ANNUAL REVIEWS

Annual Review of Statistics and Its Application

Approximate Methods for Bayesian Computation

Radu V. Craiu and Evgeny Levi

Department of Statistical Sciences, University of Toronto, Toronto, Ontario, Canada; email: radu.craiu@utoronto.ca

Annu. Rev. Stat. Appl. 2023. 10:379–99

First published as a Review in Advance on November 22, 2022

The *Annual Review of Statistics and Its Application* is online at statistics.annualreviews.org

https://doi.org/10.1146/annurev-statistics-033121-110254

Keywords

ABC, Bayesian synthetic likelihood, coresets, divide and conquer, Markov chain Monte Carlo, subsampling

Abstract

Rich data generating mechanisms are ubiquitous in this age of information and require complex statistical models to draw meaningful inference. While Bayesian analysis has seen enormous development in the last 30 years, benefitting from the impetus given by the successful application of Markov chain Monte Carlo (MCMC) sampling, the combination of big data and complex models conspire to produce significant challenges for the traditional MCMC algorithms. We review modern algorithmic developments addressing the latter and compare their performance using numerical experiments.

1. INTRODUCTION

The data science revolution has led to multiple pressure points in statistics. A statistical sample from a large population exhibits, in the twenty-first century, very different characteristics than what one would have seen merely a few years ago. The ubiquitous and almost continuous recording of many of our activities has made it relatively easy to collect enormous amounts of information that require analysis and interpretation. This drastic increase in data volume imposes sober reevaluations of most classical approaches to statistical inference.

The computational side of a Bayesian statistician's toolbox is perhaps most challenged by these developments. The impetus of Bayesian statistics that has been felt since the early 1990s has been given by the spectacular advances in computation, especially those around Markov chain Monte Carlo (MCMC) sampling. Thanks to methods in this class of algorithms, the statisticians have been liberated to think freely about the Bayesian model components used for a given problem, without worrying about the mathematical intractability of the analysis.

Indeed, given a data set $\mathbf{y}$, most of the pairings of a sampling density, $f(\mathbf{y}|\theta)$, and a prior, $p(\theta)$, result in a posterior distribution,

$$\pi(\theta|\mathbf{y}) = \frac{p(\theta)f(\mathbf{y}|\theta)}{\int p(\theta)f(\mathbf{y}|\theta)\mathrm{d}\theta}, \qquad 1.$$

that cannot be analyzed directly, usually because the denominator in Equation 1 cannot be computed analytically. The latter fact impedes the calculation of quantities of interest related to π, most of which can be expressed as

$$I = \int h(\theta)\pi(\theta|\mathbf{y})\mathrm{d}\theta \qquad 2.$$

for some function h that is determined by the question of interest. For instance, if θ is univariate and we let $h(\theta) = \theta^r$ in Equation 2, then I is equal to the rth moment of π, or $h(\theta) = \mathbf{1}_{(-\infty,\, t]}(\theta)$ leads to the cumulative distribution function of π at a point t.

The classical Monte Carlo method, devised by Metropolis & Ulam (1949) in the middle of the twentieth century, relies on sampling independently $\{\theta_1, \ldots, \theta_m\}$ from distribution π and approximating I with

$$\hat{I} = \frac{1}{m}\sum_{k=1}^{m} h(\theta_k). \qquad 3.$$

However, the unknown constant in Equation 1 creates a knowledge gap that an MCMC algorithm closes by constructing a Harris-recurrent, π-irreducible, aperiodic Markov chain whose stationary distribution is exactly the posterior distribution $\pi(\theta|\mathbf{y})$. The values taken by the chain make up the samples $\theta_1, \ldots, \theta_m$. A couple of issues emerge immediately. First, because π is the chain's stationary distribution, the samples will be approximatively distributed with π only after the chain has entered its stationary regime. Second, due to the Markov property, the samples are typically positively correlated (although for exceptions, see Frigessi et al. 2000, Craiu & Meng 2005) which reduces the amount of information they contain about π. To see that, let us imagine the extreme case in which the m samples are perfectly correlated, in which case they would provide very little information about π.

