Validating Bayesian Inference Algorithms with Simulation-Based Calibration

Sean Talts, Michael Betancourt, Daniel Simpson, Aki Vehtari, Andrew Gelman

ISERP, Columbia University, New York. e-mail: sean.talts@gmail.com.

Symplectomorphic LLC., New York. e-mail: betanalpha@gmail.com.

Department of Statistical Sciences, University of Toronto. e-mail: simpson@utstat.toronto.edu.

Department of Computer Science, Aalto University, Finland. e-mail: Aki.Vehtari@aalto.fi.

Department of Statistics and Department of Political Science, Columbia University, New York. e-mail: gelman@stat.columbia.edu.

Abstract: Verifying the correctness of Bayesian computation is challenging. This is especially true for complex models that are common in practice, as these require sophisticated model implementations and algorithms. In this paper we introduce *simulation-based calibration* (SBC), a general procedure for validating inferences from Bayesian algorithms capable of generating posterior samples. This procedure not only identifies inaccurate computation and inconsistencies in model implementations but also provides graphical summaries that can indicate the nature of the problems that arise. We argue that SBC is a critical part of a robust Bayesian workflow, as well as being a useful tool for those developing computational algorithms and statistical software.

1. Introduction

Powerful algorithms and computational resources are facilitating Bayesian modeling in an increasing range of applications. Conceptually, constructing a Bayesian analysis is straightforward. We first define a joint distribution over the parameters, θ, and measurements, y, with the specification of a prior distribution and likelihood,

$$\pi(y, \theta) = \pi(y \mid \theta)\, \pi(\theta).$$

Conditioning this joint distribution on an observation, $\tilde{y}$, yields a posterior distribution,

$$\pi(\theta \mid \tilde{y}) \propto \pi(\tilde{y}, \theta),$$

that encodes information about the system being analyzed.

Implementing this Bayesian inference in practice, however, can be computationally challenging when applied to large and structured datasets. We must make our model rich enough to capture the relevant structure of the system being studied while simultaneously being able to accurately work with the resulting posterior distribution. Unfortunately, every algorithm in computational statistics requires that the posterior distribution possesses certain favorable properties in order to work as desired. Consequently the overall performance of an algorithm is sensitive to the details of the model and the observed data, and an algorithm that works well in one analysis can fail spectacularly in another.

As we move towards adapting our models to particular problems, we place the algorithms in our statistical toolbox under stress. Moreover, the complexity of these models provides abundant

opportunity for mistakes in their specification. We must verify both that our code is implementing the model we think it is and that our inference algorithm is able to perform the necessary computations accurately. While we always get some result from a given algorithm, we have no idea how good it is without some form of validation.

Fortunately, the structure of the Bayesian joint distribution allows for the validation of *any* Bayesian computational method capable of producing samples from the posterior distribution, or an approximation thereof. This includes not only Monte Carlo methods but also deterministic methods that yield approximate posterior distributions amenable to exact sampling, such as integrated nested Laplace approximation (INLA) (Rue, Martino and Chopin, 2009; Rue et al., 2017) and automatic differentiation variational inference (ADVI) (Kucukelbir et al., 2017). In this paper we introduce *simulation-based calibration* (SBC), as demonstrated with a corrected implementation of the ideas of Cook, Gelman and Rubin (2006) for validating these algorithms in a generic and straightforward way within the scope of a given Bayesian joint distribution.

We begin in Sections 2 and 3 with a discussion of the natural self-consistency of samples from the Bayesian joint distribution and previous validation methods that have exploited this behavior. Next, Section 4 introduces the simulation-based calibration framework and examines the qualitative interpretation of the SBC output, how it identifies how the algorithm being validated might be failing, and how it can be incorporated into a robust Bayesian workflow. Finally, Section 5 consider some useful extensions of SBC, and Section 6 demonstrates the application of the procedure over a range of analyses. All code for this paper is at `https://github.com/seantalts/simulation-based-calibration`.

2. Self-Consistency of the Bayesian Joint Distribution

The most straightforward way to validate a computed posterior distribution is to compare computed expectations with the exact values. An immediate problem with this, however, is that we know the true posterior expectation values for only the simplest models. These simple models, moreover, typically have a different structure to the models of interest in applications. This motivates us to construct a validation procedure that does not require access to the exact expectations or any other property of the true posterior distribution.

A popular alternative to comparing the computed and true expectation values directly is to define a ground truth $\tilde{\theta}$, simulate data from that ground truth, $\tilde{y} \sim \pi(y \mid \tilde{\theta})$, and then quantify how well the computed posterior recovers the ground truth in some way. Unfortunately this approach is flawed as demonstrated in a simple example.

Consider the model

$$
\begin{aligned}
y \mid \mu &\sim \mathrm{N}(\mu, 1^2) \\
\mu &\sim \mathrm{N}(0, 1^2)
\end{aligned}
$$

and an attempt at verification that uses the single ground truth value $\tilde{\mu} = 0$. If we simulate from this model and draw the plausible, but extreme, data value $\tilde{y} = 2.1$, then the true posterior will be $\mu \mid \tilde{y} \sim \mathrm{N}(1.05, 0.5^2)$. As $\tilde{\mu}$ is more than two posterior standard deviations from the posterior mean, we might be tempted to say that recovery has not been successful. On the other hand, imagine that we accidentally used code that exactly fits an identical model but with the variance for both the likelihood and prior set to 10 instead of 1. In this case, the incorrectly computed posterior would be $\mathrm{N}(1.05, 5^2)$ and we might conclude that the code correctly recovered the posterior.

Consequently, the behavior of the algorithm in any *individual* simulation will not characterize the ability of the inference algorithm to fit that particular model in any meaningful way. In the example above, it might lead us to conclude that the incorrectly coded analysis worked as desired, while the correctly coded analysis failed. In order to properly characterize an analysis we need to at the very least consider multiple ground truths.

Which ground truths, however, should we consider? An algorithm might be able to recover a posterior constructed from data generated from some parts of the parameter space while faring poorly on data generated from other parts. In Bayesian inference a proper prior distribution quantifies exactly which parameter values are relevant and hence should be considered when evaluating an analysis. This immediately suggests that we consider the performance of an algorithm over the entire Bayesian joint distribution, first sampling a ground truth from the prior, $\tilde{\theta} \sim \pi(\theta)$, and then data from the corresponding data generating process, $\tilde{y} \sim \pi(y \mid \tilde{\theta})$. We can then build inferences for each simulated observation $\tilde{y}$ and then compare the recovered posterior distribution to the sampled parameter $\tilde{\theta}$.

Advantageously, this procedure also defines a natural condition for quantifying the faithfulness of the computed posterior distributions, regardless of the structure of the model itself. Integrating the exact posteriors over the Bayesian joint distribution returns the prior distribution,

$$\pi(\theta) = \int \mathrm{d}\tilde{y}\, \mathrm{d}\tilde{\theta}\, \pi(\theta \mid \tilde{y})\, \pi(\tilde{y} \mid \tilde{\theta})\, \pi(\tilde{\theta}). \tag{1}$$

In other words, for *any* model, the average of any exact posterior expectation with respect to data generated from the Bayesian joint distribution reduces to the corresponding prior expectation.

Consequently, any discrepancy between the *data averaged posterior* (the right-hand side of (1)) and the prior distribution indicates some error in the Bayesian analysis or computation. Well-defined comparisons of the prior distribution to the data-averaged posterior distribution provides a generic means of validating the analysis, at least within the scope of the modeling assumptions.

3. Existing Validation Methods Exploiting the Bayesian Joint Distribution

The self-consistency of the data-averaged posterior and the prior in (1) is not a novel observation. This behavior has been exploited in at least two earlier methods for validating Bayesian computational algorithms.

Geweke (2004) proposed a Gibbs sampler targeting the Bayesian joint distribution that alternatively samples from the posterior, $\pi(\theta \mid y)$, and the likelihood, $\pi(y \mid \theta)$. If an algorithm can generate accurate posterior samples, then this Gibbs sampler will produce accurate samples from the Bayesian joint distribution, and the marginal parameter samples will be indistinguishable from any sample of the prior distribution. The author recommended quantifying the consistency of the marginal parameter samples and a prior sample with z-scores of each parameter mean, with large z-scores indicating a failure of the algorithm to produce accurate posterior samples.

The main challenge with this method is that the diagnostic z-scores will be meaningful only once the Gibbs sampler has converged. Unfortunately, the data and the parameters will be strongly correlated in a generative model and the convergence of this Gibbs sampler will be slow, making it challenging to identify when the diagnostics can be considered.

