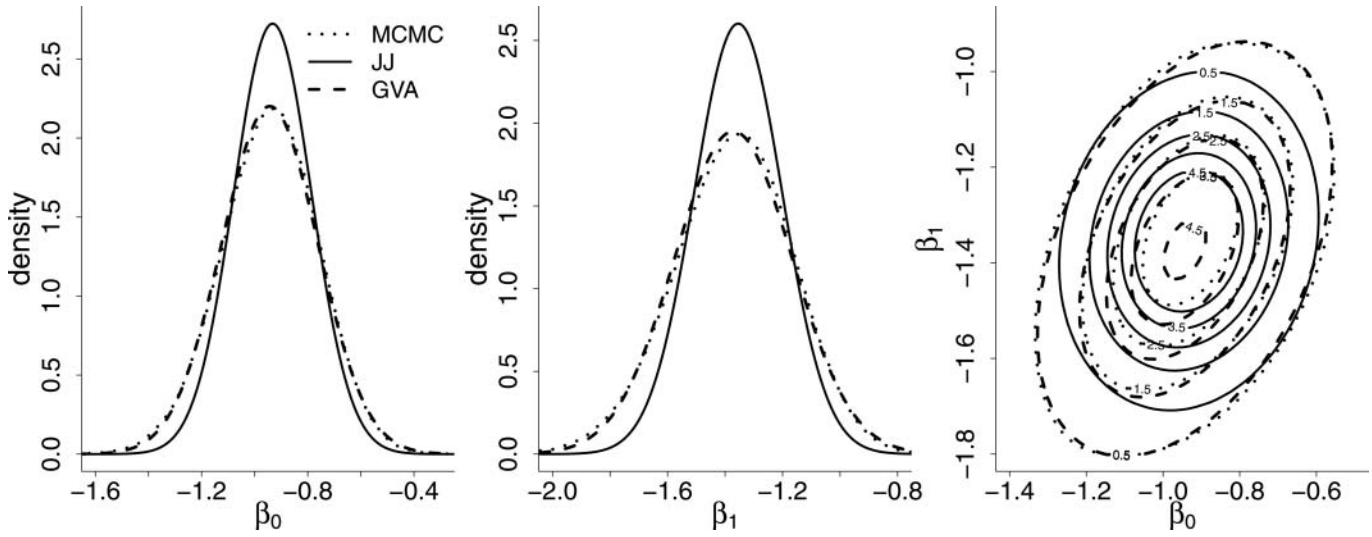

Figure 1. A comparison of tangent-based variational approximations (JJ), Gaussian variational approximations (GVA), and MCMC for the bronchopulmonary dysplasia example in Wand (2009).

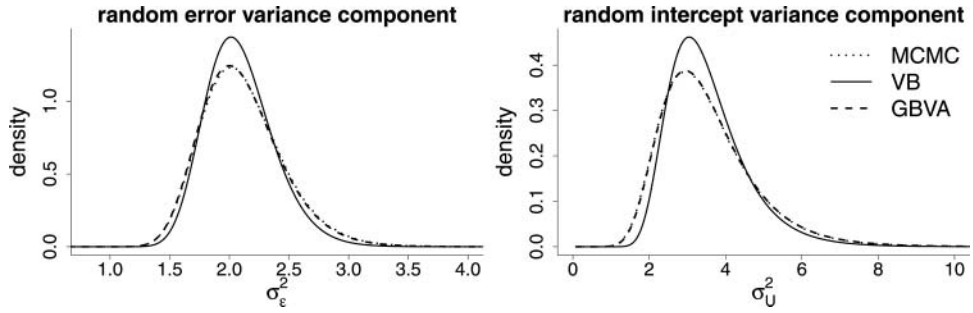

Figure 2. Posterior density estimates for the inverse variance components using VB and the grid-based variational approximation described in Ormerod (2011).

hundreds of thousands (Logsdon, Hoffman, and Mezey 2010; Carbonetto and Stephens 2011).

Criteria accuracy and speed could be considered together as one is often traded against the other. Furthermore, these should be considered in the context of the application at hand. Consider again the longitudinal data analysis example considered in section 4.1. Joseph describes the VB approximations for the variance components as "poor." We would call them "reasonable." Furthermore, these approximations, using a näive implementation in R (which does not take advantage of the random-effects structure), takes around 0.01 sec to compute on the first author's laptop. If, in the context of the analysis, the analyst was only interested in the posterior approximations of the coefficients, then VB would be the ideal choice for this problem. It is hard to compare DoIt with this in mind as the article does not report how long DoIt takes to solve this problem, but we anticipate that VB would compare favorably.

Our second objection to the comparison with variational approximations with DoIt is that all variational approximations are lumped together. For example, in section 2.5 of the article, DoIt is compared with the tangent-based variational approximation of Jaakkola and Jordan (2000), which we denote by JJ. For this problem, JJ can be markedly inferior to Gaussian variational approximation (GVA) (Ormerod and Wand 2012), as we now demonstrate. Consider the example presented in Wand (2009, sec. 6) in Figure 1 where JJ and GVA are applied. Clearly, GVA, like DoIt, appears adequately accurate for this problem, whereas JJ does not. Similarly, again considering the longitudinal data analysis example considered in section 4.1, the article compares the VB method described in Ormerod and Wand (2010) when other variational approximations are superior in terms of accuracy. Consider, in Figure 2, the grid-based variational approximation of Ormerod (2011). This approximation, like the structured mean field variational approximation described in Wand et al. (2011), offers a general method for improving variational approximations, albeit at the expense of speed. Using grid-based variational approximations, adequate approximations for the marginal posterior densities of the variance components can be obtained. In this regard, the article appears to be making a straw-man argument against variational approximations.