The success MCMC sampling had in boosting the use of Bayesian models is largely due to the ease of implementation of some of its most popular algorithms. For instance, the Metropolis–Hastings (MH) algorithm (Metropolis et al. 1953, Hastings 1970) can be implemented using the following recursive procedure.

- **Step 0:** Initialize the Markov chain at θ_0 and choose a proposal density $q(\cdot|\zeta)$ that may or may not depend on ζ.

- **Step *t*:** At the tth step ($0 \leq t \leq m-1$) do:

  PROP: Draw proposal ω_t from the density $q(\cdot|\theta_t)$;
  ACC: Set

$$\theta_{t+1} = \begin{cases} \omega_t & \text{with probability } \alpha_t \\ \theta_t & \text{with probability } 1-\alpha_t \end{cases}$$

  where

$$\alpha_t = \min\left\{1, \frac{\pi(\omega_t|\mathbf{y})q(\theta_t|\omega_t)}{\pi(\theta_t|\mathbf{y})q(\omega_t|\theta_t)}\right\}. \quad 4.$$

Because of the form of the acceptance probability (Equation 4), its calculation is not prevented by the unknown denominator in Equation 1. Nevertheless, computation of Equation 4 hinges on the ability to calculate the sampling density $f(\mathbf{y}|\theta)$ for any parameter value θ and to be able to do it m times. The challenges posed to Bayesian computation by the modern data and modeling environment have their roots in this implicit assumption.

In very broad strokes, one can speak of two main challenges in modern Bayesian computation. The first one concerns the computational price of calculating a likelihood when the data are massive, say of order N (i.e., of the order of hundreds of millions or even billions). Even in the tame case of independent and identically distributed (i.i.d.) observations, in order to know Equation 4, we will have to compute a likelihood (or sampling density) that involves N terms, and this will have to be repeated each time a new MCMC sample is produced. The cumulative cost is unsustainable, as a single MCMC iteration can take days. A second challenge emerges when the model's complexity keeps up with the data volume and yields an intractable likelihood so that Equation 4 simply cannot be computed analytically. Finally, a meta-challenge appears in Bayesian analyses that merge massive data with intractable models.

This article discusses MCMC-adjacent methodology that is used to alleviate the pressure posed on Bayesian computation by the above challenges. Space constraints impede the presentation of details and variants, but in all cases we present the main ideas and refer the interested reader to the relevant literature. In order to gauge their computational efficiency, we run numerical experiments where, using publicly available software packages, the algorithms are implemented on two statistical models.

In the next section we describe in more detail the challenges we just described, and Section 3 summarizes some of the solutions proposed to address them. When big data and complex model challenges combine, new solutions are needed. We discuss in Section 4 some recent contributions to address this double jeopardy. Numerical experiments meant to illustrate and compare different algorithms are reported in Section 5. The article ends with comments and discussion of future directions for research.

2. MODERN CHALLENGES FOR BAYESIAN COMPUTATION: MASSIVE DATA

Consider data $\mathbf{y}$ collected on N independent items so that $\mathbf{y} = \{y_1, \ldots, y_N\} \in \mathcal{X}^N$, and denote by $f(\mathbf{y}|\theta)$ the sampling distribution that depends on parameter $\theta \in \Theta \subset \mathbf{R}^d$. At each iteration of the MH sampler, one needs to compute $f(\mathbf{y}|\omega_t) = \prod_{k=1}^N f(y_k|\omega_t)$, where ω_t is the proposal in Equation 4. Modern applications often rely on data that are large enough so that the repeated calculation of $f(\mathbf{y}|\omega_t)$ is impractical or even impossible. It is also not unusual for data size to be so large as to prohibit storage on a single machine, so that computation of the likelihood also involves repeated communication between multiple machines, thus adding significantly to the computational burden.

Prompted by the obstacle of large data, computational Bayesians have designed a number of approaches to alleviate the problem. Two general ideas currently stand out in terms of popularity and usage: divide and conquer (DAC) strategies and subsampling with minimum loss of information.

