The University of British Columbia
Department of Statistics
Technical Report #230

Fast Bayesian Inference for Gaussian Process Models

Béla Nagy, Jason L. Loeppky, and William J. Welch

Department of Statistics
The University of British Columbia
333-6356 Agricultural Road
Vancouver, BC, V6T 1Z2, Canada

June 14, 2007

Abstract

In many engineering and science disciplines, deterministic computer models or codes are used to simulate complex physical processes. The computer code mathematically describes the relationship between several input variables and one or more output variables. Often the computer models in question can be computationally demanding. Thus, direct evaluation of the code for optimization or validation is not possible in general. The general strategy employed is to build a statistical model to act as a surrogate to the true underlying response surface. The approach taken here is to model the computer model as a Gaussian process with a parametric covariance function. The parameters are estimated from available data attained by running the computer code at chosen design sites. In many cases the uncertainty due to the estimation of these parameters is ignored. It is well known that ignoring this uncertainty can lead to variance estimates of predictions that are smaller than they should be. Bayesian methods can be used to account for parameter uncertainty and can lead to valid assessments of prediction uncertainty. However, Bayesian methods can be computationally expensive and difficult to implement as a black box. In this article a fast Bayesian method is presented that is both computationally efficient and easily implemented as a black box. Simulation results show that the proposed fast Bayesian method achieves remarkably accurate uncertainty estimates. This is particularly true when the dimensionality of the input space exceeds five.

1 Introduction

Computer models have been used with great success throughout the sciences and engineering disciplines, for example in climate modeling, aviation, semiconductor design, nuclear safety, etc. Implemented as computer programs, deterministic models calculate an output y for a given input vector $\boldsymbol{x}$. Depending on the complexity of the underlying mathematical model, this can be expensive computationally, creating a need for faster approximations. A common approach is to build a statistical model to approximate the output of the computer code. This has become known as the field of "computer experiments" in statistics, using Gaussian process (GP) models as computationally cheap surrogates (Sacks, Welch, Mitchell and Wynn 1989). Trading off accuracy for speed is acceptable as long as we can measure how much the surrogate's prediction of the response might deviate from the real one. However, quantifying that uncertainty has been an ongoing challenge.

In this paper we are dealing only with the prediction uncertainty that is inherent in the GP model. Of course, there are other sources of uncertainty. For example, such a simple statistical model may be an oversimplification of the complex original model. But that is outside of the scope of this investigation. Our focus is on prediction within the class of functions defined by the GP model. Quantifying prediction uncertainty means computing a prediction band about the predictor. For example, it is well-known that the plugin prediction variance formula underestimates the true uncertainty because it does not incorporate the variability due to estimating the model parameters. This leads to overly optimistic prediction bands about the predictor. A comprehensive treatment of this issue and a literature review was provided by Abt (1999).

Bayesian methods can deal with parameter uncertainty by treating them as random variables instead of fixed, unknown quantities, e.g. as in Handcock and Stein (1993). However, in practice, the cost of taking a fully Bayesian approach can be prohibitive, since non-specialists are typically unable to perform the required careful design and fine-tuning. Furthermore, there is a risk that the results will be wrong or inconclusive or just simply too slow. Iterative algorithms, such as Markov chain Monte Carlo (MCMC) may take a long time to converge and there is no definitive test to detect convergence. It is also desirable for a method to be a "black box", so that the user does not need to know its inner workings. Unfortunately, the need for data-specific hand tuning makes this an impossible dream. Our proposed solution is a fast black box Bayesian method that does not use MCMC.

2 Main results and conclusions

According to the results of this paper, Table 1 summarizes how three different methods solve the prediction uncertainty problem. As already mentioned, the plugin method in the first row of Table 1 does not take account of parameter uncertainty, but is computationally convenient. The Fast Bayesian Inference (FBI) in the second row successfully combines the desirable characteristics of the other two well-established methods. It is Bayesian in motivation, but the implementation is just an extension of the plugin that requires little additional computation to take account of parameter

Method	*Computationally Efficient*	*Black Box*	*Potentially Valid*
Traditional (plugin)	Yes	Yes	No
Fast Bayesian (FBI)	Yes	Yes	Yes
Slow Bayesian (MCMC)	No	No	Yes

Table 1: Comparing three solutions of the prediction uncertainty problem.

uncertainty and makes the inference potentially valid. Hence, run times for the first two methods are not significantly different, since both are dominated by the optimization procedure used for the maximum likelihood estimation.

The third method in Table 1 uses Markov chain Monte Carlo (MCMC) that can be inefficient computationally and in general, has no satisfactory black box implementation. Of particular concern is the fact that there is no definitive test to tell whether it has converged or not. Hence, implementations tend to be wasteful by running much longer chains than necessary to ensure convergence, rather than taking the risk of too short chains compromising validity. The interested reader can find many excellent texts about MCMC, for example Gilks, Richardson and Spiegelhalter (1998) or Robert and Casella (1999).

The FBI satisfies all three evaluation criteria: it is computationally efficient, it can be implemented as a black box, and it can potentially provide valid prediction uncertainty assessments. Moreover, simulations suggest that validity can improve as the number of dimensions increases (best if higher than five). This is a highly unusual feature that is the opposite of the "curse of dimensionality" (an expression commonly used to describe deteriorating performance with increasing dimensionality). Before presenting detailed findings in Section 4, we briefly summarize why each of the three criteria in Table 1 is important.

Computationally Efficient Efficiency is important, because eventually the curse of dimensionality impacts computation. Run times can quickly exceed what is practical as the dimensionality of the input space grows. This is especially critical in competitive industries, where shrinking design cycles create unrelenting pressure for ever faster procedures. The FBI improves the validity of the plugin method while retaining its computational complexity. This way the FBI is able to take advantage of the Bayesian idea of incorporating parameter uncertainty without paying the usual Bayesian price in escalating the computational burden.

Black Box Unless the method is a black box, its adoption will be severely restricted. Researchers or highly qualified professionals may use it for their own purposes, but non-specialists cannot be expected to read research papers or to tinker with source code and the impact on the economy is likely to stay negligible. The FBI can be readily automated, completely eliminating the need for problem-specific coding or tuning. This will enable the end user to harvest the benefits of the Bayesian approach without having to hire a Bayesian statistician or computer scientist specializing in Gaussian processes.

Potentially Valid Statisticians frequently argue that a point estimate is useless without a corresponding standard error or confidence interval. A valid $100(1-\alpha)\%$ confidence interval is expected to cover the true value approximately $100(1-\alpha)\%$ of the time over repeated realizations, where α can take any value between zero and one. For example, 95% validity gives us confidence that on average we are right 19 times out of 20. However, in the simulations of Section 4, the true coverage was significantly less than the nominal when using the prediction variance formula for the plugin, making the inference invalid. In terms of validity, the FBI outperformed the plugin in a wide range of experimental setups. More importantly, in those cases that are of practical interest, it achieves remarkably accurate uncertainty assessments. This is a very desirable property from the decision makers' perspective because it will enable them to attach correct confidence levels to the predictions.

The Gaussian process model is described in the next section. After that Section 4 compares the validity of the three inference methods with details of the assessments in Section 5. Finally, Section 6 reviews related work and Section 7 provides a summary and suggestions for future research.

3 The Gaussian process model

Equation (1) in Sacks et al. (1989) gave the following model for a deterministic computer code $y(\boldsymbol{x})$:

$$Y(\boldsymbol{x}) \;=\; \sum_{j=1}^{k} \beta_j f_j(\boldsymbol{x}) \;+\; Z(\boldsymbol{x}),$$

that is the sum of a regression model and a GP model $Z(\boldsymbol{x})$ with mean zero. However, the regression component often can be replaced by a constant mean or omitted altogether (Chen 1996; Steinberg and Bursztyn 2004), because of the flexibility of the stochastic process that can easily take on the features of the underlying function. Thus we model the computer code $y(\boldsymbol{x})$ as if it is a realization of a Gaussian stochastic process $Z(\boldsymbol{x})$ on the d-dimensional vector $\boldsymbol{x}$:

$$Y(\boldsymbol{x}) \;=\; Z(\boldsymbol{x}).$$

Setting all the β's to zero leaves only the parameters that play a role in the covariance function:

$$\mathrm{Cov}(Z(\boldsymbol{w}),\, Z(\boldsymbol{x})) \;=\; \sigma^2\, R(\boldsymbol{w},\, \boldsymbol{x}),$$

where σ^2 is the process variance and $R(\boldsymbol{w},\, \boldsymbol{x})$ is the correlation between two configurations of the input vector, $\boldsymbol{w}$ and $\boldsymbol{x}$:

$$R(\boldsymbol{w},\, \boldsymbol{x}) \;=\; \prod_{i=1}^{d} \exp\left\{-\theta_i (w_i - x_i)^2\right\},$$

where the positive θ_i range parameters control how variable the process is in a particular dimension. This is the Squared Exponential or Gaussian correlation function that is frequently used in computer experiments to model the output of deterministic computer code.