An attraction of VB is that relative ease with which it can be extended to handle complications such as missing data. This follows from the locality property of VB, which, as with MCMC, means that algorithmic components are localized on the directed acyclic graph of the Bayesian model (e.g., Wand et al. 2011, sec. 3). In Faes, Ormerod, and Wand (2011), we demonstrated the extendibility of VB to handling missingness in regression models. Since missingness leads to an increase in the size of the Bayesian model (an increase in the number of *hidden nodes* in graph theoretical language), we would expect DoIt to run into difficulties for such models.

In summary, we believe that, while DoIt is a worthy addition to non-MCMC analysis and that the results presented in the article are impressive, variational approximations still offer a competitive alternative for many problems, depending on the analyst's weighting of the aforementioned criteria.

ADDITIONAL REFERENCES

Bornkamp, B. (2011a), "Approximating Probability Densities by Iterated Laplace Approximations," *Journal of Computational and Graphical Statistics*, 20, 656–669. [234]

Carbonetto, P., and Stephens, M. (2011), "Scalable Variational Inference for Bayesian Variable Selection in Regression, and Its Accuracy in Genetic Association Studies," *Bayesian Analysis*, 6, 1–42. [234]

Carvalho, C. M., Polson, N. G., and Scott, J. G. (2010), "The Horseshoe Estimator for Sparse Signals," *Biometrika*, 97, 465–480. [234]

Faes, C., Ormerod, J. T., and Wand, M. P. (2011), "Variational Bayesian Inference for Parametric and Nonparametric Regression With Missing Data," *Journal of the American Statistical Association*, 106, 959–971. [235]

Fasshauer, G. E. (2007), *Meshfree Approximation Methods With Matlab*, (Vol. 6: Interdisciplinary Mathematical Sciences). Singapore: World Scientific. [234]

Hall, P., Humphreys, K., and Titterington, D. M. (2002), "On the Adequacy of Variational Lower Bound Functions for Likelihood-Based Inference in Markovian Models With Missing Values," *Journal of the Royal Statistical Society,* Series B, 64, 549–564. [234]

Hall, P., Ormerod, J. T., and Wand, M. P. (2011), "Theory of Gaussian Variational Approximation for a Poisson Mixed Model," *Statistica Sinica*, 21, 369–389. [234]

Hall, P., Pham, T., Wand, M. P., and Wang, S. S. J. (2011), "Asymptotic Normality and Valid Inference for Gaussian Variational Approximation," *The Annals of Statistics*, 39, 2502–2532. [234]

Jaakkola, T. S., and Jordan, M. I. (2000), "Bayesian Parameter Estimation via Variational Methods," *Statistics and Computing*, 10, 25–37. [235]

Logsdon, B. A., Hoffman, G. E., and Mezey, J. G. (2010), "A Variational Bayes Algorithm for Fast and Accurate Multiple Locus Genome-Wide Association Analysis," *BMC Bioinformatics*, 11, 1–13. [234]

Magnus, J. R., and Neudecker, H. (1988), *Matrix Differential Calculus With Applications in Statistics and Econometrics*, Chichester: Wiley. [233]

McGrory, C. A., and Titterington, D. M. (2007), "Variational Approximations in Bayesian Model Selection for Finite Mixture Distributions," *Computational Statistics and Data Analysis*, 51, 5352–5367. [234]

——— (2009), "Variational Bayesian Analysis for Hidden Markov Models," *Australian and New Zealand Journal of Statistics*, 51, 227–244. [234]
Minka, T., Winn, J., Guiver, J., and Knowles, D. (2010), "Infer.Net 2.4," Available at *http://research.microsoft.com/infernet*. [233]
Neville, S. E., Ormerod, J. T., and Wand, M. P. (2012), "Mean Field Variational Bayes for Continuous Sparse Signal Shrinkage: Pitfalls and Remedies," unpublished manuscript. Available at *http://www.uow.edu.au/~mwand/papers.html* [234,235]
Ormerod, J. T. (2011), "Grid Based Variational Approximations'," *Computational Statistics and Data Analysis*, 55, 45–56. [234,235]
Ormerod, J. T., and Wand, M. P. (2010), "Explaining Variational Approximations," *The American Statistician*, 64, 140–153. [234,235]
Ruppert, D., Wand, M. P., and Carroll, R. J. (2003), *Semiparametric Regression*, Cambridge: Cambridge University Press. [233]
Wand, M. P. (2009), "Semiparametric Regression and Graphical Models," *Australian and New Zealand Journal of Statistics*, 51, 9–41. [233]
Wand, M. P., Ormerod, J. T., Padoan, S. A., and Frühwirth, R. (2011), "Mean Field Variational Bayes for Elaborate Distributions," *Bayesian Analysis*, 6 (4), 847–900. [235]
Wang, B., and Titterington, D. M. (2006), "Convergence Properties of a General Algorithm for Calculating Variational Bayesian Estimates for a Normal Mixture Model," *Bayesian Analysis*, 1, 625–650. [234]
Wang, S. S. J., and Wand, M. P. (2011), "Using Infer.NET for Statistical Analyses," *The American Statistician*, 65, 115–126. [233]

Comment: DoIt—Some Thoughts on How to Do It

David M. Steinberg

Department of Statistics and Operations Research, The Raymond and Beverly Sackler Faculty of Exact Sciences
Tel Aviv University
Tel Aviv 69978, Israel
(*dms@post.tau.ac.il*)

Bradley Jones

SAS Institute, SAS Campus Drive
Cary, NC 27513
(*Bradley.Jones@jmp.com*)

Computational methods to explore posterior distributions, in particular Markov chain Monte Carlo (MCMC), have played a dominant role in Bayesian statistics over the last 30 years. These methods have enabled statisticians and researchers to tackle problems that defy closed-form solution, greatly expanding the scope of Bayesian analysis.