2.1. Divide and Conquer

The DAC approach is based on partitioning the sample into a number of subsamples, called batches, that are analyzed separately on a number of workers [central processing units (CPUs), graphics processing units (GPUs), servers, etc.]. After the batch-specific estimates about the parameter of interest are obtained, the results are combined so that the analyst recovers a large part of—or, ideally, all of—the information that would have been available if the whole sample were analyzed in the usual way, on a single machine. While this idea seems applicable in a wide range of scenarios, there are a couple of constraints that restrict its generality. First, the procedure is computationally effective if it is designed to minimize, or preferably eliminate, communication between the workers before combining the batch-specific results. Second, it is often difficult to produce an accurate assessment of the resulting loss of information at the combining stage. Some of the first proponents of DAC for MCMC sampling were Neiswanger et al. (2013), Scott et al. (2016), and Wang & Dunson (2013). In their approach, the subposterior distribution corresponding to the jth batch is defined as

$$\pi^{(j)}(\theta|\mathbf{y}^{(j)}) \propto f(\mathbf{y}^{(j)}|\theta)[p(\theta)]^{1/J}, \qquad 5.$$

where f and p are as in Equation 1, $\mathbf{y}^{(j)}$ is the data that were assigned to batch j, $1 \leq j \leq J$, and J is the total number of batches. With this choice, one immediately gets that $\prod_{j=1}^{J} \pi^{(j)} \propto \pi(\theta|\mathbf{y})$. Both Neiswanger et al. (2013) and Scott et al. (2016) consider ways to combine samples from the subposteriors $\pi^{(j)}(\theta)$, $1 \leq j \leq J$, in situations in which all posteriors, batch-specific and full data ones, are Gaussian or can be approximated by mixtures of Gaussians. In this case, one can demonstrate that a weighted average of samples from all the $\pi^{(j)}$s have density π. The use of the Weierstrass transform for each posterior density, proposed by Wang & Dunson (2013), extends the range of theoretical validity beyond Gaussian distributions. The authors also establish error bounds between the approximation and the true posterior. Nemeth & Sherlock (2018) use a Gaussian process (GP) approximation of each subposterior. Once again, the Gaussian nature of the approximation makes recombination possible and relatively straightforward. Limitations of the method are strongly linked with those of GP-based estimation. For instance, when the subposterior samplers are sluggish, large MCMC samples might be needed, which, in turn, makes the calculation of the GP-based approximation very expensive. The idea of using the values of the subposterior at each MCMC sample is adopted also by Changye & Robert (2019), who propose the subposteriors $\pi^{(j)} \propto \{[p(\theta)]^{1/J} f(\mathbf{y}^{(j)}|\theta)\}^{\lambda_j}$. The scale factor λ_j is used to control the uncertainty in the subposterior. Alternative ways to define the subposteriors are produced by Entezari et al. (2018), who use $\pi^{(j)} \propto p(\theta)[f(\mathbf{y}^{(j)}|\theta)]^J$. The intuitive idea is to match the size of the original sample and the batch-specific one. Their approach has been applied successfully to Bayesian additive regression tree (Chipman et al. 2010, Pratola 2016) models.

2.2. Subsampling

Subsampling approaches are mostly developed under two assumptions. The first one is that with massive data one expects a certain amount of redundancy, so it is possible to obtain the same likelihood when we eliminate a proportion of the sample as long as the remaining observations are properly weighted. A simple illustration is one in which R observations are identical, so that $R - 1$ of them can be taken out of the likelihood calculation if the term corresponding to the

remaining one is raised to power R. The second idea is that one might use only a small percentage of the sample to find accurate (e.g., unbiased) approximations of the quantities needed to run an MCMC sampler. For instance, in the case of an MH sampler, the pseudomarginal approach of Andrieu & Roberts (2009) demonstrates that the stationary distribution of the chain is the same when the likelihoods involved in Equation 4 are replaced with unbiased estimators. The pseudomarginal idea continues to influence the subsampling methodology for MCMC. While some divide the subsampling methods into exact and approximate, we refrain from using similar taxonomy because, in our opinion, all such methods introduce some level of approximation into the computation of interest.