Cook, Gelman and Rubin (2006) avoided the auxiliary Gibbs sampler entirely by considering cumulative distribution function (CDF) values (quantiles) approximated using samples from the simulated posterior distribution. They use the notation θ to represent any scalar model parameter

or function of parameters. They noted that if $\tilde{\theta} \sim \pi(\theta)$ and $\tilde{y} \sim \pi(y \mid \tilde{\theta})$ then the exact posterior CDF values for each parameter,

$$q(\tilde{\theta}) = \int \mathrm{d}\theta \, \pi(\theta \mid \tilde{y}) \, \mathbb{I}[\theta < \tilde{\theta}],$$

will be uniformly distributed provided that the posteriors are absolutely continuous. Consequently any deviation from the uniformity of the computed posterior CDF values indicates a failure in the implementation of the analysis.

The authors suggest quantifying the uniformity of these CDF values by transforming them into z-scores with an application of the inverse normal CDF. The absolute values of the z-scores can then be visualized to identify deviations from normality and hence uniformity of the CDF values. At the same time these deviations can be summarized with a χ^2 statistic.

This procedure works well in certain examples, as demonstrated by Cook, Gelman and Rubin (2006), but it can run into problems with MCMC samples as the empirical CDF values only asymptotically approach the true values. This makes it difficult to determine whether a deviation from normality is due to pre-asymptotic behavior or error in the posterior computations. In addition the description of the algorithm in Cook, Gelman and Rubin (2006) is incomplete in that it neglected to mention the continuity correction used for its quantile computation, as implemented in Cook (2006).

In particular, because there are only $L + 1$ positions in a posterior sample of size L in between which the prior sample $\tilde{\theta}$ can fall, an empirically approximated CDF value of the prior draw $\tilde{\theta}$ within the posterior sample $(\theta_1, \ldots, \theta_L)$,

$$q = \frac{1}{L} \sum_{l=1}^{L} \mathbb{I}[\theta_l < \tilde{\theta}],$$

is fundamentally discrete, taking one of $L + 1$ evenly spaced values on $[0, 1]$. This discretization causes artifacts when visualizing the CDF values and it requires some continuity corrections for the finite instances where the estimated CDF value equals 0 or 1. At the same time, autocorrelation in the simulations modifies the distributions of test statistics that were worked out implicitly assuming independence, a point recognized in the recent correction (Gelman, 2017). With attempts at smoothing, we may fix visual artifacts but we have found no exact proofs of distribution for these continuous estimators.

4. Simulation-Based Calibration

We can work around the discretization artifacts of Cook, Gelman and Rubin (2006) by considering a similar consistency criterion that is immediately compatible with sampling-based algorithms. In this section we introduce *simulation-based calibration* (SBC) based on comparing histograms of rank statistics to the discrete uniform distribution that would arise if the analysis has been correctly implemented.

SBC requires just one assumption: that we have a generative model for our data. Given such a model, we can run any given algorithm over many simulated observations and the self-consistency condition (1) provides a target to verify that the algorithm is accurate over that ensemble, and hence sufficiently *calibrated* for the assumed model. This calibration ensures that certain one dimensional

test statistics are correctly distributed under the assumed model and is similar to checking the coverage of a credible interval under the assumed model.

Importantly, this calibration is limited exclusively to the computational aspect of our analysis. It offers no guarantee that the posterior will cover the ground truth for any single observation or that the model will be rich enough to capture the truth at all. Understanding the range of posterior behaviors for a given observation requires a more careful *sensitivity analysis* while validating the model assumptions themselves requires a study of *predictive performance*, such as posterior predictive checks (PPCs, e.g., Gelman et al. (2013), chapter 6). Where SBC uses samples from the joint prior distribution $\pi(\theta, y)$, PPCs use the posterior predictive distribution for predicting new data $\tilde{y}$, $\pi(\tilde{y}|y)$. We view both of these checks as a vital part of a robust Bayesian workflow.

In this section we first demonstrate the expected behavior of rank statistics under a proper analysis and construct the SBC procedure to exploit this behavior. We then demonstrate how deviations from the expected behavior are interpretable and help identify the exact nature of implementation error.

4.1. Validating Consistency With Rank Statistics

Consider the sequence of samples from the Bayesian joint distribution and resulting posteriors,

$$\begin{aligned} \tilde{\theta} &\sim \pi(\theta) \\ \tilde{y} &\sim \pi(y \mid \tilde{\theta}) \\ \{\theta_1, \ldots, \theta_L\} &\sim^{\text{iid}} \pi(\theta \mid \tilde{y}). \end{aligned} \tag{2}$$

The relationship (1) implies that the prior sample, $\tilde{\theta}$, and an exact posterior sample, $\{\theta_1, \ldots, \theta_L\}$, will be distributed according to the the same distribution. We will study coherence of parameters or unidimensional summaries one at a time. For any one-dimensional function of parameters, $h : \Theta \to \mathbb{R}$, the *rank statistic* of the prior sample relative to the posterior sample,

$$r(\{h(\theta_1), \ldots, h(\theta_L)\}, h(\tilde{\theta})) = \sum_{l=1}^{L} \mathbb{I}[h(\theta_l) < h(\tilde{\theta})], \tag{3}$$

will be uniformly distributed across the integers $\{0, 1, \ldots, L\}$.

Theorem 1. *Let $\tilde{\theta} \sim \pi(\theta)$, $\tilde{y} \sim \pi(y \mid \tilde{\theta})$, and $\{\theta_1, \ldots, \theta_L\} \sim \pi(\theta \mid \tilde{y})$ for any joint distribution $\pi(y, \theta)$. The rank statistic of any one-dimensional random variable over θ is uniformly distributed over the integers $\{0, 1, \ldots, L\}$.*

The proof is given in Appendix B.

There are many ways of testing the uniformity of the rank statistics, but the SBC procedure, outlined in Algorithm 1, exploits a histogram of rank statistics for a given random variable to enable visual inspection of uniformity (Figure 1). We first draw a sample of size N from the Bayesian joint distribution. For each replicated generated dataset we then sample L exact samples from the posterior distribution and compute the corresponding rank statistic. We then bin the L rank statistics in a histogram spanning the $L+1$ possible values, $\{0, \ldots, L\}$. If only correlated posteriors samples can be drawn then the procedure can be modified as discussed in Section 5.1.

In order to help identify deviations, each histogram is complemented with a gray band indicating 99% of the variation expected from a uniform histogram. Formally, the vertical extent of the band

Algorithm 1 SBC generates a histogram from an ensemble of rank statistics of prior samples relative to corresponding posterior samples. Any deviation from uniformity of this histogram indicates that the posterior samples are inconsistent with the prior samples. For a multidimensional problem the procedure is repeated for each parameter or quantity of interest to give multiple histograms.

Initialize a histogram with bins centered around $0, \ldots, L$.

for n in N **do**
  Draw a prior sample, $\tilde{\theta} \sim \pi(\theta)$
  Draw a simulated data set, $\tilde{y} \sim \pi(y \mid \tilde{\theta})$
  Draw posterior samples $\{\theta_1, \ldots, \theta_L\} \sim \pi(\theta \mid \tilde{y})$
  for each scalar summary h **do**
    Compute the rank statistic $r(\{h(\theta_1), \ldots, h(\theta_L)\}, h(\tilde{\theta}))$ as defined in (3)
    Increment the histogram with $r(\{h(\theta_1), \ldots, h(\theta_L)\}, \mid h(\tilde{\theta}))$

Analyze the histogram for uniformity.

extends from the 0.005 to the 0.995 quantiles of the binomial$(N, (L+1)^{-1})$ distribution so that under uniformity we expect that, on average, the counts in only one bin in a hundred will deviate outside this band.

In complex problems computational resources often limit the number of replications, N, and hence the sensitivity of the resulting SBC histogram. In order to reduce the noise from small replications it can be beneficial to uniformly bin the histogram, for example by pairing neighboring ranks together into a single bin to give $B = L/2$ total bins. Our experiments have shown that keeping $N/B \approx 20$ lead to a good tradeoff between the expressiveness of the binned histogram and the necessary variance reduction. Choosing $L + 1$ to be divisible by a large power of 2 makes this re-binning easier; for example, instead of generating a sample of size 1000 in a problem with known computational limitations, one could generate a sample of size $1024 - 1 = 1023$ from the posterior distributions.

Regardless of the binning, however, it will be difficult to identify sufficiently small deviations in the SBC histogram and it can be useful to consider alternative visualizations of the rank statistics. We consider this in Section 5.2.