The likelihood is a function of σ^2 and the d-dimensional vector of range parameters $\boldsymbol{\theta}$:

$$L(\sigma^2,\, \boldsymbol{\theta}) \;\propto\; \frac{1}{(\sigma^2)^{\frac{n}{2}}\, |R|^{\frac{1}{2}}} \;\exp\left\{-\frac{\boldsymbol{y}^T R^{-1} \boldsymbol{y}}{2\sigma^2}\right\}, \tag{1}$$

where $\boldsymbol{y}$ is the data vector of length n and R is the $n \times n$ design correlation matrix that is a function of $\boldsymbol{\theta}$. If $\boldsymbol{\theta}$ is known, then the Best Linear Unbiased Predictor (BLUP) of the response at a new $\boldsymbol{x}_0$ is

$$\hat{y}_0(\boldsymbol{\theta}) \;=\; r(\boldsymbol{x_0})^T R^{-1} \boldsymbol{y}, \tag{2}$$

where $r(\boldsymbol{x_0})$ is a vector of correlations between the new $\boldsymbol{x}_0$ and the design points (a function of $\boldsymbol{\theta}$).

Furthermore, if σ^2 is also known, then the Mean Squared Error of the BLUP is

$$\mathrm{MSE}_{\hat{y}_0}(\sigma^2,\, \boldsymbol{\theta}) \;=\; \sigma^2 \left(1 \;-\; r(\boldsymbol{x_0})^T R^{-1} r(\boldsymbol{x_0})\right), \tag{3}$$

and these two formulas enable one to construct valid normality-based point-wise prediction bands (having a perfect match between the nominal and the true coverage at all levels under this model). However, that validity is dependent on the assumption that all parameters are known.

But in practice, usually none of the parameters are known. Instead, they have to be estimated by maximizing (1) to get the estimates $\hat{\sigma}^2$ and $\hat{\boldsymbol{\theta}}$. When we plug in $\hat{\sigma}^2$ in place of σ^2 and $\hat{\boldsymbol{\theta}}$ in place of $\boldsymbol{\theta}$ in (2) and (3), we lose validity in the sense that the estimator of (3) based on $\hat{\sigma}^2$ and $\hat{\boldsymbol{\theta}}$ is biased to be too small relative to the true mean squared error given by (3) based on σ^2 and $\boldsymbol{\theta}$. This problem is well known in both the computer experiments and the geostatistics literature (see the review in Abt (1999) for more details).

4 Simulation results

Much to our surprise, we witnessed that the potential validity of the FBI improved as we increased the number of dimensions. To dramatize that improvement, in this section we present the lowest ($d = 1$) and the highest ($d = 10$) dimensional case. (Complete results can be found in the Appendix for all $d = 1, \ldots, 10$, showing progressively improving validity from $d = 1, \ldots, 5$, plateauing at near perfection for $d = 6, \ldots, 10$).

Starting with a one-dimensional example, Figure 1 plots the true coverage probabilities (on the vertical axis) versus the nominal coverage (on the horizontal axis) of the prediction bands from 1% coverage to 99% coverage on both axes. The solid black curve is for the traditional plugin method, the dashed one is for the FBI, and the dotted one is for the MCMC. The gray diagonal represents the "optimal" or "perfectly valid" solution of the prediction uncertainty problem (i.e. the ideal method would achieve a perfect match between the nominal and true coverages, providing a curve matching this diagonal).

Around the 1% mark all three curves are indeed very close to the diagonal and that means that if one wanted a 1% prediction band, then any method would be valid. However, in practice, we are typically more interested in the high end (90% or greater). For example, the end points of the three curves in the top-right corner are for the 99% nominal confidence level. Clearly, we are not getting 99% coverage. The true coverage probability (CP) is less for all methods. However, the CPs of the two Bayesian methods are closer to the required nominal 99% coverage than the CP of the plugin (a finding that was consistently replicated in all dimensions).

The simulation in Figure 1 used $\theta = 2$ for the range parameter and $n = 10$ data points.

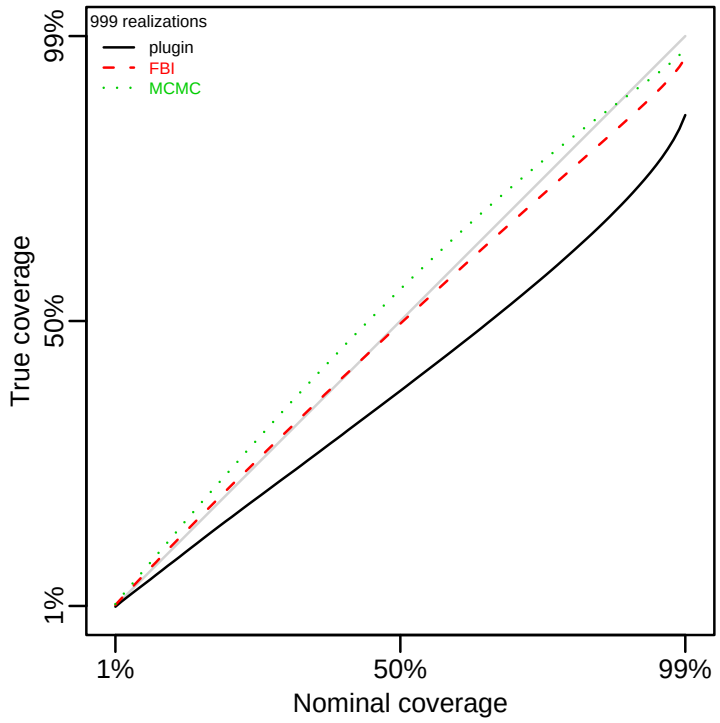


Figure 1: Coverage probabilities for $\theta = 2$, $n = 10$, and $d = 1$.

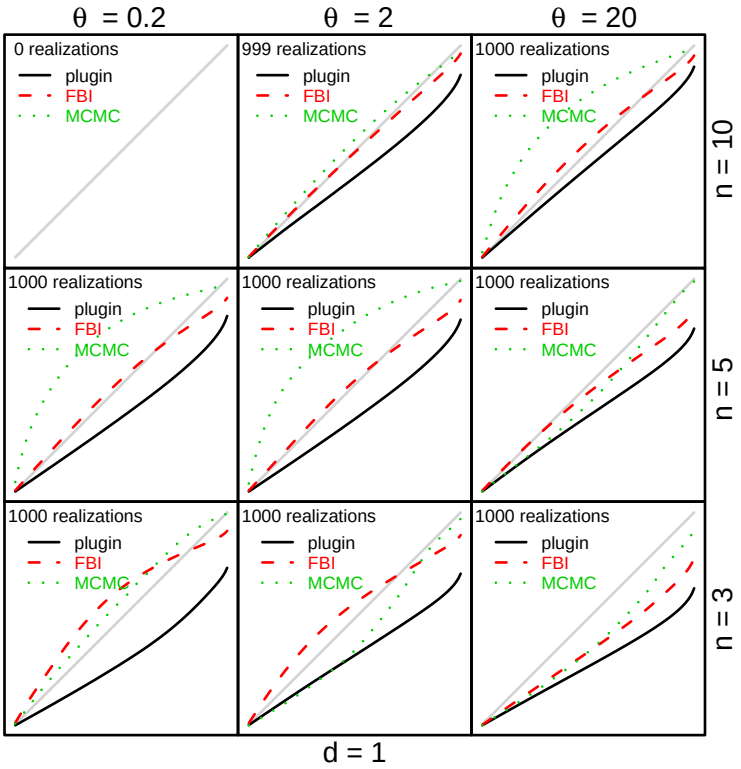


Figure 2: One-dimensional simulation results.