Early efforts include those of Korattikara et al. (2014) and Bardenet et al. (2014), who propose estimating the acceptance probability (Equation 4) using only a random subset of the data. The latter authors demonstrate that, with probability higher than a threshold set in place by the user, their method yields estimates that are equal to the ones produced by the full data likelihood. However, one does not know in advance the size of the sample needed at each iteration and thus must be able, in principle, to access most of it at all times. A review of early subsampling methods is provided by Bardenet et al. (2017).

2.2.1. Coresets. The process of establishing which sample points are redundant must have theoretical backing, lest it lead to a very different posterior distribution without any hope to control or assess the error incurred. The coreset approach of Campbell & Broderick (2019) offers theoretical guarantees about the quality of the approximation resulting from sample reduction. Consider the log-likelihood obtained from N i.i.d. observations,

$$l(\theta|\mathbf{y}) = \sum_{i=1}^{N} l_i(\theta|y_i), \tag{6.}$$

where $l_i(\theta|y_i) = \log f(y_i|\theta)$. The aim of the coreset method is to find a set of weights $\{w_i : 1 \le i \le N\}$, most of them zero, so that

$$\|\Lambda(\theta|\mathbf{W}, \mathbf{y}) - l(\theta|\mathbf{y})\| \le \epsilon \|l(\theta|\mathbf{y})\|, \tag{7.}$$

for all $\theta \in \Theta$, where $\mathbf{W} = (W_1, \ldots, W_N)$ is the vector of weights, and $\Lambda(\theta|\mathbf{W}, \mathbf{y}) = \sum_{i=1}^{N} W_i l_i(\theta|y_i)$ is the weighted log-likelihood of the coreset. The weights found by Campbell & Broderick (2019) are defined as

$$W_i = \frac{\sigma}{\sigma_i}\frac{M_i}{M}, \tag{8.}$$

where

$$\sigma_i = \sup_{\theta \in \Theta} \left\| \frac{l_i(\theta|y_i)}{l(\theta|\mathbf{y})} \right\| \tag{9.}$$

is called the sensitivity of the ith observation, $\sigma = \sum_{i=1}^{N} \sigma_i$, M is the size of the coreset, and $(M_1, \ldots, M_N) \sim \text{Multi}\,(M, \{\frac{\sigma_i}{\sigma} : 1 \le i \le N\})$ are multinomial draws. One can think of the sensitivity in Equation 9 as a measure of the influence of the ith observation on the whole likelihood as θ varies. As expected, the algorithm will retain observations that correspond to relatively higher likelihood values, but more importantly, it allows some evaluation of the error incurred when the sample is reduced. The ideas that led to the weights in Equation 8 illustrate the general principles of the approach, but improvements are possible when one considers other norms in Equations 7 and 9. For instance, Campbell & Broderick (2019) consider the l_is as vectors in a Hilbert space, link the norm to the inner product in the space, and include directionality in the selection of the coresets. The latter allows replacing the simultaneous selection of the coreset elements by a more

intuitive procedure in which samples are sequentially added to minimize the residual error. In Section 3 we implement the coreset approach for logistic regression, as presented by Huggins et al. (2016). For this model, the coreset is built along the principles delineated above and requires some specific tuning. The parameter space is taken to be a Euclidean ball of radius R, which is a reasonable working assumption in the case of a logistic regression with standardized covariates. The sensitivity measure for each point is modified after K-clustering the entire sample. A measure of spread within each cluster is used to construct upper bounds for the sensitivity of each point in the sample. The intuition guiding this choice is that clusters whose data vectors are tightly bundled together will be well represented in the coreset by only a few points, while clusters with more spread will need to contribute more points. Overall, the coreset construction is intuitive and offers many possible directions for future research. The biggest challenge is the evaluation of approximating error induced in the posterior when replacing the full sample by the coreset, although some promising initial results exist (Manousakas et al. 2020).

2.2.2. Random subsampling. One can think of coreset subsampling as a static approach, in the sense that the subsample is selected once and the Bayesian analysis is subsequently conducted using the coreset in lieu of the original sample. A more dynamic approach is considered by Quiroz et al. (2018), who propose to use all the data for inference, just not at once. Their idea is to use a different subset of the data each time the MCMC chain is updated. For instance, in the case of an MH sampler, a different subset of individuals will contribute to the likelihood needed in the calculation of Equation 4 at each iteration. Following the development of pseudomarginal strategies, Andrieu & Vihola (2015) studied the convergence properties of an MH or a random walk Metropolis (RWM) sampler in which the likelihood in Equation 4 is replaced by an unbiased estimator. They have shown that the efficiency of the MCMC sample increases when the variance of the unbiased estimator decreases.