4.2. Interpreting SBC

What makes the SBC procedure particularly useful is that the deviations from uniformity in the SBC histogram can indicate *how* the computed posteriors are incorrect. We follow an observation from the forecast calibration literature (Anderson, 1996; Hamill, 2001), which suggests that the way the rank histogram deviates from uniformity can indicate bias or mis-calibration of the computed posterior distributions.

An illustrative histogram without any appreciable deviations is shown in Figure 1. The histogram of rank statistics is consistent with the expected uniform behavior, here shown with the 99% interval in light gray and the median in dark gray.

Figure 2 illustrated the deviation from uniformity exhibited by correlated posterior samples. The correlation between the posterior samples causes them to cluster relative to the proceeding prior sample, biasing the ranks to extremely small or large values. We describe how to process correlated posterior samples generated from Markov chain Monte Carlo algorithms in Section 5.1.

Next, consider a computational algorithm that produces, on average, posteriors that are *under-dispersed* relative to the true posterior. When averaged over the Bayesian joint distribution this

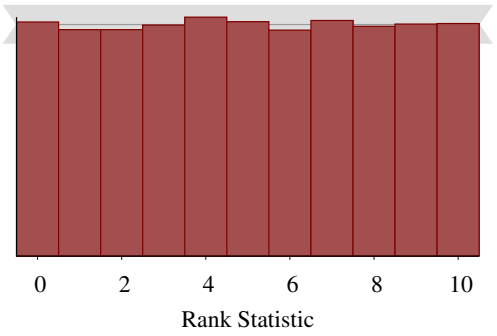


FIG 1. *Uniformly distributed rank statistics are consistent with the ranks being computed from independent samples from the exact posterior of a correctly specified model.*

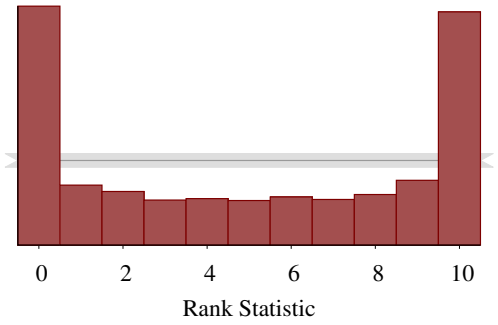
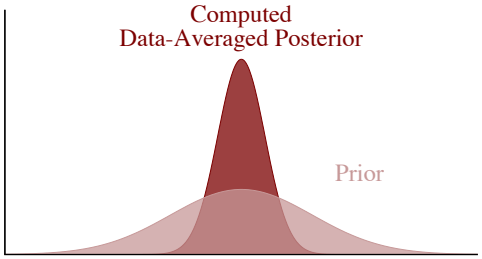
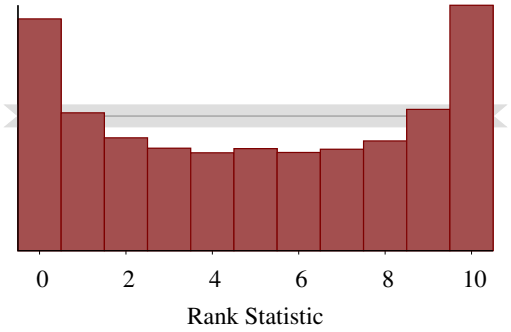


FIG 2. *The spikes at the boundaries of the SBC histogram indicate that the procedure is not creating independent samples from the posterior distribution consistent with the prior simulations.*

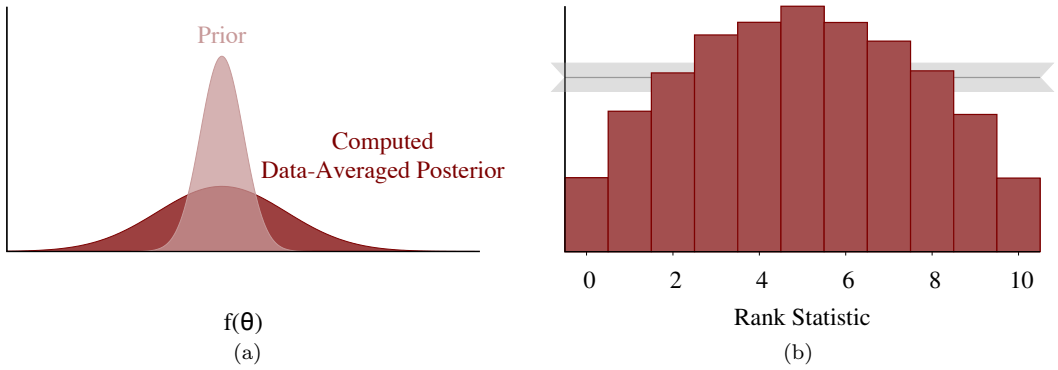


(a)



(b)

FIG 3. *A symmetric ∪ shape indicates that the computed data-averaged posterior distribution (dark red) is underdispersed relative to the prior distribution (light red). This implies that on average the computed posterior will be narrower than the true posterior.*



(a) (b)

FIG 4. *A symmetric, ∩-shaped distribution indicates that the computed data-averaged posterior distribution (dark red) is overdispersed relative to the prior distribution (light red). This implies that on average the computed posterior will be wider than the true posterior.*

results in a data-averaged posterior distribution (1) that is underdispersed relative to the prior distribution (Figure 3a), and hence rank statistics that are biased towards the middle that manifests as a ∪-shaped histogram (Figure 3b).

Conversely, an algorithm that computes posteriors that are, on average, *overdispersed* relative to the true posterior produces a histogram of rank statistics with a ∩ shape (Figure 4). For example, this is commonly assumed to happen when using variational inference with Kullback-Leibler divergence computed from the true distribution to the approximate (mass covering behavior), or can happen with a bug in MCMC algorithm implementation.

Finally, we might have an algorithm that produces posteriors that are biased above or below the true posterior. For example, this can happen when MCMC algorithm has difficulties reaching narrow parts of the posterior, or when normal approximation is used for skewed posterior. This bias results in a data-averaged posterior distribution biased in the same direction relative to the prior distribution (Figure 5a) and rank statistics that are biased in the opposite direction (Figure 5b). For example, posterior samples biased to smaller values results in higher rank statistics, whereas posterior samples biased to larger values results in lower rank statistics.

A misbehaving analysis can in general manifest many of these deviations at once. Because each deviation is relatively distinct from the others, however, in practice the systematic deviations are readily separated into the different behaviors if they are large enough.

4.3. Simulation-Based Calibration Plays a Vital Role in a Robust Bayesian Workflow

SBC is one of the few tools for evaluating the critical but frequently unexamined choice of computational method made in any Bayesian analysis. We have already argued that performance on a single simulated observation is, at best, a blunt instrument. Moreover, while most theoretical results only provide asymptotic comfort, SBC adapts to the specific model design under consideration.

Furthermore, because SBC validates accuracy through one-dimensional random variables, it can make sense to use carefully chosen random variables to make targeted assessments of an analysis

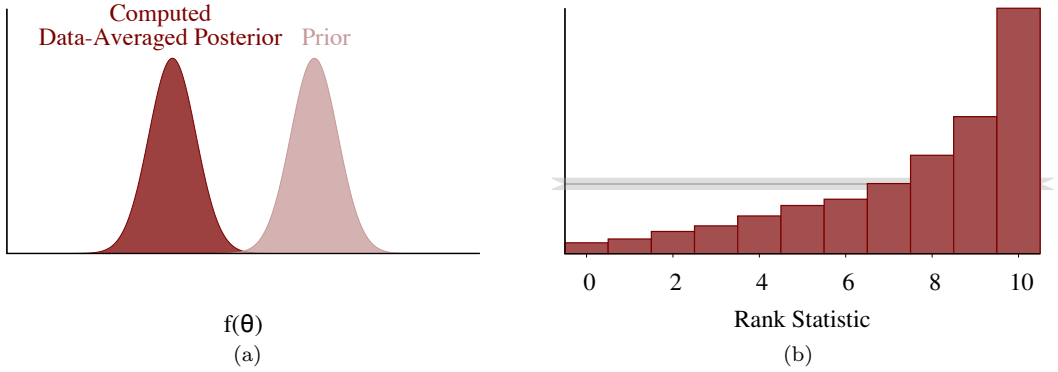


FIG 5. *Asymmetry in the rank histogram indicates that the computed data-averaged posterior distribution (dark red) will be biased in the opposite direction relative to the prior distribution (light red). This implies that on average the computed posterior will be biased in the same opposite direction.*

based on our inferential needs and priorities. For example, in a model for election forecasting such as Heidemanns, Gelman and Morris (2020), it would make sense to look at coverage for predictions of the Electoral College total and key swing states. As needs and priorities change we can run SBC again using the same posterior samples to verify the analysis anew.