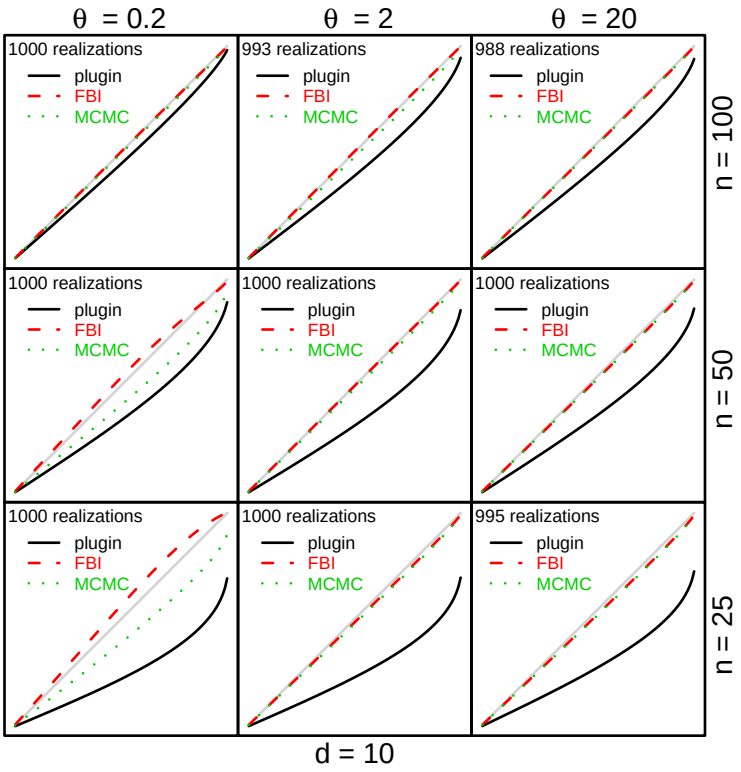


Figure 3: 10-dimensional simulation results.

Choosing $n = 10$ for $d = 1$ follows the recommendation of Sacks (e.g. Chapman, Welch, Bowman, Sacks and Walsh (1994)), who observed that ideally, the sample size should be at least 10 times the number of dimensions. However, that is not always feasible, so in addition to the rule of thumb $n = 10\,d$, we also explored $n = 10\,d/2$ and $n = 10\,d/4$ for all $d = 1, \ldots, 10$ (even though $10\,d/4$ is much too small to be useful in practice).

Setting θ to 2 is a "reasonable" choice, in a sense that numbers around 2 frequently arise in applications. We also tried two extreme cases to see how the three methods compare outside that "reasonable" domain: $\theta = 0.2$ and $\theta = 20$. In fact, setting θ to 0.2 already pushes the limits of the standard double precision representation in the one input case: numerical difficulties arise because the high correlations in the $n \times n$ correlation matrix (all close to one) make it ill-conditioned (nearly singular). Hence, computations failed for the $\theta = 0.2$, $n = 10$ case for $d = 1$.

All the other combinations of the three levels of θ and n for $d = 1$ are summarized in Figure 2. Note that the simulation in the top-left corner was the only case that completely failed for $d = 1$. Also note that the plot next to that (in the middle of the top row) is just a shrunk version of Figure 1, that is based on 999 realizations because one of the realizations (out of the total $1,000$) had to be excluded because of numerical difficulties. For the rest, computations were successfully completed for all $1,000$ simulated data sets (realizations of the GP), as indicated by the realization count in the top-left corner of each plot. (Calculations of the CPs were always restricted to the "successful" subset of the $1,000$ realizations and all failures were excluded).

Figure 3 summarizes the simulations for $d = 10$. Compared to $d = 1$, the difference is astonishing: overall, the Bayesian methods' validity improved dramatically by increasing d, but the plugin's showed much slower change. Perhaps the most surprising 10-dimensional result is that the Bayesian methods achieved almost perfect prediction uncertainty assessments at all coverage levels (from 1% to 99%) in seven cases out of nine. The two exceptions are for $\theta = 0.2$ with $n = 50$ and $n = 25$, where the FBI tends to become too conservative leading to over-coverage, while the MCMC results in significant under-coverage. Similar patterns can be seen for the other values of $d > 5$. Complete results and implementation details can be found in the Appendix.

5 Methods

5.1 Plugin

The plugin method is straightforward and we have already described it in Section 3. To compare it with the other two methods, here we are presenting it as an algorithm that has two steps: estimation and prediction.

1. Obtain the Maximum Likelihood Estimates (MLEs) of the parameters: $\hat{\sigma}^2$ and $\hat{\boldsymbol{\theta}}$.

2. For prediction, plug in the MLEs into equations (2) and (3) as if they were the true parameters.

5.2 Fast Bayesian Inference

Bayesian analysis starts with a prior distribution on the model parameters. We recommend the uniform prior on the log scale for both the process variance and the range parameters. This has several advantages, although the propriety of the posterior is not guaranteed (Berger, De Oliveira and Sansó 2001).

First of all, the uniform prior is the fastest of all priors, since it does not need any computation (the posterior is proportional to the likelihood). The second advantage is that σ^2 can be integrated out analytically from (1) to get the integrated likelihood that is a function of only $\boldsymbol{\theta}$:

$$L(\boldsymbol{\theta}) \propto (\boldsymbol{y}^T R^{-1} \boldsymbol{y})^{-\frac{n}{2}} |R|^{-\frac{1}{2}}.$$

It is interesting to note that $L(\boldsymbol{\theta}) = L(\hat{\sigma}^2(\boldsymbol{\theta}), \boldsymbol{\theta})$, where

$$\hat{\sigma}^2(\boldsymbol{\theta}) = \frac{\boldsymbol{y}^T R^{-1} \boldsymbol{y}}{n}, \tag{4}$$

which means that $L(\boldsymbol{\theta})$ can also be viewed as the profile likelihood that is maximized over all σ^2 given $\boldsymbol{\theta}$. Proofs can be found in Nagy, Loeppky and Welch (2007).

The third advantage is that the log transformation for $\boldsymbol{\theta}$ brings the profile/integrated likelihood closer to the normal distribution. This was shown by Nagy et al. (2007) for the one-dimensional case, and simulations suggested that the log transformation reduced non-normality in higher dimensions as well.

Using the notation $\boldsymbol{\tau} = (\log \theta_1, \ldots, \log \theta_d)^T = \log \boldsymbol{\theta}$ for the transformed parameter vector and $\boldsymbol{\theta} = (\exp \tau_1, \ldots, \exp \tau_d)^T = \exp \boldsymbol{\tau}$ for the inverse transformation, the new likelihood function $L(\exp \boldsymbol{\tau})$ tends to have a shape that is closer to a normal with respect to $\boldsymbol{\tau}$ than the shape of the original $L(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. Using a uniform prior for the transformed parameters means that this new likelihood function as a function of $\boldsymbol{\tau}$ is also the (unnormalized) marginal posterior for $\boldsymbol{\tau}$.

The fourth advantage is that the log transformation makes the numerical optimization of the likelihood/posterior unconstrained: $\boldsymbol{\tau} \in \mathbb{R}^d$. This is the first step of the Fast Bayesian Inference, that can be summarized as follows:

1. Maximize the log-likelihood/log-posterior $\log L(\exp \boldsymbol{\tau})$ to get the MLE of $\boldsymbol{\tau}$, denoted $\hat{\boldsymbol{\tau}}$.
2. Compute the Hessian (the matrix of second derivatives) at $\hat{\boldsymbol{\tau}}$, denoted $H_{\hat{\boldsymbol{\tau}}}$.
3. Sample from the multivariate normal distribution $N(\hat{\boldsymbol{\tau}}, -H_{\hat{\boldsymbol{\tau}}}^{-1})$ to obtain M Monte Carlo samples: $\boldsymbol{\tau}^{(1)}, \ldots, \boldsymbol{\tau}^{(M)}$.
4. Following standard Bayesian practice, the FBI predictor is given by the average:

$$\frac{1}{M} \sum_{i=1}^{M} \hat{y}_0(\exp \boldsymbol{\tau}^{(i)}),$$

and its Mean Squared Error is given by the the variance decomposition formula:

$$\frac{1}{M} \sum_{i=1}^{M} \mathrm{MSE}_{\hat{y}_0}\left(\hat{\sigma}^2(\exp \boldsymbol{\tau}^{(i)}), \exp \boldsymbol{\tau}^{(i)}\right) + \frac{1}{M-1} \sum_{j=1}^{M} \left(\hat{y}_0(\exp \boldsymbol{\tau}^{(j)}) - \frac{1}{M} \sum_{i=1}^{M} \hat{y}_0(\exp \boldsymbol{\tau}^{(i)}) \right)^2,$$

that is the average MSE of the plugin predictors plus the sample variance of those predictors. It is instructive to compare this sequence to that of the plugin in the previous subsection. We can see that the first steps are equivalent: both methods start by locating the MLE. After that the plugin method jumps into the prediction phase right away assuming that the value found at the mode is

the one best estimate of the truth.