Joseph's ingenious DoIt algorithm uses ideas developed over the last 20–25 years on statistical modeling of deterministic functions to develop a direct approximation to complex posterior distributions, without the need for the large sequential samples required by MCMC. The method can be applied to a wide variety of problems and offers the promise of accurate results with substantially reduced computing. The approximation is a weighted sum of Gaussians, which leads to the significant advantages that it is simple to normalize and it is easier to compute marginal densities. We think that the method has great potential and applaud Dr. Joseph for this important new idea. Our comments focus on some issues where we think further work might lead to additional improvements in the method.

1. THE DOIT POSTERIOR DENSITY APPROXIMATION

The DoIt approximation is a linear combination of basis functions of the form $g(\boldsymbol{\theta};\boldsymbol{\nu}_i,\boldsymbol{\Sigma}) = \exp\{-0.5(\boldsymbol{\theta}-\boldsymbol{\nu}_i)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta}-\boldsymbol{\nu}_i)\}$, where $\boldsymbol{\nu}_i$ is an evaluation point and $\boldsymbol{\Sigma}$ plays the role of a covariance matrix in a multivariate Gaussian density. The matrix $\boldsymbol{\Sigma}$ is clearly important in determining the quality of the DoIt approximation. But there are three important issues to consider: (1) What happens if the variances are too large? (2) What happens if the variances are too small? (3) What happens if the orientation is chosen poorly?

To see what can happen when the variances are too large, we consider the nonelliptical posterior density from Haario et al. that is studied in section 3.2. There is a single posterior mode at (0,3). The second derivatives of $\log[h(\boldsymbol{\theta})]$ at the mode lead to a diagonal covariance matrix whose entries are 100 and 1, respectively. Even for evaluation points very close to the mode, the associated Gaussian basis functions assign nonnegligible density to $\boldsymbol{\theta}$ values that have negligible posterior density (e.g., $\boldsymbol{\theta} = (15,3)$, with ν at the mode). With evaluation sites in the same region, the kriging predictor will "correct" for this error, but to do so, it must assign some basis functions positive coefficients and others negative coefficients. So, the problem of potentially negative density values is compounded. In this case, we think that it would be beneficial to "shrink" the variances (relative to the second derivatives) for the purpose of fitting the DoIt approximation. The fraction of negative coefficients in the initial DoIt fit could be a useful diagnostic here—a large fraction of negative coefficients may suggest that the variances are too large.

Large variances can also cause computational problems. The initial kriging estimator involves solving the linear system $\mathbf{Gc} = \mathbf{h}$, in which $\mathbf{G}$ is a correlation matrix. With large variances, $\mathbf{G}$ may be an ill-conditioned matrix for which the solution is numerically unstable. For the Haario et al. example with our cross-validation estimates of the variances, 20 of the 100 singular values of $\mathbf{G}$ were effectively 0.

On the other hand, the problem of having variances that are too small is that the Gaussians centered near the mode will fail to

TECHNOMETRICS, AUGUST 2012, VOL. 54, NO. 3
DOI: 10.1080/00401706.2012.697247

assign positive density to a sufficiently wide range of θ space. DoIt addresses this problem directly by including Gaussians centered away from the mode. Working with variances that are too small will thus result in more evaluation sites than are really necessary *if* the true posterior is approximately multivariate Gaussian. However, if the posterior is not multivariate Gaussian, then it seems to be much safer to use smaller variances and to compensate by adding evaluation sites. We wonder, in fact, if it would not be good general practice to always use variances smaller than those suggested by the second derivatives.

The orientation of $\boldsymbol{\Sigma}$ relates to the correlation among the parameters. Imagine having two parameters whose posterior is very close to multivariate normal with a correlation above 0.9. A single normal density, centered at the mode and with the correct covariance, will provide a good approximation to the posterior. An approximation using only normal densities for two independent random variables, though, will require a large number of evaluation sites to approximate the posterior. In the absence of second-derivative information, DoIt proceeds by working with a diagonal $\boldsymbol{\Sigma}$, that is, by summing densities of independent Gaussians. This approach will require many evaluation sites to compensate for the lack of orientation. Moreover, the orientation issue can team up with that of determining the variances. If we need to sum many independent Gaussians to approximate a correlated posterior, DoIt should be more successful with "local" Gaussians, that is, with variances that are distinctly smaller than the marginal posterior variances. We are convinced that the diagonal elements of $\boldsymbol{\Sigma}$ should be close to the marginal variances only if there is accurate information on orientation.

The above arguments suggest that substantial improvements to DoIt might be possible via better choice of $\boldsymbol{\Sigma}$. One option is to compute a first approximation to the posterior from an initial sample of points, use that first approximation to estimate the posterior covariance matrix, and perform a further approximation using that estimate as $\boldsymbol{\Sigma}$ in place of the original diagonal matrix. One might take this a step further and attempt to derive local versions of $\boldsymbol{\Sigma}$ for each evaluation site. For example, with the posterior density of Haario et al., the orientation of $\boldsymbol{\Sigma}$ should no doubt depend on the evaluation site, with independence near the mode (as indicated by the second-derivative information at the mode), positive correlation in the "lower-left" branch and negative correlation in the "lower-right" branch of the density. Perhaps, the initial DoIt density estimator could be used to compute such local orientation information.

The current DoIt method seems best suited to posteriors in which the parameters are nearly independent. Thus, it might be worthwhile to think carefully about the parameterization so that near independence is achieved. Of course, in many problems that will be difficult to accomplish.