The downside of using SBC in practice is that it is expensive; instead of fitting a single observation we have to fit N simulated observations before even considering the measured data. These fits, however, are embarrassingly parallel, which makes it possible to leverage access to computational resources through multicore personal computers, computing clusters, and cloud computing. For example, all of the examples in Section 6 were run on clusters and took, at most, a few hours.

The procedure can be sped up further by reducing the number of independent samples, producing a cruder test that can still catch gross problems in an analysis.

5. Extending Simulation-Based Calibration

SBC provides a straightforward procedure for validating simulation-based algorithms applied to Bayesian analyses, but the procedure can be limited in a few circumstances. In this section we discuss some small modifications that allow SBC to remain useful in some common practical circumstances.

5.1. Mitigating the Effect of Autocorrelation

The confidence bands we have shown in histogram plots (and also the confidence bands later shown in ECDF difference plots) are based on assumption of independence. SBC histograms can become lumpy if the posterior samples are dependent, making the confidence bands uncalibrated (Säilynoja, Bürkner and Vehtari, 2021) which makes it difficult to assess bias in the samples. As the mixing efficiency of Markov chains can vary in different parts of the parameter space (Vehtari et al., 2021), the amount of lumpiness in different parts of histogram can also vary. Thus there is no general characteristic shape of the histogram due to the dependencies (although, for example, slow mixing

in tails can increase lumpiness specifically in edges of the histograms, increasing the probability of accidental ∪-shape). As taking into account the full dependency would be very challenging, we instead thin the Markov chains to obtain samples with negligible dependence.

The amount of dependency over the chain is commonly measured using autocorrelation time estimates and corresponding effective sample size (ESS) estimates (see, e.g., Vehtari et al., 2021). When autocorrelation time is negligible, ESS is close to the nominal sample size. The autocorrelation time and ESS depend on h. As the mixing efficiency can be different in different parts of the parameter space, and the rank statistic is closely related to cumulative distribution function $P(h(\theta) \leq h^*)$, we suggest computing minimum ESS with h^* being empirical quantiles of $h(\theta)$ (e.g.,19 equispaced quantiles; see Vehtari et al. (2021) for examples). Ranks are MCMC estimators of probabilities. Autocorrelation in the Markov chains induces varying and correlated errors in those probability estimators which invalidate confidence bands that assume uncorrelated samples. We could keep thinning until ESS is close to the thinned nominal sample size. In practice, we have observed that when ESS is the minimum ESS over the empirical quantiles, thinning by $\lceil L/\text{ESS} \rceil$ reduces the autocorrelation sufficiently to produce samples that are well suited for the SBC. When running the SBC procedure over multiple quantities of interest we suggest thinning the chain just once using the largest thinning value determined with the above procedure over all quantities. This approach works also for antithetic Markov chains (not rare with dynamic Hamiltonian Monte Carlo) which have negative autocorrelations for even lags. Antithetic chains may have the effective sample size larger than the nominal size for some quantiles, but not for all.

Algorithm 2 Simulation-based calibration can be applied to the correlated posterior samples generated by a Markov chain provided that the Markov chain can be thinned to L near-independent samples at each iteration.

Initialize a histogram with bins centered around $0, \ldots, L$.

for n in N **do**
  draw a prior sample $\tilde{\theta} \sim \pi(\theta)$
  draw a simulated data set $\tilde{y} \sim \pi(y \mid \tilde{\theta})$
  run a Markov chain for L' iterations to generate the correlated posterior samples,
    $\{\theta_1, \ldots, \theta_{L'}\} \sim \pi(\theta \mid \tilde{y})$
  compute the effective sample size, ESS$[h]$ of $\{\theta_1, \ldots, \theta_{L'}\}$ for the one-dimensional summary h
  if ESS$[h] < 0.95L$ **then**
    rerun the Markov for $L' \cdot \lceil L/\text{ESS}[h] \rceil$ iterations
  uniformly thin the correlated sample to L states and truncate any leftover samples at L
  compute the rank statistic $r(\{h(\theta_1), \ldots, h(\theta_L)\}, h(\tilde{\theta}))$ as defined in (3)
  increment the histogram with $r(\{h(\theta_1), \ldots, h(\theta_L)\}, h(\tilde{\theta}))$

Analyze the histogram for uniformity.

Although some autocorrelation will remain in a sample that has been thinned by Algorithm 2, if $0.95L < \text{ESS}[h] < 1.05L$, the effect of the dependency in assessing uniformity is small (Säilynoja, Bürkner and Vehtari, 2021). If desired, more conservative thinning strategies such as the truncation rules of Geyer (1992) can remove autocorrelation completely from the sample. A sample thinned with these rules is typically much smaller than the sample achieved by thinning based on the effective sample size, and we have not seen any significant benefit for SBC from the increased computation time needed for these more elaborate thinning methods to date.

Deviations that cannot be mitigated by thinning provide strong evidence that the Markov chain Monte Carlo estimators do not follow a central limit theorem or follow it with a very slow conver-

gence rate so that the Markov chains are not adequately exploring the target parameter space. This is particularly useful given that establishing central limit theorems for particular Markov chains and particular target distributions is a notoriously challenging problem even in relatively simple circumstances.

5.2. Simulation-Based Calibration for Small Deviations

The SBC histogram provides a general and interpretable means of identifying deviations from uniformity of the rank statistics and hence inaccuracies in our posterior computation, at least when the inaccuracies are large enough. For small deviations, however, the SBC histogram may not be sensitive enough for the deviations to be evident and other visualization strategies may be advantageous.

One option is to bin the SBC histogram multiple times to see if any deviation persists regardless of the binning. This approach, however, is ungainly to implement when there are many parameters and can be difficult to interpret. In particular, considering multiple histograms introduces a vulnerability to multiple testing biases.

Another approach is to pair the SBC histogram with the empirical cumulative distribution function (ECDF) which reduces variation at small and large ranks, making it easier to identify deviations around those values (Figure 6b). The deviation of the empirical CDF away from the expected uniform behavior is especially useful for identifying these small deviations (Figure 6c). Säilynoja, Bürkner and Vehtari (2021) present an efficient way to compute simultaneous confidence bands for ECDF difference plots.

More subtle deviations can be isolated by considering more particular summary statistics, such as ranks quantiles or averages. While these have the potential to identify small biases they can also be harder to interpret and not as sensitive to the systematic deviations that manifest in the SBC histogram. Identifying a robust suite of diagnostic statistics is an open area of research and at present we recommend using the SBC histogram whenever possible.

6. Experiments

In this section we consider the application of SBC on a series of examples that demonstrates the utility of the procedure for identifying and correcting incorrectly implemented analyses. For each example we implement the SBC procedure using a posterior sample of size $L = 100$ so that, if the algorithm is properly calibrated, then the rank statistics will follow a discrete uniform distribution on $\{0, 1, \ldots, 100\}$. The experiments in Section 6.1 through Section 6.3 used $N = 10,000$ replicated observations while the experiment in Section 6.4 used $N = 1000$ replicated observations.

6.1. Misspecified Prior

Let's first consider the case where we build our posterior using a different prior than that which we use to generate prior samples. This is not an uncommon mistake, even when models are specified in probabilistic programming languages.

Consider the linear regression model, $y_i \sim \mathrm{N}(\alpha + \beta x_i, 1.2^2)$, $i = 1, \ldots, n$ (Listing 2 in the Appendix) but with the prior on β modified to $\mathrm{N}(0, 1^2)$. With the prior samples still drawn according to $\mathrm{N}(0, 10^2)$, we expect that the posterior for β will be underdispersed relative to the prior even when