The FBI is more careful. In the second step it looks at the curvature at the mode to quantify the uncertainty in the estimation of the *point* estimate. For example, if the surface is flat, that means high uncertainty and the corresponding normal approximation in step 3 will have a high variance reflecting that uncertainty.

In the final step, the FBI averages predictions based on the sample from that normal distribution as if it was from the true posterior. Again, there is a part that is identical to the plugin method, since for each sample point, equations (2) and (3) are used to calculate the predictor and its Mean Squared Error, respectively (also using (4) to estimate σ^2 for a given $\boldsymbol{\tau}^{(i)}$ in the sample). This way the FBI will have many predictions to average (one for each sample point), while the plugin method will have just one. Hence, the plugin can be viewed as a special case of the FBI with sample size one.

5.3 Markov chain Monte Carlo

There are many possible ways of constructing an MCMC algorithm to sample from a distribution that is only known up to a scale. One of the simplest is the Metropolis random walk algorithm (Metropolis, Rosenbluth, Rosenbluth, Teller and Teller 1953) that has been used successfully in many high-dimensional problems. To enable direct comparison with the FBI, everything was done on the log scale using the same $\boldsymbol{\tau}$-parameterization. Also, the first two moments of the $N(\hat{\boldsymbol{\tau}},\ -H^{-1}_{\hat{\boldsymbol{\tau}}})$ normal approximation for the FBI were utilized to help the implementation in step 1 and step 3 of the algorithm, respectively:

1. Initialize $\boldsymbol{\tau}^{(1)}$ at $\hat{\boldsymbol{\tau}}$.
2. To select a direction for a random walk step, sample an integer j uniformly from $1, \ldots, d$.
3. Given the current $\boldsymbol{\tau}^{(i)}$, set $\boldsymbol{\tau}^*$ to $\boldsymbol{\tau}^{(i)}$ and then add to the jth coordinate of $\boldsymbol{\tau}^*$ a normal random deviate with mean zero and standard deviation equal to three times the standard error in the jth dimension, estimated from the Hessian: $\sqrt{-H^{-1}_{\hat{\boldsymbol{\tau}}}(j,j)}$.
4. Compute the acceptance ratio for $\boldsymbol{\tau}^*$, given $\boldsymbol{\tau}^{(i)}$:
$$\alpha \ = \ \min\left\{ 1,\ \frac{L(\exp \boldsymbol{\tau}^*)}{L(\exp \boldsymbol{\tau}^{(i)})} \right\}.$$
5. Set $\boldsymbol{\tau}^{(i+1)}$ to $\boldsymbol{\tau}^*$ with probability α and to $\boldsymbol{\tau}^{(i)}$ with probability $1-\alpha$.
6. Repeat steps 2–5 until i reaches the desired sample size.

When this algorithm works well, it constructs a Markov chain whose stationary distribution is the posterior distribution. The resulting sample then can be used for prediction exactly the same way as the sample for the FBI (step 4 in Section 5.2). In other words, once the sampling is done, the treatment of the samples are identical.

But that does not mean that the samples are equivalent or similar. The FBI draws an independent, identically distributed (iid) sample from the normal approximation of the posterior. In contrast, the MCMC algorithm constructs a dependent sample from the original posterior. That immediately explains why the MCMC is so much slower than the FBI: because the sample is not

iid, it needs a much larger sample size (exactly how large is an open question).

Another difference is that the FBI always samples from a proper density function but that is not guaranteed for the MCMC. Unlike its normal approximation, the original posterior may not be proper (i.e. the integral is not finite) and in that case the sample collected by the MCMC is meaningless because the posterior is not proportional to any density function. This fact alone should be enough to deter anyone from using this method (or any other MCMC algorithm) in a black box fashion.

But there are other reasons, too. For example, diagnostic tools are not black box either. Among other things, typically, one is expected to look at the trace plots after each run. But in our simulation study there were almost $89,000$ MCMC attempts in total. Clearly, we had to find a more efficient way for evaluating success or failure.

We ended up with two arbitrary, but not very restrictive minimum cut-off values for the "acceptance rate" and the "mean effective sample size" measures and only allowed samples that met both criteria. All other realizations were classified as failures and not used in any further calculations (see the Appendix for more details).

In summary, it is difficult to know to what extent the two criteria detected non-convergence or any other pathology of the MCMC sample. Nevertheless, this questionable attempt at black box MCMC showed fairly good overall validity. Relatively few runs had to be disqualified (usually less than 10 out of $1,000$), enabling head-to-head comparison with the FBI. Conditionally on the success of the remaining (qualifying) runs, differences seen between the MCMC and the FBI should reflect the difference between the original posterior and its normal approximation.

6 History and related work

Normal approximations based on posterior modes are certainly not new. The idea can be traced back to Laplace (1774). However, it appears to be underutilized in this context. Williams and Barber (1998) used the Laplace approximation for Gaussian process classification. Karuri (2005) used the normal approximation for GP regression in one and two dimensions and observed that on the log scale the posteriors were closer to normal. Nagy et al. (2007) showed that in the one-dimensional case the log transformation improved approximate normality of the likelihood/posterior when using the Squared Exponential (Gaussian) correlation function.

7 Discussion

Fast Bayesian Inference represents a middle ground between two extremes. The traditional plugin method is extreme because it makes inference based solely on the estimate found at the mode, ignoring the uncertainty around it (its sample size is one). At the other extreme, the slow Bayesian method is inefficient because it ignores the mode and constructs a large dependent sample as it explores every corner of the posterior by MCMC.

The FBI corrects the plugin's deficiency by incorporating the parameter uncertainty around the mode. Unlike the slow Bayesian method, the FBI does not need a huge sample (or burn-in), because its sample is iid. This is the main advantage of sampling directly from the normal approximation, instead of the original posterior that is only known up to a scale.

Using the uniform prior for the log transformed parameters has four advantages:

1. The prior needs no computation, since the likelihood is the unnormalized posterior.
2. The process variance can be integrated out to reduce dimensionality by one.
3. The log transformation reduces non-normality of the likelihood/posterior.
4. The log transformation enables the use of unconstrained optimization algorithms.

The latter two suggest that they can be combined in numerically stable situations for a fifth advantage: a dramatic speed up of the numerical optimization by Newton's method that can double the number of correct digits at each iteration (quadratic convergence). When the log transformation makes the likelihood/posterior nearly normal, then the log-likelihood/log-posterior becomes nearly quadratic, and that is the kind of function that can be optimized very efficiently with Newton-type algorithms. However, more work is needed to determine when this can be done reliably, because often Newton's method is not as robust as derivative-free optimizers.

In summary, the FBI presents a compelling solution to the prediction uncertainty problem by combining the benefits of the other two alternatives and avoiding their drawbacks. It is computationally efficient, it can be implemented as a black box, and it can potentially provide valid prediction uncertainty assessments.

In practice, it can save both time and money. That may include both software development time/cost or run time/cost. What is perhaps the most important (and the most difficult to quantify) is the effect of the more valid predictions that can lead to better decisions.

The FBI is also fast to implement, especially as an add-on to an existing implementation of the plugin method, since it is a straight extension of that. It is our hope that we presented convincing arguments to facilitate its adoption without delay. Why keep using the invalid plugin, when its valid Bayesian upgrade is also fast and ready for production?

However, from the research perspective, much works remains to be done. For example, the greatest mystery is how the FBI becomes more valid as the dimensionality increases. One hypothesis is that the log transformation does not work as well in lower dimensions. To inspect this possibility, we will expand our investigation to the family of power transformations (Tukey 1957).

Finally, it is important to point out that when one expects the FBI to give valid predictions, one needs to keep in mind the two fundamental limitations of our study. The first one is that all our data came from the true Gaussian process model. But for real data, the assumption of a zero-mean stationary Gaussian process (with the Gaussian correlation function) may be inadequate or totally wrong and results will be entirely dependent on the real underlying function.