Should all the evaluation sites be retained for computing the approximation? As proposed, DoIt includes all the evaluation sites in the approximation to the density. Can a good approximation be computed using just a subset of these sites? The entries in the matrix $\mathbf{G}$ can be viewed as correlations, in which sites that are "close together" (in the metric defined by $\boldsymbol{\Sigma}$) have high correlations. It is well known that the presence of sites that are close to one another can cause $\mathbf{G}$ to be seriously ill-conditioned. We already noted that we encountered such a problem for the Haario et al. example with our sample of 100 points. So, one advantage of "thinning" the sites is to improve the condition number of $\mathbf{G}$ and avoid potential computational error. Sites that are located in regions of very low posterior probability are likely to have very small coefficients, so eliminating them altogether might not degrade the approximation. Developing clear rules, though, for which sites to drop is a challenging problem.

In section 2.3, Joseph shows how his initial DoIt estimator can be modified to obtain a convex combination of Gaussian densities, which must be nonnegative. He uses quadratic programming to find this estimator. An alternative computational strategy is to use nonnegative least squares; see Lawson and Hanson (1995) for details.

The final DoIt approximation takes the convex combination of Gaussian densities and multiplies it by a linear correction term. The tuning constant a in the correction term is estimated as the weighted average of the ratios $z_i = h(\boldsymbol{\nu}_i)/w(\boldsymbol{\nu}_i)$, where h is the unnormalized, computed posterior density and w is the approximation to h based on the convex combination of Gaussians with nonnegative coefficients (the DoIt approximation in Equation (9), but for the unnormalized density). If this mixture of Gaussians is a good approximation, all the ratios should be close to 1. Averaging ratios, which are naturally asymmetric, is a risky endeavor; we worry that some unusually high ratios, even when downweighted by the estimated density in (9), could lead to a poor estimate of the tuning constant. At the least, we would recommend checking the values of the ratios before averaging. If an automatic computation is needed, we would suggest averaging the log ratios (perhaps with trimming to remove any wild ratios) and then exponentiating back to the ratio scale to estimate a in the correction term.

Joseph's DoIt method approximates the posterior density directly, whereas Bliznyuk et al. (2008) and Fielding et al. (2011) proposed approximations to the log of the posterior density. Joseph conjectures that the DoIt approximation is more successful, especially for multimodal densities, because they are more likely to be additive when they are not logged. An alternative explanation is more closely tied to the nature of the kriging approximation used by Joseph. The kriging approximation involves a linear combination of basis functions that are derived from the Gaussian process (GP) correlation function. In DoIt, the basis functions are densities and it is not surprising that a linear combination of densities is more successful in approximating a density than in approximating a log density. Further, the kriging predictor tends to 0 as it moves away from the evaluation sites; this is desirable in the density scale, but not in the log density scale, where we need to have arbitrarily small values as θ moves away from the modes. For example, if the posterior is approximately normal, the log density decreases quadratically. To reproduce this behavior in the kriging model, one would need a regression model with second-order effects in all components of θ, adding many additional parameters to the GP. Thus, modeling the log density by kriging and exponentiating back to approximate the density does not look like an attractive option.

2. THE SAMPLING STRATEGY

Joseph makes two constructive suggestions for locating the initial sample of evaluation points: (1) identify posterior modes

and locate points about the modes, ideally in a manner that reflects covariance, estimated from second derivatives at each mode, and (2) take evaluation points that reflect the prior. Neither method guarantees a good initial sample. Consider again the posterior density from Haario et al. that is studied in section 3.2. The initial DoIt sample based on the modal information will be an ellipse of points parallel to the axes, with $-30 < \theta_1 < 30$ and $0 < \theta_2 < 6$. This initial sample will miss the region where the density is large for smaller values of θ_2 and will cover large regions where the density is very close to 0.

In the nanowire experiment (section 3.1), Joseph based the initial evaluation sample of 600 points on the maximum likelihood estimate (MLE). Figure 6 in Joesph's paper shows that the sample has most values of γ_3 in the interval [2.8,3.8] and most values of γ_4 in the interval [2,2.65]; the corresponding intervals for $\theta_j = \exp(\gamma_j)$ are [16.4,44.7] and [7.4,14.2], respectively. The posterior densities plotted in figure 7 show substantial probability outside these intervals for both parameters.

Basing an initial sample on modal information will also run aground if there are multiple modes and not all are identified. We have observed disjoint modes in some nonlinear regression problems. For example, in seismic event location, there can be completely isolated posterior modes if the array of seismic stations that detect the event does not provide good triangulation. (In the extreme case, imagine all the stations are located on a north–south line. It will be impossible to know if signals arriving at the stations have come from the east or the west of the stations and there will be matching modes on either side. Stations are never fully linear, but sometimes approximately so, for example, in locating an offshore seismic event.)

It is also not clear what to do if the parameter has a fixed limit and the MLE is at or near the limit. A common example would be a variance component whose MLE is 0. In the binomial example in section 2, the MLE for the probability is 1 and θ has a monotone increasing likelihood; however, it is not difficult to combine the likelihood with the prior to find the posterior mode, which is finite. In that example, Joseph adopts the alternative parameter θ, which is not bounded to an interval. The same strategy is used for the Poisson parameter in the leading example, where the analysis is done for $\log(\theta)$. We think that such transformation will generally be good practice with DoIt.

Taking an initial sample from the prior is less appealing. With the Bernoulli example, the binomial parameter is limited to [0,1] and the prior assigns mass to a wide range of probabilities, with much of the mass on extreme probabilities near 0 or 1. Suppose, though, that one had a location parameter or a regression coefficient in the model. Many Bayesian analyses assign flat priors to such parameters; for example, see section 4.1 in the article. Sampling from the prior is then not an option. Priors that are approximately flat (e.g., a normal prior with a very large variance) could lead to evaluation points that miss the modes completely unless the number of initial points is quite large, especially in higher dimension.

There is always some potential to miss the regions of high posterior density, even when the posterior mode is known. For example, if the parameters have high posterior correlation, there may be only a narrow region where the posterior density is nonnegligible. The evaluation sample may have almost no overlap with that region unless the mode and the correlation are known.