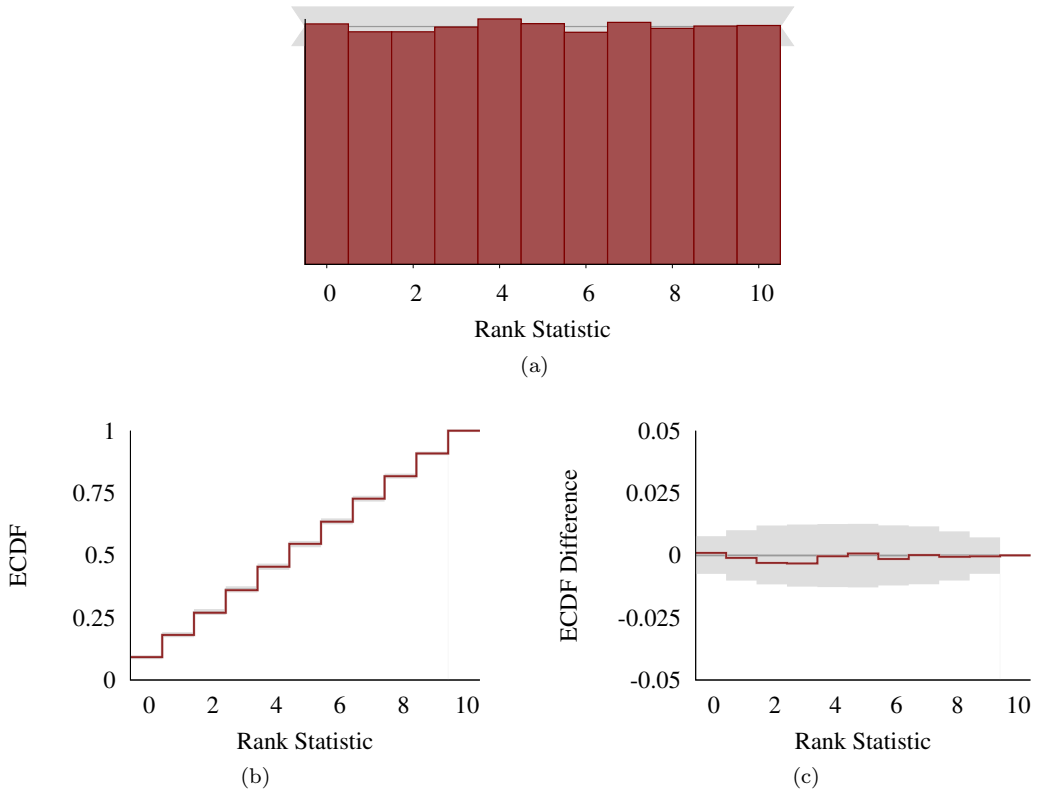


FIG 6. *In order to emphasize small deviations at low and large ranks we can pair the (a) SBC histogram with the corresponding (b) empirical cumulative distribution function (dark red) along with the variation expected of the empirical cumulative distribution function under uniformity. (c) Deviations are often easier to identify by subtracting the expected uniform behavior from the empirical cumulative distribution function.*

the computation is exact. This should then lead to the deviation demonstrated in Figure 3 and, indeed, we see a ∪ shape in the SBC histogram (using Algorithm 2 to account for autocorrelation in the sample for β (Figure 7).

6.2. Biased Markov chain Monte Carlo

Hierarchical models implemented with a centered parameterization (Papaspiliopoulos, Roberts and Sköld, 2007) are known to exhibit a challenging geometry that can cause MCMC algorithms to return biased posterior samples. While some algorithms, such as Hamiltonian Monte Carlo (Neal et al., 2011; Betancourt and Girolami, 2013), allow for diagnostics capable of identifying this problem, these diagnostics are not available for general MCMC algorithms. Consequently the SBC procedure should be particularly useful in hierarchical models if it can identify this problem.

Here we consider a hierarchical model of the eight schools problem (Rubin, 1981; Gelman et al., 2013) using a centered parameterization (Listing 3 in the Appendix): $y_j \sim \mathrm{N}(\theta_j, \sigma_j^2)$, $\theta_i \sim$

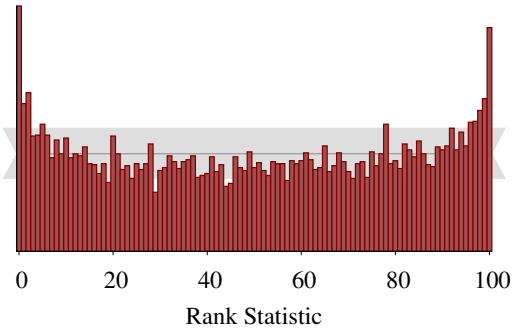


FIG 7. *When the data are simulated using a much wider prior than was used to fit the model, the SBC histogram for a regression parameter β exhibits a $\cup$ shape.*

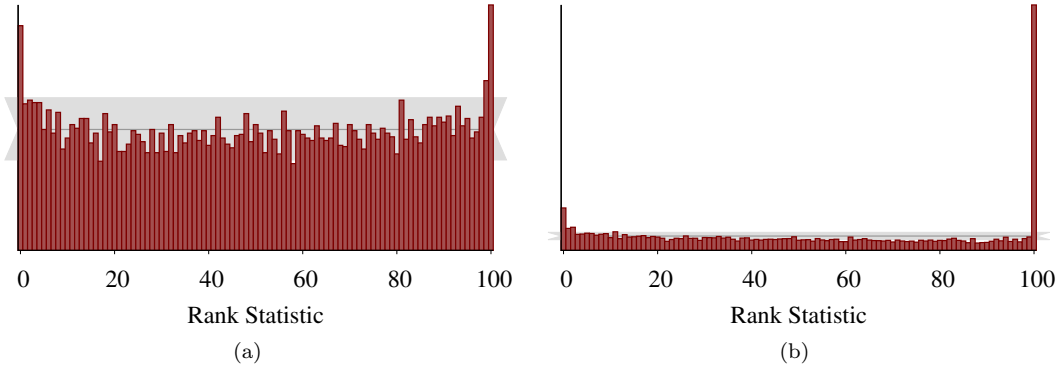


FIG 8. *Histograms show SBC for (a) $\theta[1]$ and (b) τ in the 8 schools centered parameterization of Section 6.2, after thinning the MCMC sample. These plots demonstrate problems with this Hamiltonian Monte Carlo implementation.*

$\mathrm{N}(\mu, \tau^2)$, $j = 1, \ldots, J$, with the vectors y and σ known. In this example, the centered parameterization exhibits a classic funnel shape that contracts into a region of strong curvature around small values of τ, making it difficult for most Markov chain methods to adequately explore.

As shown in Figure 8, the SBC rank histogram for τ produced from Algorithm 1 clearly demonstrates that the posterior samples from Stan's dynamic Hamiltonian Monte Carlo extension of the NUTS algorithm (Hoffman and Gelman, 2014; Betancourt, 2017) are biased below the prior samples, consistent with the known pathology (Figure 9b). Here we used Algorithm 1 instead of 2 because the algorithm's unfaithfulness is evident over the deviation caused by the autocorrelation. Moreover, the extra computation required to return an effective sample size of $L = 100$ post-thinning is impractical here as the centered parameterization, among other failing HMC diagnostics, has a low effective sample size per sample rate.

The corresponding non-centered parameterization, $y_i \sim \mathrm{N}(\theta_j, \sigma_j^2)$, $\theta_j = \mu + \tau\tilde{\theta}_j$, $\tilde{\theta}_j \sim \mathrm{N}(0, 1)$, $j = 1, \ldots, J$, should behave much better: the mathematical model is unchanged, but when the data are consistent with values of τ near zero (as in the eight schools example), the geometry is such that

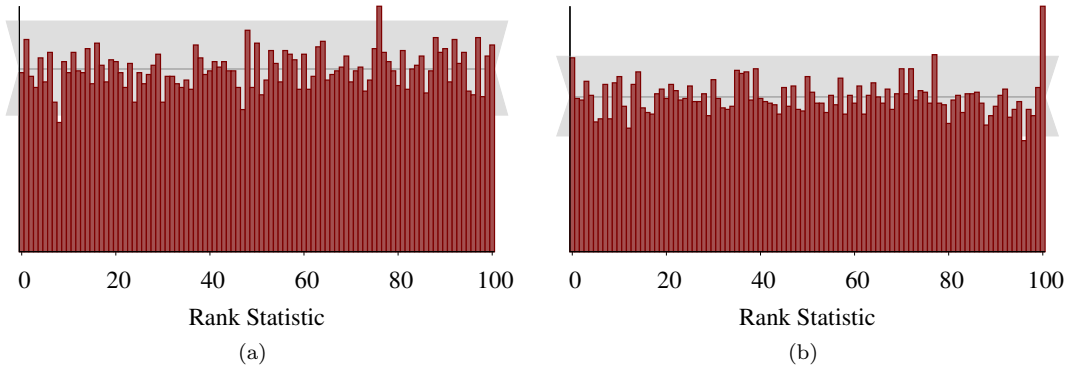


FIG 9. *Once thinned (a), the SBC histogram for τ from the 8 schools non-centered parameterization in Section 6.2 show no evidence of bias. Without thinning, the SBC histogram for τ in the same model, (b), exhibits characteristic signs of autocorrelation in the posterior sample.*

sampling is much more efficient on the space of $(\tilde{\theta}, \mu, \tau)$ than on the original space of (θ, μ, τ). And, indeed, the SBC histogram thinned using Algorithm 2 (Figure 9) shows no deviation from uniformity as we expected given that Hamiltonian Monte Carlo is known to yield accurate computation for this analysis. If the SBC histogram is computed without thinning (Figure 9), the autocorrelation manifests as a large spikes at $L = 100$, consistent with the discussion in Section 5.1.