The second serious limitation is that we studied the frequentist properties of the prediction bands in terms of coverage probabilities. Hence, validity is implied only over a long sequence of identical trials, according to the classical frequentist interpretation. But in practice, most of the time there is just one unique data set.

However, the use of this criterion is not limited to frequentists. It is not uncommon for Bayesians to use it as a "sanity check" for their Bayesian credible regions. For example, Bayarri and Berger (2004) argue that "there is a sense in which essentially everyone should ascribe to frequentism" and provide the following version of the frequentist principle: "In repeated practical use of a statistical procedure, the long-run average actual accuracy should not be less than (and ideally should equal) the long-run average reported accuracy".

Acknowledgments

This research was funded by the Natural Sciences and Engineering Research Council of Canada and the National Program on Complex Data Structures of Canada. Computations were made using WestGrid, which is funded in part by the Canada Foundation for Innovation, Alberta Innovation and Science, BC Advanced Education, and the participating research institutions. WestGrid equipment is provided by IBM, Hewlett Packard and SGI.

References

ABT, M. (1999). Estimating the prediction mean squared error in Gaussian stochastic processes with exponential correlation structure. *Scandinavian Journal of Statistics,* **26** 563–578.

BAYARRI, M. J. and BERGER, J. O. (2004). The interplay of Bayesian and frequentist analysis. *Statistical Science,* **19** 58–80.

BERGER, J. O., DE OLIVEIRA, V. and SANSÓ, B. (2001). Objective Bayesian analysis of spatially correlated data. *Journal of the American Statistical Association,* **96** 1361–1374.

CARTER, C. K. and KOHN, R. (1994). On Gibbs sampling for state space models. *Biometrika,* **81** 541–553.

CHAPMAN, W. L., WELCH, W. J., BOWMAN, K. P., SACKS, J. and WALSH, J. E. (1994). Artic sea ice variability: Model sensitivities and a multidecadal simulation. *Journal of Geophysical Research,* **99** 919–936.

CHEN, X. (1996). *Properties of Models for Computer Experiments.* Ph.D. thesis, University of Waterloo.

GILKS, W. R. E., RICHARDSON, S. E. and SPIEGELHALTER, D. J. E. (1998). *Markov Chain Monte Carlo in Practice.* Chapman & Hall Ltd.

HANDCOCK, M. S. and STEIN, M. L. (1993). A Bayesian analysis of kriging. *Technometrics,* **35** 403–410.

KARURI, S. W. (2005). *Integration in Computer Experiments and Bayesian Analysis*. Ph.D. thesis, University of Waterloo.

LAPLACE, P. S. (1774). Memoir on the probability of the causes of events. Tome Sixième. Mémoires de Mathématique et de Physique (English translation by S. M. Stigler 1986. Statist. Sci., 1(19):364-378).

MCKAY, M. D., BECKMAN, R. J. and CONOVER, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics,* **21** 239–245.

MEASE, D. and BINGHAM, D. (2006). Latin hyperrectangle sampling for computer experiments. *Technometrics,* **48** 467–477.

METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., TELLER, A. H. and TELLER, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics,* **21** 1087–1091.

NAGY, B., LOEPPKY, J. L. and WELCH, W. J. (2007). Correlation parameterization in random function models to improve normal approximation of the likelihood or posterior. Tech. Rep. 229, Department of Statistics, The University of British Columbia. URL `http://www.stat.ubc.ca/Research/TechReports/techreports/229.pdf`.

ROBERT, C. P. and CASELLA, G. (1999). *Monte Carlo Statistical Methods*. Springer-Verlag Inc.

SACKS, J., WELCH, W. J., MITCHELL, T. J. and WYNN, H. P. (1989). Design and analysis of computer experiments (C/R: P423-435). *Statistical Science*, **4** 409–423.

STEINBERG, D. M. and BURSZTYN, D. (2004). Data analytic tools for understanding random field regression models. *Technometrics*, **46** 411–420.

TUKEY, J. W. (1957). On the comparative anatomy of transformations. *The Annals of Mathematical Statistics*, **28** 602–632.

WILLIAMS, C. K. I. and BARBER, D. (1998). Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20** 1342–1351.

Appendix

The first part of the Appendix describes the simulation procedure and explains why it is relevant to current practice. Then a table is presented summarizing the coverage probabilities for the nominal 90%, 95%, and 99% levels. That is followed by 10 figures for the the complete simulation results in $d = 1, \ldots, 10$, where d is the dimensionality of the input space.

The simulation plan can be viewed as a set of 10 statistically designed experiments for $d = 1, \ldots, 10$. For each experiment, the design was a 3×3 full-factorial with $1,000$ replicates. The two factors were the range parameter θ and the sample size n, both at three levels (equally spaced on the log scale): $\theta = 0.2,\ 2,\ 20$ and $n = 10\,d/4,\ 10\,d/2,\ 10\,d$ (where $10\,d/4$ was rounded up to the nearest integer).

To obtain $1,000$ replicates for a given combination of θ and n, the following four steps were repeated (attempted) $1,000$ times:

1. Select an n point design by Latin hypercube sampling[1] in the d-dimensional unit hypercube $[0,\ 1]^d$.
2. Generate a realization $\boldsymbol{y}$ of the Gaussian process over the n design points by setting the range parameter to θ in all dimensions and the process variance to one.
3. Sample 10 new points uniformly in the unit hypercube $[0,\ 1]^d$ for prediction.
4. Compute the predictors for the three methods with their mean squared errors for the 10 new points from the data $\boldsymbol{y}$.

[1] Although there are many improved variants of Latin hypercubes, e.g. Mease and Bingham (2006), the original random version of McKay, Beckman and Conover (1979) was used here because of the enormous number of realizations generated.

This sequence was devised to represent a typical real world scenario. Latin hypercubes are the design of choice for GP models (Sacks et al. 1989) for prediction at new, untried inputs anywhere in $[0,\ 1]^d$. Note that step 2 or 4 could fail because of numerical issues, leading to an unsuccessful realization (missing value) for that particular replicate (not included in subsequent analysis). The only case when this had a serious impact on results was the $\theta = 0.2$, $n = 10$ case in one dimension, as already discussed in Section 4.

The Monte Carlo sample size for the FBI was $M = 400$, minus those sample points that ran into numerical difficulties caused by the ill-conditioning of the correlation matrix. This happened mostly in lower-dimensional cases, especially in $d = 1$. The MCMC sample size was $N = 100,000$ (after $10,000$ burn-in). Unlike the FBI sample, the MCMC sample did not suffer from numerical problems because problematic points would never be accepted by the algorithm, since the likelihood/posterior was set to zero whenever the Cholesky-decomposition of the correlation matrix failed. An MCMC run was considered successful if the acceptance rate was at least 15% and the Mean Effective Sample Size (MESS) was at least 50. Both measures were calculated after the burn-in phase.

The following formula was used for the MESS:

$$\text{MESS} \ = \ \frac{1}{d}\sum_{i=1}^{d} N \left[1 + 2\sum_{k=1}^{1000}\left(1 - \frac{k}{N}\right)\hat{\rho}_k(i)\right]^{-1},$$

where $\hat{\rho}_k(i)$ is the kth sample autocorrelation in the ith dimension (Carter and Kohn 1994).

Another way to look at the ill-conditioning problem is to project it back to the distribution where the sample came from. For example, one could say that the FBI did not sample from a normal distribution, just a truncated normal with all numerically problematic areas having densities set to zero. One could similarly argue that the MCMC did not sample from the true posterior, because it was truncated for numerical stability and the uncomputable parts of the parameter space were excluded.

Coverage probabilities were calculated by averaging the individual CPs over all new points and all successful realizations. A realization was considered successful if all operations for all three methods completed without error. It is straightforward to compute an individual CP. Suppose that we want to predict the output Y_0 at a new, untried input $\boldsymbol{x}_0$. Since the true model is known during the simulation, we know that conditionally on the realized data, Y_0 is normally distributed with mean μ_0 and variance σ_0^2, where μ_0 and σ_0^2 are given by equations (2) and (3), respectively.

Now suppose that after estimation, the predictor for Y_0 was μ_1 with mean squared error σ_1^2. This amounts to mis-specifying the distribution of the random variable Y_0 as $N(\mu_1, \sigma_1^2)$ instead of the true $N(\mu_0, \sigma_0^2)$.