One of the important features of DoIt is the ability to correct an initial approximation by adding new evaluation sites. Joseph's strategy for selecting sites has three components: (1) choose where the density is large, (2) choose where the kriging predictor suffers from uncertainty, and (3) choose near an existing, but poorly predicted, evaluation site. Point (3) is optional and is recommended if the other criteria may lead to many local optima. Intuitively, all three criteria appear reasonable, but some words of caution are also necessary. Our most important concern is that the region over which criterion (19) in the article is optimized is not explicitly stated. If new evaluation sites are limited to the region from which the initial sample was chosen, many points with high density might be ruled out; see our earlier discussion of the initial sample for the Haario et al. example based on second derivatives. Point (3) suggests searching locally about an existing evaluation site, but it is not clear how that local region should be defined. We agree that some flexibility may be needed here, but at the least, more detailed guidelines are important.

Avoiding evaluation sites where the density is small is appealing. However, this may not meet objectives. If the goal is to have an especially good assessment of the posterior probability of a particular region in Θ space, we will most likely want evaluation sites there, even if the density is small. If we need to compute a tail probability or a credible region, it may be important to add sites where the density is small. Another concern is that we do not know the density, so must rely on our current estimate. If our current estimator has missed some region where there is an additional mode, the sampling scheme assures that we will continue to miss it. We may be interested in transformations of the parameters; values with high density for the parameters of interest may fail to have high density for the parameters used in modeling. Finally, sampling only in the vicinity of existing evaluation sites is reasonable only if we already have a broad sample that extends beyond the region where the density is nonnegligible. Going back to the nanowire experiment, with the knowledge that the posterior density places nonnegligible mass outside the points in the initial sample, it seems essential to add some evaluation sites that are more extreme.

3. SUMMARY

Our comments are intended to generate discussion about possible improvements to and limitations of the DoIt method. Although we have focused on those aspects, we want to reaffirm our belief that DoIt has potential to be a useful tool for Bayesian inference, especially for problems involving few parameters.

ACKNOWLEDGMENT

Our work was funded by a grant from the United States–Israel Binational Science Foundation.

ADDITIONAL REFERENCE

Lawson C. L., and Hanson R. J. (1995), *Solving Least Squares Problems*, Philadelphia, PA: Society for Industrial and Applied Mathematics. [237]

Rejoinder

V. Roshan JOSEPH

H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332
(*roshan@gatech.edu*)

It was a pleasure and honor to read the in-depth and thought-provoking discussions on DoIt. I thank the Editor for organizing the discussions and all the discussants for sharing their valuable insight of the methodology. In this rejoinder, I will focus my attention on some of the main issues raised in the discussions and respond to them in the following four sections.

1. COMPARISONS

The discussants have compared the DoIt approximation to several alternative methods for Bayesian computation, which have unearthed some of the shortcomings of DoIt and paved the way for its improvement. These are discussed below.

1.1 Variational Bayes

Professors Ormerod and Wand revisit the orthodontic example of Section 4.1 and compare VB with DoIt in terms of speed and accuracy. Hundred iterations of the VB algorithm given by Ormerod and Wand (2012) took about 0.06 sec in my desktop, whereas DoIt took about 6 sec. This shows that VB is 100 times faster than DoIt in this particular example. However, this does not include the set-up time for deriving the VB algorithm given by Ormerod and Wand (2012) and also the time for writing a code for implementing it. On the other hand, the DoIt implementation is easy. The four steps given in the discussion by Ormerod and Wand remain the same irrespective of the problem. The user needs to update only the likelihood and prior. Thus, if the setup and coding times are also taken into account, one will find DoIt producing results in much shorter time than VB.

The VB approximation produced about 30% error in the variance estimates of the two variance components, whereas the estimation error with the DoIt approximation is negligibly small. The grid-based variational approximation proposed by Ormerod (2011) offers a promising method for improving the accuracy of the VB approximation. Interestingly, DoIt can also be used for this purpose. Conversely, VB can be used to improve the DoIt approximation whenever the VB implementation is readily available.

Consider again the orthodontic example. The Laplace approximation gives $\hat{\boldsymbol{\beta}} = (23.81, 0.66, 1.16)'$, $\hat{\gamma}_\epsilon = \log \hat{\sigma}_\epsilon^2 = 0.705$, and $\hat{\gamma}_u = \log \hat{\sigma}_u^2 = 1.1096$, whereas the VB estimates of the posterior mode are $\hat{\boldsymbol{\beta}} = (23.81, 0.66, 1.16)'$, $\hat{\gamma}_\epsilon = 0.717$, and $\hat{\gamma}_u = 1.183$. We can see that although the estimates of the fixed effects are unchanged, there is significant change in the estimation of the variance components. The estimates from the VB are better than those from the Laplace approximation and, therefore, a better design for DoIt can be generated using these VB estimates instead of the Laplace's estimates. I sampled $m = 50$ points using the maximin Latin hypercube design (MmLHD), but centering at the VB estimate. The results are plotted in Figure 1. We can see that the VB approximation on the variance components are substantially improved using the DoIt approximation and is reasonably close to the 250-point DoIt approximation produced in Figure 12 of Section 4.1. For comparison, the 50-point DoIt approximation using the original Laplace approximation is also plotted in the same figure, which is not as good as the VB+DoIt approximation.

I completely agree with Ormerod and Wand that VB can handle some models that DoIt cannot and vice-versa. Thus, an integration of these two techniques can potentially mitigate their deficiencies and produce much better results in some applications.