6.3. ADVI can fail for simple models

We next consider automatic differentiation variational inference (ADVI) applied to a simple linear regression model (Listings 1 and 2 in the Appendix). In particular, we run the implementation of ADVI in Stan 2.17.1 that returns exact samples from a variational approximation to the posterior. Here we use Algorithm 1 again because we know that ADVI does not produce autocorrelated posterior samples.

As shown in Figure 10, SBC immediately reveals that the variational approximation found by ADVI drastically underestimates the posterior for the slope, β. Compare this with the results from Hamiltonian Monte Carlo (Figure 11), which yields a rank histogram consistent with uniformity.

6.4. INLA is slightly biased for spatial disease prevalence mapping

Finally let's consider a sophisticated spatial model for HIV prevalence fit to data from the 2003 Demographic Health Survey in Kenya (Corsi et al., 2012). We follow the experimental setup of Wakefield, Simpson and Godwin (2016) and fit the model using INLA.

The data were collected by dividing Kenya into 400 enumeration areas and in the ith area randomly sampling m_i households, with the jth household containing N_{ij} people. Both m_i and N_{ij} are chosen to be consistent with the Kenya DHS 2003 AIDS recode. The number of positive

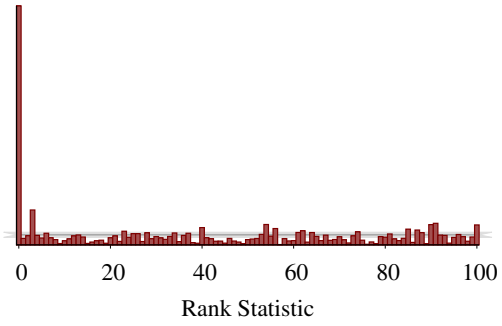


FIG 10. *The SBC histogram resulting from applying ADVI on the simple linear regression model indicates that the algorithm is strongly biased towards larger values of β in the true posterior.*

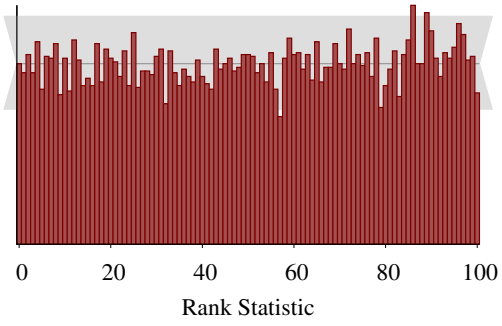


FIG 11. *SBC Algorithm 2 applied to a linear regression analysis indicates no issues as the empirical rank statistics (red) are consistent with the variation expected of a uniform histogram (gray).*

responses y_{ij} is modeled as

$$\begin{aligned} y_{ij} &\sim \text{binomial}(N_{ij}, p_{ij}) \\ p_{ij} &= \text{logit}^{-1}(\beta_0 + S(x_i) + \epsilon_{ij}), \end{aligned}$$

where $S(\cdot)$ is a Gaussian process, x_i is the centroid of the ith area, and ϵ_{ij} are iid Gaussian error terms with standard deviation τ. Following the computation reasoning of Wakefield, Simpson and Godwin (2016) we approximate $S(\cdot)$ using the stochastic partial differential equation approximation (Lindgren, Rue and Lindström, 2011) to a Gaussian process with isotropic covariance function

$$c(x_1, x_2; \rho, \sigma) = \frac{\sqrt{8}\sigma^2}{\rho} \|x_1 - x_2\| K_1\left(\frac{\sqrt{8}}{\rho}\|x_1 - x_2\|\right),$$

where ρ is the distance at which the spatial correlation between points is approximately 0.1, σ is the pointwise standard deviation, and $K_1(\cdot)$ is a modified Bessel function of the second kind.

To complete the model, we must specify priors on β_0, ρ, σ, and τ. We specify a $\text{N}(-2.5, 1.5^2)$ prior on the logit baseline prevalence β_0. This prior is based on the national HIV prevalence across the world ranges from 0.3% to 20% (Central Intelligence Agency, 2018). We use penalized complexity priors (Simpson et al., 2017; Fuglstad et al., 2019) on the remaining parameters tuned to ensure $\Pr(\rho < 0.1) = \Pr(\sigma > 1) = \Pr(\tau > 1) = 0.1$.

One of the quantities of interest for this model is the average prevalence over a subregion A of Kenya,

$$\frac{1}{|A|}\int_A \text{logit}^{-1}(\beta_0 + S(x))\, dx.$$

Wakefield, Simpson and Godwin (2016) suggested fitting this model using the `R-INLA` package to speed up the computation. As the quantity of interest is a nonlinear transformation of a number of parameters, we need to use `R-INLA`'s approximate posterior sampler, which is a relatively recent feature (Seppä et al., 2019).

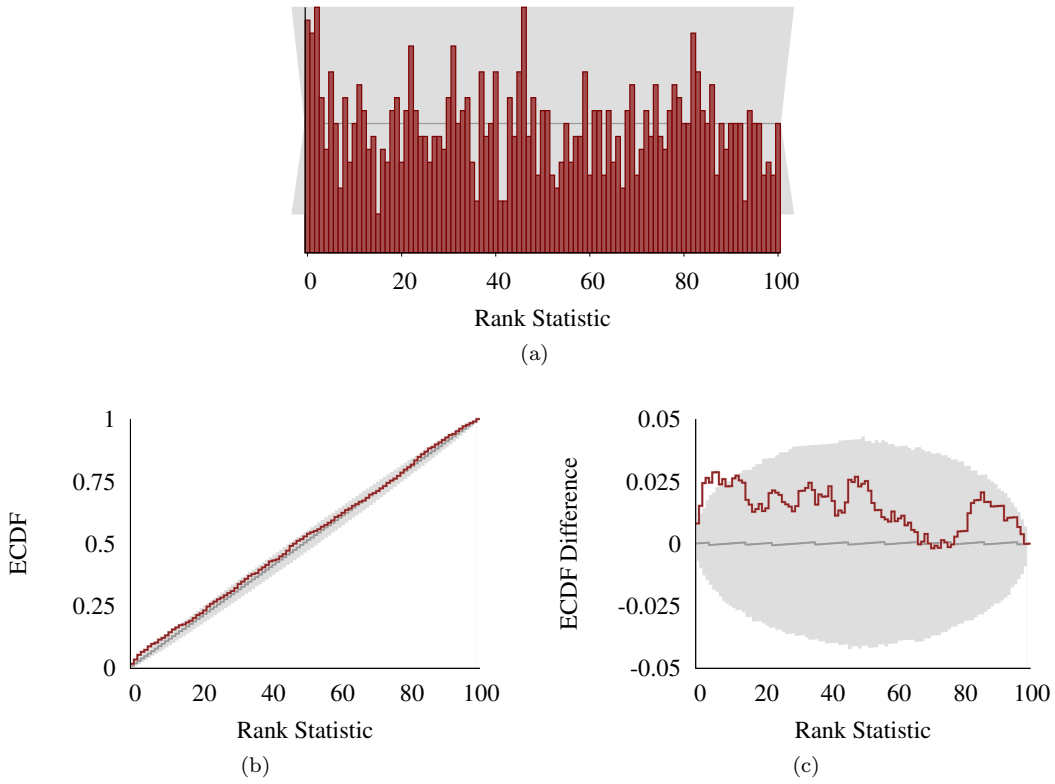


FIG 12. *(a) The SBC histogram for the average prevalence of a spatial model doesn't exhibit any obvious deviations, although the large span of the expected variation (gray) suggests that this test maybe too noisy to capture some potentially important discrepancies. (b) The empirical cumulative distribution function (dark red), however, shows that there is a small deviation at low ranks beyond the variation expected from a uniform distribution (gray). (c) The deviation is more evident by looking at the difference between the empirical cumulative distribution function and the stepwise-linear behavior expected of a discrete uniform distribution.*

Figure 12a shows the SBC histogram for $N = 1000$ replications. The histogram shows that all of the ranks fall within the gray bars, but the large span of the bars indicates that the visual diagnostic may be too noisy to capture some potentially important discrepancies. In our tests, we saw that it's common for deviations from a uniform distribution to be sufficiently severe that this histogram will still exhibit the signs of a poorly fitting procedure. Hence for a more fine-scale view of the fit we follow the recommendation in Section 5.2 and consider the ECDF (Figure 12b, c). Here we see that low ranks are seen slightly more often in the computed ranks than we would expect from a uniform distribution.