Then the CP of a normality-based $100(1-\alpha)\%$ prediction interval about μ_1 is

$$P_0\left(\mu_1 - \sigma_1\, z_{\alpha/2} \ < \ Y_0 \ < \ \mu_1 + \sigma_1\, z_{\alpha/2}\right) \ =$$

$$= \ P_0\left(\frac{\mu_1 - \sigma_1\, z_{\alpha/2} - \mu_0}{\sigma_0} \ < \ \frac{Y_0 - \mu_0}{\sigma_0} \ < \ \frac{\mu_1 + \sigma_1\, z_{\alpha/2} - \mu_0}{\sigma_0}\right) \ =$$

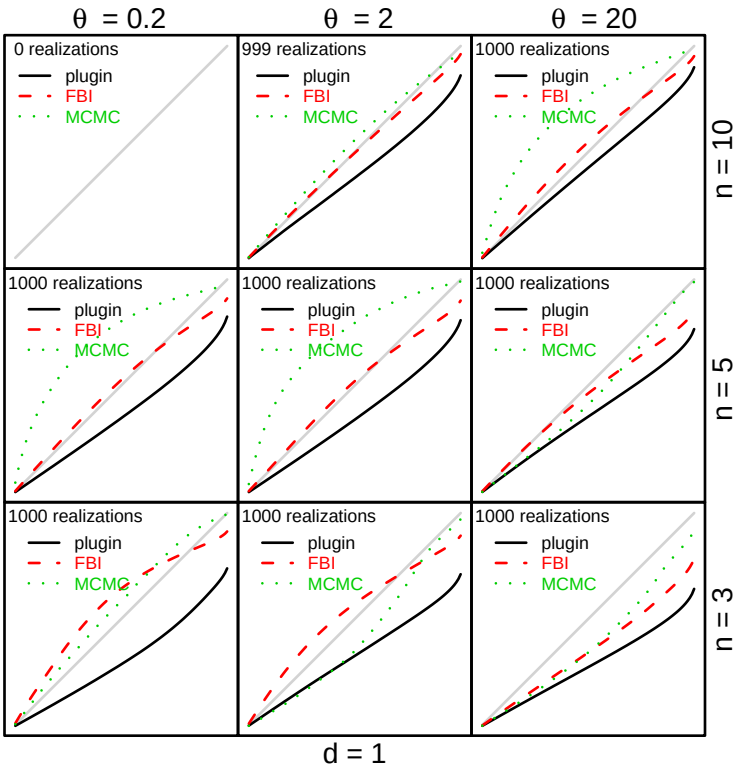
$$= \ \Phi\left(\frac{\mu_1 + \sigma_1\, z_{\alpha/2} - \mu_0}{\sigma_0}\right) \ - \ \Phi\left(\frac{\mu_1 - \sigma_1\, z_{\alpha/2} - \mu_0}{\sigma_0}\right),$$

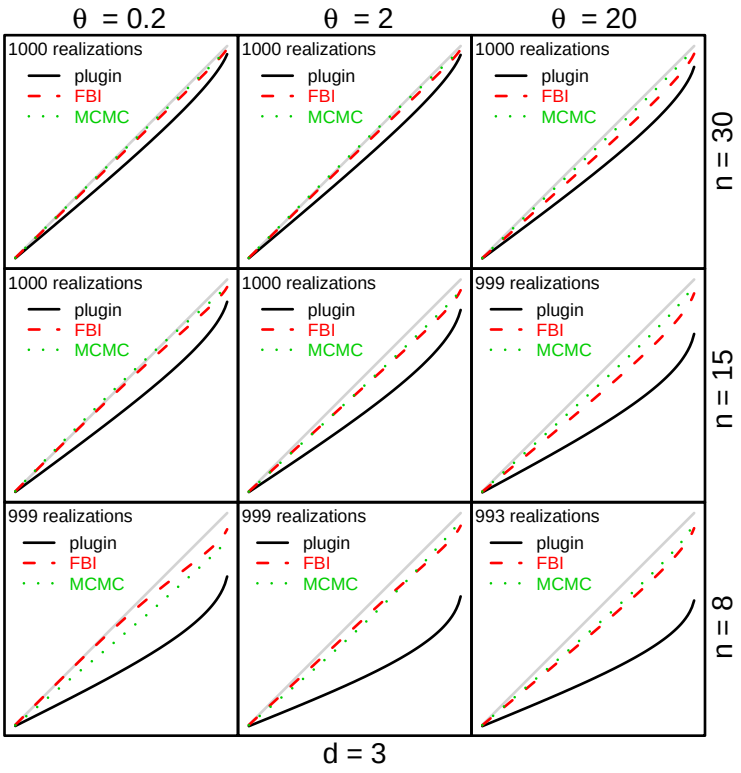
where P_0 denotes the true probability distribution, Φ is the cumulative distribution function of the standard normal $N(0,1)$, and $z_{\alpha/2}$ satisfies $\Phi(-z_{\alpha/2}) = \alpha/2$.

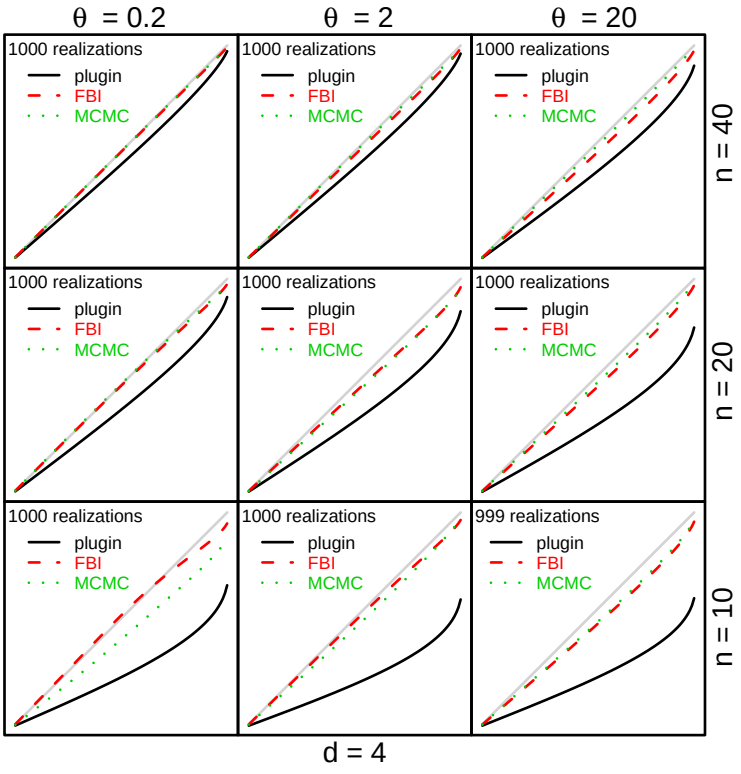
The following table is a summary of the CPs for the nominal 90%, 95%, and 99% confidence levels. The two-digit numbers in the table are truncated percentages without the percent sign and without the fractional parts (rounded down). The 3×3 arrangement inside each cell follows the layout of the plots by the three levels of θ horizontally and the three levels of n vertically.