1.2 Iterated Laplace Approximation

Dr Bornkamp has given a very clear exposition of the advantages and disadvantages of DoIt and iterLap. Consider the comparison he made in Section 2.1 using the banana example. The iterLap technique identified 11 mixture components, which fits the posterior beautifully and in much less time than what it took for DoIt using the sequential design. I will show here that with a slight twist in the implementation, DoIt can do better! Instead of the 100-run MmLHD, I generated a 500-run MmLHD and removed points for which h_i is less than 1% of the maximum observed value ($\max_i h_i$). This idea of using a subset of points was suggested by Professor Steinberg and Dr. Jones in their discussion. This approach removed 323 points resulting in a design of 177 points. The DoIt approximation obtained using these 177 points is shown in Figure 2. We can see that it gives an excellent approximation, which is obtained using only a quarter of the number of function evaluations used by iterLap. The fitting took about 2.3 sec in my desktop, slightly more than iterLap. I suspect that the DoIt fitting time can be reduced if my rudimentary R code can be transferred into C++ source code.

TECHNOMETRICS, AUGUST 2012, VOL. 54, NO. 3
DOI: 10.1080/00401706.2012.700165

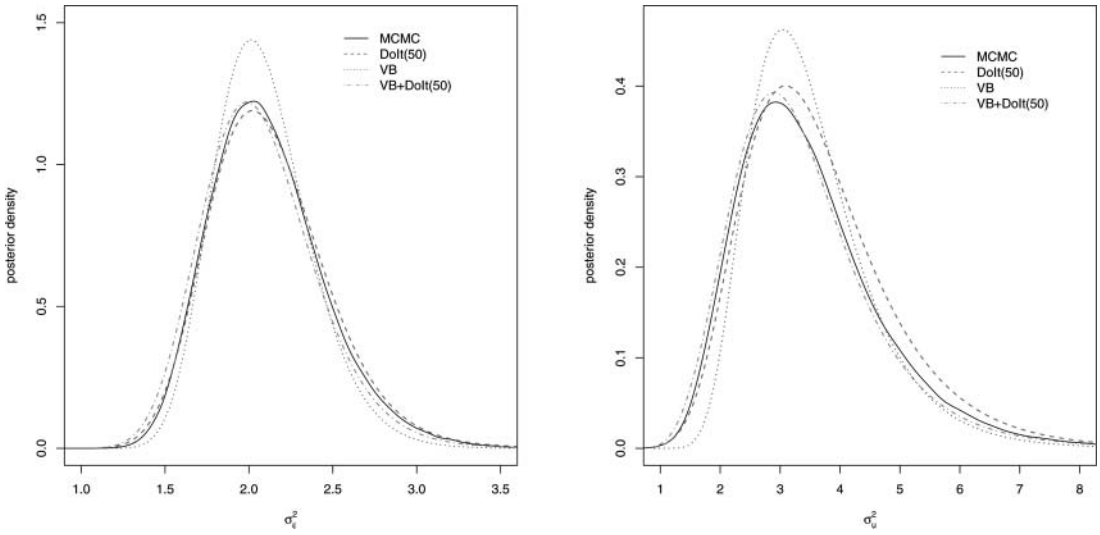

Figure 1. Comparison of VB and DoIt using $m = 50$ points for the variance components in the orthodontic example. The online version of this figure is in color.

However, this approach of generating a large design and removing points from low probability regions loses its appeal in high dimensions where the points can be sparse. Moreover, if the function evaluations are expensive, then removing points from the design can be quite wasteful. Therefore, it is important to develop an efficient algorithm that can quickly generate a space-filling design in high-probability regions. I believe that such an algorithm is possible and will be developed in the near future.

I also applied DoIt to the 11-dimensional nonlinear model mentioned in Section 2.1 of Bornkamp's discussion. Different from Bornkamp's implementation, $\mathbf{\Sigma}$ was estimated through numerical differentiation and $\mathbf{\Lambda}$ based on crossvalidation. I generated a 1100-run MmLHD and removed the points from the low probability regions as done for the banana example, which resulted in a 436-run design. Fitting DoIt on this design took about 1 min, which is much larger than the 6 sec taken by iterLap. On the other hand, iterLap evaluated the function about 25,000 times, whereas DoIt used only 1100 function evaluations. Thus, I agree with Bornkamp's conclusion that DoIt will have an advantage over iterLap only when the function evaluations are expensive.

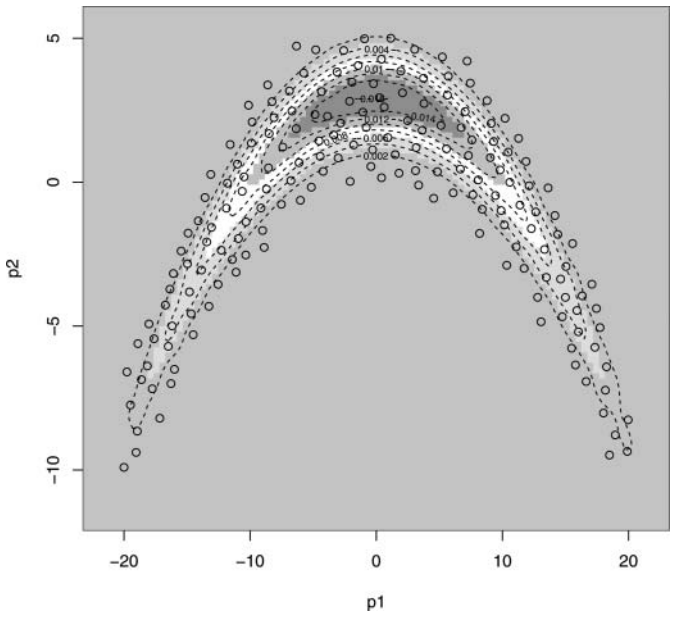

Figure 2. The DoIt approximation using a subset of design points in the banana example. The online version of this figure is in color.