It is not surprising that INLA exhibits some bias in this example. Binomial data with low expected counts does not contain much information, which poses some problems for the Laplace approximation. Even though this feature is only present when the observed values of y_{ij}/N_{ij} are close to zero, the SBC procedure is a sufficiently sensitive instrument to identify the problem. Overall, we would view INLA as a good approximation in a country like Kenya where the national prevalence is around 5.4%, while it would be inappropriate in Australia where the prevalence is 0.1% (Central Intelligence Agency, 2018). If we repeated this type of survey in a country with only 0.1% prevalence, however, then we would end up with too many zero observations for the method to be useful.

7. Discussion

In this paper, we introduce simulation-based calibration (SBC), a readily-implemented procedure that can identify sources of poorly implemented analyses, including biased computational algorithms or incorrect model specifications. The visualizations produced by the procedure allow us to not only identify that a problem exists but also to learn how the problem will affect resulting inferences. The ability to both identify and interpret these issues makes SBC an important step in a robust Bayesian workflow.

Our reliance on interpreting the SBC diagnostic through visualization, however, can be a limitation in practice, especially when dealing with models featuring a large number of parameters. One immediate direction for future work is to develop reliable numerical summaries that quantify deviations from uniformity of each SBC histogram and provide automated diagnostics that can flag certain parameters for closer inspection.

The SBC histograms are only able to assess the calibration of one-dimensional posterior summaries, although this problem can somewhat be addressed by using summaries such as $h(\theta) = \theta_1\theta_2$. Alternatively, the fraction of posterior samples in a multidimensional subset A can be compared with the prior probability $\pi(A)$. This is just using an indicator function as the SBC random variable. An interesting extension of these ideas would be to incorporate some of the advances in multivariate calibration of probabilistic forecasts (Gneiting et al., 2008; Thorarinsdottir, Scheuerer and Heinz, 2013).

Global summaries, such as a χ^2 goodness-of-fit test of the SBC histogram with respect to a uniform response, are natural options, but we found they did not perform particularly well in the above examples. The reason for this is that the deviation from uniformity tends to occur in only a few systematic ways, as discussed in Section 4.2, whereas these tests consider only global behavior and hence do not exploit these known failure modes. A potential alternative is to report a number of summaries that are designed to be sensitive to the specific types of deviation from uniformity we might expect to see.

Going beyond SBC, another future direction is to examine approximate calibration based on only one or two prior predictive replications, which in practice can be enough to catch many of computational problems, often arising from poorly identified models or bugs in our code.

A limitation of SBC is the requirement of a generative model: the prior $\pi(\theta)$ should be a proper distribution and one should be able to sample from the joint model $\pi(\theta, y)$. These conditions apply in many but not all Bayesian models in practice. Another concern is the behavior when the prior is very weak relative to the likelihood, so that a huge number of replications would be needed to test the computation in the relevant region of parameter space. This could cause problems if the computation works well for some parameter values but not for others.

As discussed in Gelman et al. (2020), it is common in practice to use weak priors, out of some combination of respect for non-Bayesian tradition and concern about missing important data features by oversmoothing. When priors are very weak, prior predictive simulation can result in datasets that result in computational challenges (for example, discrete data that are nearly all zeroes) which in turn can cause problems with SBC even if the algorithm in question would perform well on more realistic datasets. More generally, there is a tension between the goal of using SBC—or any validation procedure—to test a general algorithm, and the goal of checking that it works well for a particular example.

Acknowledgements

We thank Bob Carpenter, Chris Ferro, Mitzi Morris, and two anonymous reviewers for their helpful comments. The plot in Figure 12(c) shares the same derivation as the `inla.ks.plot` function written by Finn Lindgren and found in the R-INLA package. We thank the Academy of Finland (grant 313122), Sloan Foundation (grant G-2015-13987), U.S. National Science Foundation (grant CNS-1730414), Office of Naval Research (grants N00014-15-1-2541, N00014-16-P-2039, and N00014-19-1-2204), Defense Advanced Research Projects Agency (grant DARPA BAA-16-32), Institute of Education Sciences (grant R305D190048), and Schmidt Futures for partial support of this research.

References

Central Intelligence Agency (2018). Country comparison :: HIV/AIDS - Adult prevalence rate. World Factbook. https://www.cia.gov/library/publications/the-world-factbook/rankorder/2155rank.html. Accessed: 2018-04-04.

Anderson, J. L. (1996). A method for producing and evaluating probabilistic forecasts from ensemble model integrations. *Journal of Climate* **9** 1518–1530.

Betancourt, M. (2017). A conceptual introduction to Hamiltonian Monte Carlo. *arXiv:1701.02434*.

Betancourt, M. J. and Girolami, M. (2013). Hamiltonian Monte Carlo for hierarchical models. *arXiv:1312.0906*.

Cook, S. (2006). BayesValidate (R package). https://CRAN.R-project.org/package=BayesValidate.

Cook, S. R., Gelman, A. and Rubin, D. B. (2006). Validation of software for Bayesian models using posterior quantiles. *Journal of Computational and Graphical Statistics* **15** 675–692.

Corsi, D. J., Neuman, M., Finlay, J. E. and Subramanian, S. (2012). Demographic and health surveys: a profile. *International Journal of Epidemiology* **41** 1602–1613.

Fuglstad, G.-A., Simpson, D., Lindgren, F. and Rue, H. (2019). Constructing priors that penalize the complexity of Gaussian random fields. *Journal of the American Statistical Association* **114** 445–452.

Gelman, A. (2017). Correction to Cook, Gelman, and Rubin (2006). *Journal of Computational and Graphical Statistics* **26** 940.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A. and Rubin, D. B. (2013). *Bayesian Data Analysis, third edition*. CRC Press.

Gelman, A., Vehtari, A., Simpson, D., Margossian, C. C., Carpenter, B., Yao, Y., Kennedy, L., Gabr, J., Bürkner, P.-C. and Modrák, M. (2020). Bayesian workflow. *arXiv:2011.01818*.

Geweke, J. (2004). Getting it right: Joint distribution tests of posterior simulators. *Journal of the American Statistical Assocation* **98** 799–804.

Geyer, C. J. (1992). Practical Markov chain Monte Carlo. *Statistical Science* 473–483.

Gneiting, T., Stanberry, L. I., Grimit, E. P., Held, L. and Johnson, N. A. (2008). Assessing probabilistic forecasts of multivariate quantities, with an application to ensemble predictions of surface winds. *Test* **17** 211.

Hamill, T. M. (2001). Interpretation of rank histograms for verifying ensemble forecasts. *Monthly Weather Review* **129** 550–560.

Heidemanns, M., Gelman, A. and Morris, E. (2020). An updated dynamic Bayesian forecasting model for the 2020 election. *Harvard Data Science Review* **2**.

Hoffman, M. D. and Gelman, A. (2014). The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research* **15** 1351–1381.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A. and Blei, D. M. (2017). Automatic differentiation variational inference. *Journal of Machine Learning Research* **18** 430–474.

Lindgren, F., Rue, H. and Lindström, J. (2011). An explicit link between Gaussian fields and Gaussian Markov random fields: The stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **73** 423–498.

Neal, R. M. et al. (2011). MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo* (S. Brooks, A. Gelman, G. L. Jones and X. L. Meng, eds.) CRC Press.

Papaspiliopoulos, O., Roberts, G. O. and Sköld, M. (2007). A general framework for the parametrization of hierarchical models. *Statistical Science* **22** 59–73.

Rubin, D. B. (1981). Estimation in parallel randomized experiments. *Journal of Educational Statistics* **6** 377–401.

Rue, H., Martino, S. and Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **71** 319–392.

Rue, H., Riebler, A., Sørbye, S. H., Illian, J. B., Simpson, D. P. and Lindgren, F. K. (2017). Bayesian computing with INLA: A review. *Annual Review of Statistics and Its Application* **4** 395–421.

Seppä, K., Rue, H., Hakulinen, T., Läärä, E., Sillanpää, M. J. and Pitkäniemi, J. (2019). Estimating multilevel regional variation in excess mortality of cancer patients using integrated nested Laplace approximation. *Statistics in Medicine* **38** 778–791.

Simpson, D., Rue, H., Riebler, A., Martins, T. G., Sørbye, S. H. et al. (2017). Penalising model component complexity: A principled, practical approach to constructing priors. *Statistical Science* **32** 1–28.

Säilynoja, T., Bürkner, P.-C. and Vehtari, A. (2021). Graphical test for discrete uniformity

and its applications in goodness of fit evaluation and multiple sample comparison. *Technical report, Department of Computer Science, Aalto University.*

Thorarinsdottir, T. L., Scheuerer, M. and Heinz, C. (2013). Assessing the calibration of high-dimensional ensemble forecasts using rank histograms. *Journal of Computational and Graphical Statistics* **25** 105–122.