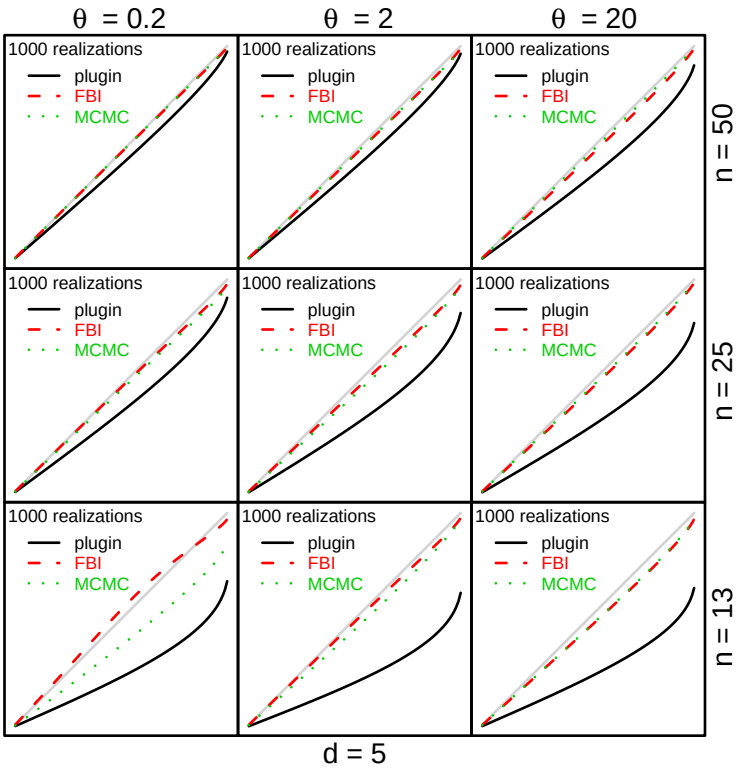
	90%			95%			99%		
	plugin	FBI	MCMC	plugin	FBI	MCMC	plugin	FBI	MCMC
$d = 1$	72 76 67 66 64 61 59 52	85 85 82 81 75 85 81 65	89 94 94 95 88 94 87 79	78 82 73 72 69 66 64 57	90 89 86 84 79 87 84 71	93 96 95 97 94 96 92 85	85 89 81 80 76 73 70 63	95 94 90 89 84 90 88 78	96 98 97 98 97 98 96 90
$d = 2$	80 80 75 71 70 59 50 45 39	87 87 83 84 82 76 76 76 68	88 89 88 89 87 85 82 85 78	87 86 82 78 76 65 56 50 44	92 92 88 89 87 82 80 81 74	93 93 93 93 92 91 87 90 84	94 94 90 86 84 74 64 58 51	96 96 94 94 92 89 86 87 82	97 97 97 96 96 96 91 95 91
$d = 3$	81 81 74 73 68 57 53 45 44	87 87 83 85 82 78 81 82 78	88 88 88 87 84 85 77 84 81	88 87 81 80 75 64 60 51 50	92 92 89 90 87 85 85 87 84	93 93 93 92 89 91 82 90 88	95 94 89 88 84 73 69 60 58	97 97 95 95 94 92 91 93 91	97 97 98 96 94 96 87 95 94
$d = 4$	83 81 74 75 67 60 49 43 44	88 87 84 86 83 84 84 85 81	89 88 88 87 83 86 75 85 83	89 88 81 82 74 66 55 49 50	93 92 90 91 89 89 89 90 87	94 93 93 92 89 92 81 90 89	96 95 89 90 84 76 65 58 59	98 97 96 96 95 96 93 95 94	98 97 98 96 94 97 87 96 95
$d = 5$	83 81 74 74 65 62 50 45 48	88 87 87 86 85 86 88 86 84	88 87 88 85 84 87 72 84 84	89 88 81 81 73 69 56 51 54	93 92 92 91 91 92 92 91 90	93 92 93 90 90 92 78 90 90	96 95 90 90 83 78 67 62 64	98 97 97 96 96 97 96 96 96	98 97 98 95 95 98 84 96 96
$d = 6$	83 80 74 74 65 63 49 46 49	88 87 88 87 87 87 90 88 85	88 86 88 82 85 87 72 85 85	89 87 81 81 72 70 56 53 55	93 92 93 92 92 92 94 93 91	93 91 94 88 91 93 78 91 91	96 95 90 90 83 80 67 63 65	98 97 98 97 97 97 97 97 97	98 96 98 94 96 98 85 96 97
$d = 7$	84 79 76 72 64 65 50 47 51	89 88 88 86 88 87 92 88 86	88 85 88 81 86 88 73 86 86	90 86 83 80 72 73 57 54 57	94 93 94 92 93 93 95 93 92	93 91 94 87 91 93 79 91 91	96 94 91 89 83 82 68 64 67	98 98 98 97 98 98 98 98 97	98 96 98 93 96 98 86 97 97
$d = 8$	84 79 76 71 64 66 49 48 51	89 88 89 87 89 88 93 88 86	89 86 89 80 86 88 73 86 86	90 86 83 79 72 74 56 55 58	94 93 94 92 94 93 96 93 92	94 91 94 86 92 93 79 92 92	97 94 91 89 83 83 67 66 68	98 98 98 97 98 98 98 98 97	98 97 98 93 97 98 87 97 97
$d = 9$	84 78 77 71 65 67 50 50 52	89 89 89 88 89 88 94 88 87	88 86 89 79 87 88 74 86 86	90 85 84 79 73 74 57 57 59	94 94 94 93 94 93 97 93 92	94 92 94 85 93 93 80 92 92	97 93 92 88 84 84 69 68 69	98 98 98 97 98 98 99 98 97	98 97 98 93 97 98 88 97 97
$d = 10$	84 77 78 70 66 68 49 50 54	89 89 89 90 89 88 94 88 87	88 87 89 79 88 88 75 87 87	90 85 84 78 74 75 57 57 61	94 94 94 94 94 93 97 93 93	94 92 94 85 93 93 81 92 92	97 93 93 88 84 85 68 69 72	98 98 98 98 98 98 99 98 98	98 97 98 93 98 98 89 97 97

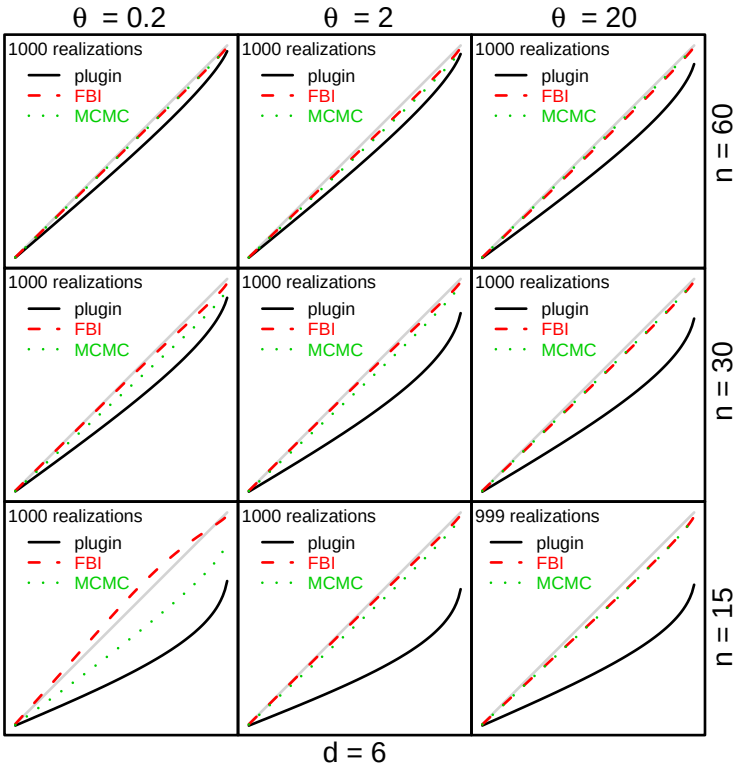
The following 10 figures compare the validity of the three methods for $d = 1, \ldots, 10$, for all combinations of the three levels of θ and the three levels of n. In addition to the gray diagonal in the middle, three curves were plotted for the three methods relating the true coverage probabilities on the vertical axis (from 1% to 99%) to the nominal coverage on the horizontal axis (from 1% to 99%). Plots are based on the realizations that were classified as successful, out of $1,000$ attempts in total. Counts for the number of realizations included in the final calculations are shown in the top-left corner of each plot.