1.3 Quasi-Monte Carlo

Professors Dasgupta and Meng raise the question: Is DoIt just for Quasi-Monte Carlo (QMC)? I would like to clarify that although both QMC and DoIt share the same goal of approximating integrals, the two approaches are quite different. The approach in QMC is to generate a low discrepancy sequence in a unit hypercube, similar in spirit to a space-filling design, and approximate the integrals by Monte Carlo (MC) average, whereas DoIt tries to model the posterior through smooth interpolation and approximate the integrals analytically. Therefore, DoIt has an advantage over QMC when the posterior densities are smooth.

To illustrate their differences, consider the binary data example in Section 2.2 of the article. The Laplace approximation is given by $\theta|y \sim N(2.37, 2.67^2)$. A van der Corput sequence (see, e.g., Monahan 2011, p. 287) of length 21 is generated and rescaled into the interval $[2.37 - 15, 2.37 + 15]$. The QMC estimates of the posterior mean and variance are given by

$$\hat{\theta}_{\text{QMC}} = E(\theta|y)_{\text{QMC}} = \frac{\sum_{i=1}^{m} h_i v_i}{\sum_{i=1}^{m} h_i} \quad \text{and}$$

$$\text{var}(\theta|y)_{\text{QMC}} = \frac{\sum_{i=1}^{m} h_i (v_i - \hat{\theta}_{\text{QMC}})^2}{\sum_{i=1}^{m} h_i}.$$

They are plotted in Figure 3. The DoIt approximations are also fitted using the same van der Corput sequence for $m = 1, \ldots, 21$. The estimates of posterior mean and variance (computed using the formulas given in Section 2.4 of the article) are

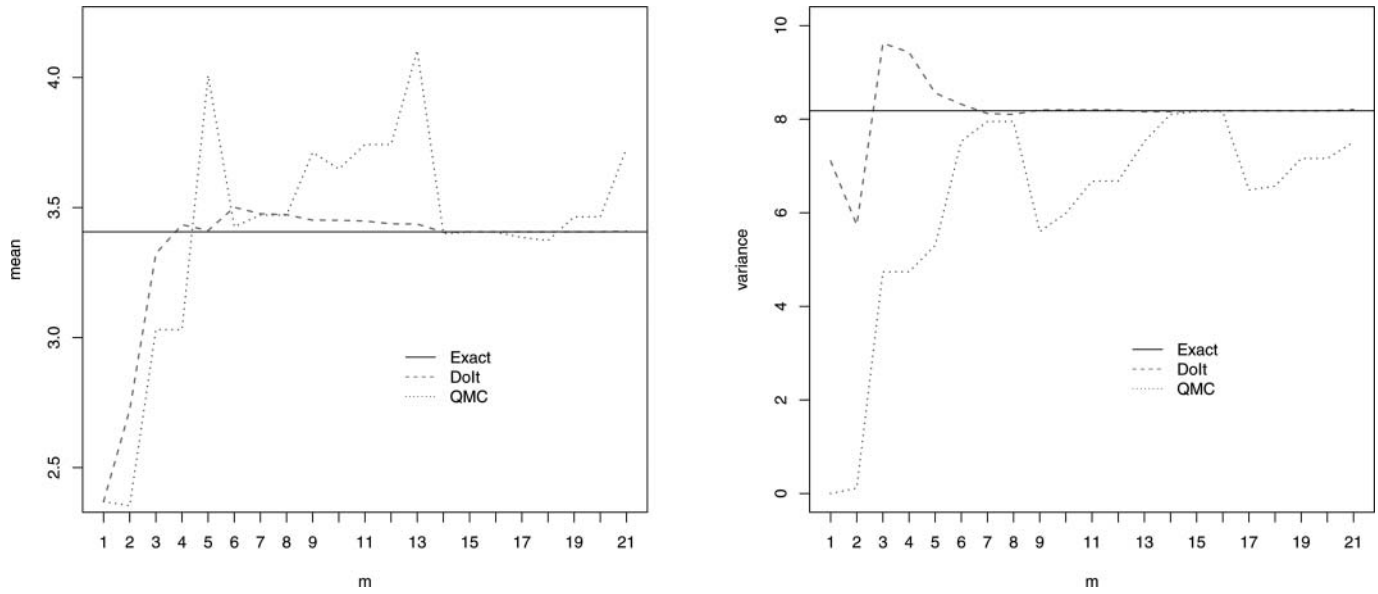

Figure 3. Estimates of the posterior mean and variance in the binary data example using a van der Corput sequence. The online version of this figure is in color.

also plotted in Figure 3. We can see that the estimates from the DoIt converge much faster to the true values than those from the QMC.

Another major difference of DoIt from QMC is that, QMC relies heavily on uniform sampling in a hypercube, whereas no such uniformity is necessary for DoIt. This is a great advantage for DoIt, because it allows DoIt to place points strategically in high probability regions using sequential design enabling a better approximation of the posterior density.

1.4 GP-MCMC

Markov chain Monte Carlo (MCMC) is a well-established methodology for Bayesian computation. So, naturally, if we encounter a computationally expensive posterior, we can try to approximate it using a Gaussian process (GP) model and then apply a suitable MCMC technique such as the one described by Fielding et al. (2011). Professors Dasgupta and Meng investigate this approach further, describing improvements over the Fielding et al. approach. They use the banana-shaped posterior example for illustration and show that even a 50-run design can approximate the posterior well using the GP-MCMC approach. I would like to note that the excellent performance of GP-MCMC in this example is not unexpected because the log-posterior is a quadratic function in θ_2 and a simple fourth-order polynomial in θ_1. The GP will do well in approximating such a simple function!