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B. and Bürkner, P.-C. (2021). Rank-normalization, folding, and localization: An improved $\widehat{R}$ for assessing convergence of MCMC. *Bayesian Analysis*. doi:10.1214/20-BA1221.

Wakefield, J., Simpson, D. and Godwin, J. (2016). Comment: Getting into Space with a Weight Problem. *Journal of the American Statistical Association* **111** 1111–1118.

Appendix A: Code Listings

We advise the reader to keep in mind that the Stan modeling language parameterizes the normal distribution using the mean and standard deviation whereas we have used a mean and variance parameterization throughout this text.

Listing 1. *Data generating process for linear regression:*

```
data {
  int<lower=1> N;
  real X[N];
}
generated quantities {
  real beta;
  real alpha;
  real y[N];
  beta = normal_rng(0, 10);
  alpha = normal_rng(0, 10);
  for (n in 1:N)
    y[n] = normal_rng(X[n] * beta + alpha, 1.2);
}
```

Listing 2. *Inference model for linear regression:*

```
data {
  int<lower=1> N;
  vector[N] X;
  vector[N] y;
}
parameters {
  real beta;
  real alpha;
}
model {
  beta ~ normal(0, 10);
  alpha ~ normal(0, 10);
  y ~ normal(X * beta + alpha, 1.2);
}
```

Listing 3. *8 schools, centered parameterization:*

```
data {
  int<lower=0> J;
  real y[J];
  real<lower=0> sigma[J];
}
parameters {
  real mu;
  real<lower=0> tau;
  real theta[J];
}
model {
  mu ~ normal(0, 5);
```

```
  tau ~ normal(0, 5);
  theta ~ normal(mu, tau);
  y ~ normal(theta, sigma);
}
```

Listing 4. *8 schools, non-centered parameterization:*

```
data {
  int<lower=0> J;
  real y[J];
  real<lower=0> sigma[J];
}
parameters {
  real mu;
  real<lower=0> tau;
  real theta_tilde[J];
}
transformed parameters {
  real theta[J];
  for (j in 1:J)
    theta[j] = mu + tau * theta_tilde[j];
}
model {
  mu ~ normal(0, 5);
  tau ~ normal(0, 5);
  theta_tilde ~ normal(0, 1);
  y ~ normal(theta, sigma);
}
```

Appendix B: Proof of Theorem 2

Theorem 2. *Let $\tilde{\theta} \sim \pi(\theta)$, $\tilde{y} \sim \pi(y \mid \tilde{\theta})$, and $\{\theta_1, \ldots, \theta_L\}$ sampled independently from $\pi(\theta \mid \tilde{y})$ for any joint distribution $\pi(y, \theta)$. The rank statistic of any scalar summary of θ is uniformly distributed over the integers $\{(0, 1, \ldots, L\}$.*

Proof. Consider the scalar summary $h : \Theta \to \mathbb{R}$ and let $\tilde{h} = h(\tilde{\theta})$ be its evaluation with respect to the prior sample with $h_l = h(\theta_l)$ the evaluation of the random variable with respect to one draw from the posterior sample. Similarly let $\pi(h)$ and $\pi(h \mid \tilde{y})$ denote the probability density function of the prior density function and posterior density function, respectively.

Without loss of generality, relabel the elements of the posterior sample such that they are ordered with respect to the random variable,

$$h_1 \leq h_2 \leq \ldots \leq h_{L-1} \leq h_L.$$

We can then write the probability mass function of the prior rank statistic as

$$\begin{aligned}
\pi(r) &= \int \mathrm{d}\tilde{h}\,\mathrm{d}y\,\pi(\tilde{y},\tilde{h})\frac{L!}{r!(L-r)!}\mathbb{P}\left[h_l < \tilde{h}\right]\cdot\mathbb{P}\left[h_l \geq \tilde{h}\right] \\
&= \frac{L!}{r!(L-r)!}\int \mathrm{d}\tilde{h}\,\mathrm{d}y\,\pi(\tilde{y},\tilde{h})\mathbb{P}\left[h_l < \tilde{h}\right]\cdot\mathbb{P}\left[h_l \geq \tilde{h}\right] \\
&= \frac{L!}{r!(L-r)!}\int \mathrm{d}\tilde{h}\,\mathrm{d}y\,\pi(\tilde{y},\tilde{h})\left[\prod_{l=1}^{r}\int_{-\infty}^{\tilde{h}}\mathrm{d}h_l\,\pi(h_l \mid \tilde{h},\tilde{y})\right]\left[\prod_{l=r+1}^{L}\int_{\tilde{h}}^{\infty}\mathrm{d}h_l\,\pi(h_l \mid \tilde{h},\tilde{y})\right] \\
&= \frac{L!}{r!(L-r)!}\int \mathrm{d}\tilde{h}\,\mathrm{d}y\,\pi(\tilde{y},\tilde{h})\left[\int_{-\infty}^{\tilde{h}}\mathrm{d}h_l\,\pi(h_l \mid \tilde{h},\tilde{y})\right]^{r}\left[\int_{\tilde{h}}^{\infty}\mathrm{d}h_l\,\pi(h_l \mid \tilde{h},\tilde{y})\right]^{L-r} \\
&= \frac{L!}{r!(L-r)!}\int \mathrm{d}\tilde{h}\,\mathrm{d}y\,\pi(\tilde{y},\tilde{h})\left[\int_{-\infty}^{\tilde{h}}\mathrm{d}h_l\,\pi(h_l \mid \tilde{h},\tilde{y})\right]^{r}\left[1-\int_{0}^{\tilde{h}}\mathrm{d}h_l\,\pi(h_l \mid \tilde{h},\tilde{y})\right]^{L-r}
\end{aligned}$$

Once we condition on the simulated observation $\tilde{y}$ the posterior samples are independent of the prior sample and

$$\pi(h_l \mid \tilde{h},\tilde{y}) = \pi(h_l \mid y).$$

Consequently

$$\begin{aligned}
\pi(r) &= \frac{L!}{r!(L-r)!}\int \mathrm{d}\tilde{h}\,\mathrm{d}y\,\pi(\tilde{y},\tilde{h})\left[\int_{-\infty}^{\tilde{h}}\mathrm{d}h_l\,\pi(h_l \mid y)\right]^{r}\left[1-\int_{0}^{\tilde{h}}\mathrm{d}h_l\,\pi(h_l \mid y)\right]^{L-r} \\
&= \frac{L!}{r!(L-r)!}\int \mathrm{d}\tilde{h}\,\mathrm{d}y\,\pi(\tilde{h} \mid y)\,\pi(y)\left[\int_{-\infty}^{\tilde{h}}\mathrm{d}h_l\,\pi(h_l \mid y)\right]^{r}\left[1-\int_{0}^{\tilde{h}}\mathrm{d}h_l\,\pi(h_l \mid y)\right]^{L-r} \\
&= \frac{L!}{r!(L-r)!}\int \mathrm{d}y\,\pi(y)\int \mathrm{d}\tilde{h}\,\pi(\tilde{h} \mid y)\left[\int_{-\infty}^{\tilde{h}}\mathrm{d}h_l\,\pi(h_l \mid y)\right]^{r}\left[1-\int_{0}^{\tilde{h}}\mathrm{d}h_l\,\pi(h_l \mid y)\right]^{L-r}
\end{aligned}$$

Moreover because the model from which $\tilde{\theta}$ and $\tilde{y}$ are simulating is the same as the model used to construct the posterior

$$\pi(h_l \mid y) = \pi(\tilde{h} \mid y).$$

We can then consider the change of variables

$$u(y) = \int_{-\infty}^{\tilde{h}}\mathrm{d}\theta\,\pi(\theta \mid y)$$

which gives

$$\begin{aligned}
\pi(r) &= \frac{L!}{r!(L-r)!} \int \mathrm{d}y \, \pi(y) \int \mathrm{d}u \, [u]^r \, [1-u]^{L-r} \\
&= \frac{L!}{r!(L-r)!} \int \mathrm{d}y \, \pi(y) \frac{r!(L-r)!}{(L+1)!} \\
&= \frac{L!}{r!(L-r)!} \frac{r!(L-r)!}{(L+1)!} \int \mathrm{d}y \, \pi(y) \\
&= \frac{1}{L+1} \int \mathrm{d}y \, \pi(y) \\
&= \frac{1}{L+1},
\end{aligned}$$

consistent with a uniform distribution over ranks, as desired.

□