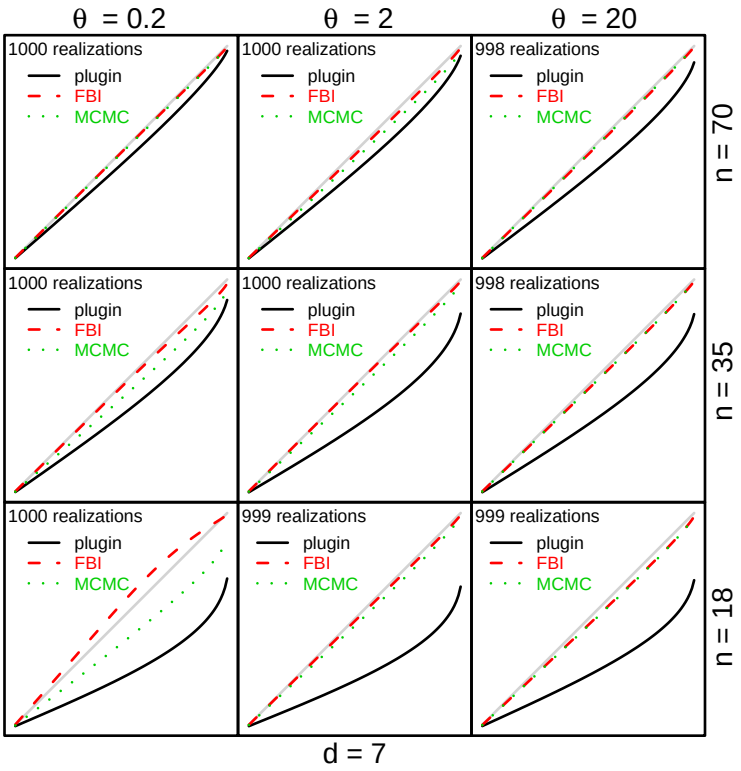

θ = 0.2
θ = 2
θ = 20
0 realizations
999 realizations
1000 realizations
987 realizations
992 realizations
925 realizations
plugin
FBI
MCMC
n = 10
n = 5
n = 3
n = 20
d = 1
d = 2

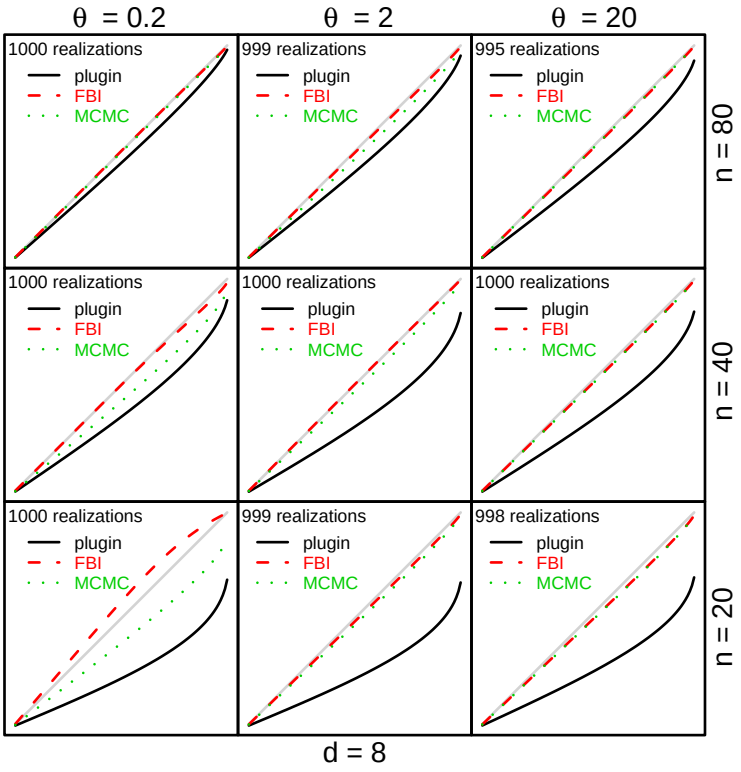

θ = 0.2
θ = 2
θ = 20
1000 realizations
1000 realizations
1000 realizations
999 realizations
1000 realizations
999 realizations
999 realizations
993 realizations
plugin
FBI
MCMC
n = 30
n = 15
n = 8
d = 3

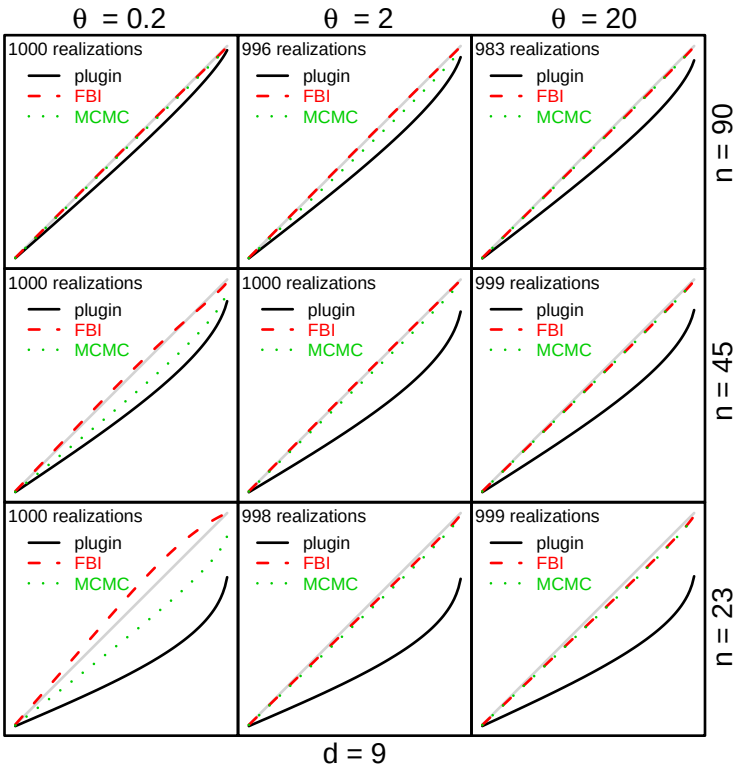

θ = 0.2
θ = 2
θ = 20
1000 realizations
plugin
FBI
MCMC
n = 40
n = 20
n = 10
d = 4

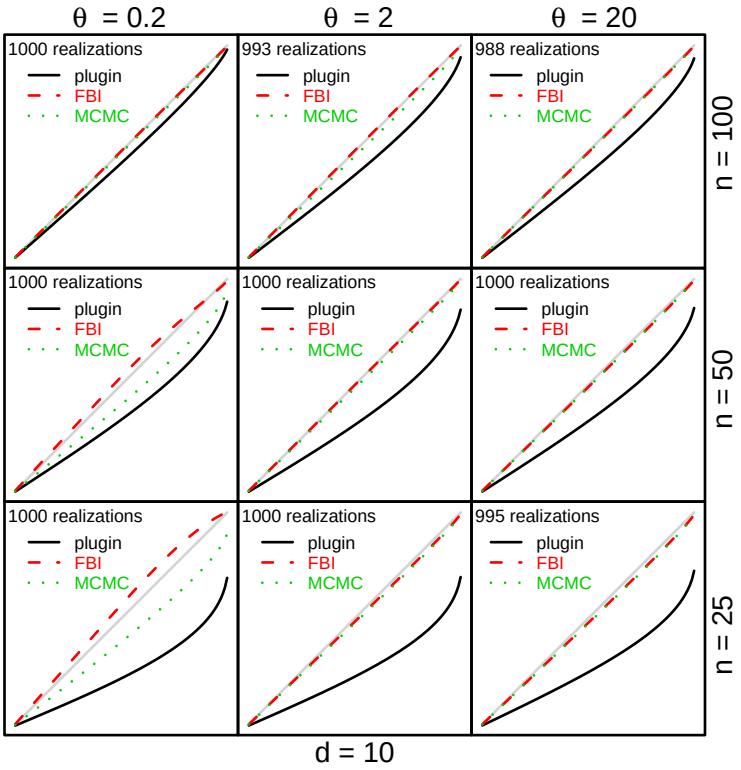

θ = 0.2
θ = 2
θ = 20
1000 realizations
plugin
FBI
MCMC
n = 50
n = 25
n = 13
d = 5


θ = 0.2
θ = 2
θ = 20
1000 realizations
999 realizations
plugin
FBI
MCMC
n = 60
n = 30
n = 15
d = 6


θ = 0.2
θ = 2
θ = 20
1000 realizations
1000 realizations
998 realizations
1000 realizations
1000 realizations
998 realizations
1000 realizations
999 realizations
999 realizations
plugin
FBI
MCMC
n = 70
n = 35
n = 18
d = 7


θ = 0.2
θ = 2
θ = 20
1000 realizations
999 realizations
995 realizations
1000 realizations
1000 realizations
1000 realizations
1000 realizations
999 realizations
998 realizations
plugin
FBI
MCMC
n = 80
n = 40
n = 20
d = 8


θ = 0.2
θ = 2
θ = 20
1000 realizations
996 realizations
983 realizations
1000 realizations
1000 realizations
999 realizations
1000 realizations
998 realizations
999 realizations
plugin
FBI
MCMC
n = 90
n = 45
n = 23
d = 9


θ = 0.2
θ = 2
θ = 20
1000 realizations
993 realizations
988 realizations
1000 realizations
1000 realizations
1000 realizations
1000 realizations
1000 realizations
995 realizations
plugin
FBI
MCMC
n = 100
n = 50
n = 25
d = 10