The main difference between GP-MCMC and DoIt is that the former approximates the logarithm of the unnormalized posterior using a GP model and then computes the Bayesian integrals using MCMC, whereas the latter directly approximates the unnormalized posterior using a GP model and obtains the Bayesian integrals analytically. The drawback of the DoIt's direct approximation of the unnormalized posterior is the presence of negative values, whereas the drawback of the GP-MCMC is the additional computational effort needed for simulation. Thus, both approaches have pros and cons. Here, I would like to highlight one of the major advantages of DoIt's analytical computation over a simulation-based computation. Consider a nonlinear Bayesian optimal design problem. Let $\boldsymbol{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ be the experimental design and $\boldsymbol{y}$ the data generated from a probability model $p(y|\boldsymbol{x}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are the unknown parameters. An optimal design criterion based on Shannon information gain is given by $\max_{\boldsymbol{D}} E\{\log p(\boldsymbol{\theta}|\boldsymbol{D}, \boldsymbol{y})\}$, where the expectation is taken with respect to the joint distribution of $\boldsymbol{\theta}$ and $\boldsymbol{y}$. Suppose we obtain N samples using an MC/MCMC approach. Then, the problem reduces to (see, e.g., Müller 1999)

$$\max_{\boldsymbol{D}} \frac{1}{N} \sum_{i=1}^{N} \log p(\boldsymbol{\theta}_i|\boldsymbol{D}, \boldsymbol{y}_i). \tag{1}$$

The term $p(\boldsymbol{\theta}_i|\boldsymbol{D}, \boldsymbol{y}_i)$ itself is extremely complicated to calculate. In turn, it needs to be evaluated N times to obtain Equation (1). Furthermore, since Equation (1) needs to be evaluated hundreds or thousands of times inside an optimization algorithm, this becomes a very challenging problem to solve. Having analytical expressions for such objective functions as offered by DoIt can take us a long way in developing efficient algorithms for solving such hard problems.

2. DESIGN OF EXPERIMENTS

Several issues related to the experimental design have surfaced in the discussions of Professor Lee, and Professor Steinberg and Dr. Jones. As pointed out by Lee, searching for modes using an optimization algorithm before fitting DoIt is a good idea. This is quite evident from the excellent performance of Bornkamp's iterLap algorithm. In the present implementation of the DoIt, all the evaluations done during the optimization are discarded, which can be quite wasteful. However, it is not clear how to efficiently integrate the evaluation sites visited during

the optimization with a space-filling design. This is an open research problem.

Several suggestions for improving the design were made by Steinberg and Jones; the most interesting among them is the use of a subset of design points. The examples in Section 1.2 of this rejoinder demonstrate the value of this approach. Removing points from the low-probability region helps in substantially reducing the computations without sacrificing the quality of approximation. I found that it also helps in reducing the occurrence of negative values in the DoIt approximation.

Steinberg and Jones point out that I did not make any concrete recommendation for choosing the neighborhood for the optimization in Equation (19) of the sequential design criterion. I was intentionally vague about this because we only need to invoke a local optimizer instead of a global optimizer and therefore, the neighborhood definition becomes less important. They also point out that if the initial space-filling design misses to capture some high-probability regions, then the sequential design will continue to miss it. This can be true and is probably the major weakness of the proposed sequential design criterion. The proposed criterion is capable only to improve the accuracy of the posterior in the region of sampling, but not so capable in moving out to unexplored regions. Therefore, the initial space-filling design is very crucial for the success of DoIt.

3. INTERPOLATION

Professor Lee mentions that a strict interpolation is not necessary to obtain a good fit and that a shrinkage estimator can do better. I would like to clarify that the mixture normal density shrinks the approximated posterior only at the unobserved locations and therefore, the DoIt approximation is still an interpolator. In the GP or kriging literature, shrinkage is usually achieved by introducing a nugget term into the model. Gramacy and Lee (2012) showed that such a nugget predictor can outperform the GP predictor in some cases. However, our follow-up work on this approach (Ba and Joseph, in press) shows that the prediction can be further improved by building an interpolator over the nugget predictor.

Professor Steinberg and Dr Jones point out that substantial improvements to DoIt can be made via a better choice of $\boldsymbol{\Sigma}$. I could not agree more on this! Their suggestion for using local versions of $\boldsymbol{\Sigma}$ for each evaluation sites is a brilliant idea. In fact, this is a crucial idea behind the success of Bornkamp's iterLap algorithm. In my initial investigations I have tried modifying DoIt using similar ideas, but left in frustration due to its increased computational complexity especially when dealing with computationally expensive posteriors. I believe that a clever implementation of this idea can make a substantial improvement to the current implementation of DoIt.

4. IMPLEMENTATION

Professors Ormerod and Wand, and Dr Bornkamp have asked for the computer implementation of DoIt. The online Supplementary Materials associated with this article (posted on the journal web site) contain R codes of the examples presented in the article and the rejoinder. At this moment, some of the choices such as starting point in the optimization, etc. are not automated. There is a lot more development needed for DoIt to work successfully in a wide variety of problems. The discussions have rightly pointed out the strengths and weakness of the method and have given directions for future research. I hope that this method will evolve over the next several years and will become a useful addition to the Bayesian computational toolbox. In any case, the current implementation in R will at least help you to do it!

ADDITIONAL REFERENCES

Ba, S., and Joseph, V. R. (in press), "Composite Gaussian Process Models for Emulating Expensive Functions," *Annals of Applied Statistics*. [242]

Gramacy, R. B., and Lee, H. K. H. (2012), "Cases for the Nugget in Modeling Computer Experiments," *Statistics and Computing*, 22, 713–722. [242]

Monahan, J. F. (2011), *Numerical Methods of Statistics* (2nd ed.), Cambridge: Cambridge University Press. [240]

Müller, P. (1999), "Simulation Based Optimal Design," in *Bayesian Statistics 6*, eds. J. O. Berger, J. M. Bernardo, A. P. Dawid, and A. F. M. Smith, London: Oxford University Press, pp. 459–474. [241]

Ormerod, J. T., and Wand, M. P. (2012), "Gaussian Variational Approximate Inference for Generalized Linear Mixed Models," *Journal of Computational and Graphical Statistics*, 21, 2–17. [239]