The use of subsampling within MCMC proposed by Quiroz et al. (2018) can be applied quite generally, and it is attractive because it addresses both the construction of the unbiased estimator for the likelihood and the reduction of its variance.

Given a random subsample of $\mathbf{y}$ of size m, $\mathbf{y_u} = \{y_{u_1}, \ldots, y_{u_m}\}$, where $\mathbf{u} = \{u_1, \ldots, u_m\}$ are i.i.d. random variables uniformly distributed over $\{1, \ldots, N\}$, the estimator

$$l_m(\theta|\mathbf{y_u}) = \frac{1}{m}\sum_{k=1}^{m} l_{u_k}(\theta|y_{u_k}) \qquad 10.$$

is unbiased for the average log-likelihood $\frac{1}{N}l(\theta|\mathbf{y})$. However, when m is much smaller than N, it has a large variance, subjecting the pseudomarginal chain that uses Equation 10 instead of the full-sample likelihood in Equation 4 to an increased risk of poor mixing, since an unusually high value of the likelihood at the current state of the chain will make it unlikely to accept a proposal. A reduction in variance is desirable and can be achieved via control variates, $\mathbf{q}(\theta) = \{q_1(\theta), \ldots, q_N(\theta)\}$, and via a modified estimator of $l(\theta|\mathbf{y})$ in Equation 6,

$$\tilde{l}_m(\theta|\mathbf{y_u}, \mathbf{q}) = \sum_{i=1}^{N} q_i(\theta) + \frac{N}{m}\sum_{k=1}^{m}(l_{u_k}(\theta|y_{u_k}) - q_{u_k}(\theta)). \qquad 11.$$

The notation implies that $\mathbf{q}$ might change with θ. Indeed, when the likelihood is unimodal, the construction of the control variate follows that of Bardenet et al. (2017), who use for each θ a Taylor series expansion of $l(\theta|\mathbf{y})$ around a fixed point, θ^*, which is a point centrally located in Θ (e.g., the maximum likelihood estimate), so that for all $1 \le i \le N$,

$$q_i(\theta) = l_i(\theta^*|y_i) + (\theta - \theta^*)^{\mathrm{T}}\frac{\mathrm{d}}{\mathrm{d}\theta}l_i(\theta^*|y_i) + \frac{1}{2}(\theta - \theta^*)^{\mathrm{T}}\frac{\mathrm{d}^2}{\mathrm{d}\theta^2}l_i(\theta^*|y_i)(\theta - \theta^*). \qquad 12.$$

This control variate construction is called parameter expanded by Quiroz et al. (2018) because it is obtained using an expansion in the parameter space. With this modification, running an MH chain for, say, M iterations requires the evaluation of $N + mM$ item-specific likelihood terms, $l_i(\theta \mid y_i)$, for Equation 11 and MN for Equation 6. This can translate into significant reduction of computation effort when $m \ll N$.

When the likelihood is multimodal or the Taylor approximation is poor, for example, when θ and θ^* are distanced, the authors discuss an alternative construction that identifies a number of centroids $\mathbf{y}_1^*, \ldots, \mathbf{y}_r^*$ via clustering of the data and use Taylor series expansions around each centroid to define the so-called data expanded control variates.

We remark that this is not a standard application of the control variate swindle in Monte Carlo (Fieller & Hartley 1954) since, for instance, the random variables defined in Equation 12 are not identically distributed. The reduction in variance requires a careful derivation in which the source of variability is provided by the finite distribution of the random vector $(u_1, \ldots u_m)$. The latter can be sampled at random at each iteration, or one can use the ideas of Deligiannidis et al. (2018) and allow dependence between the us in consecutive iterations to further reduce the variance of Equation 11. In Section 5, we implement the subsampling method with random or correlated selection of indices and parameter or data expanded control variates.

Finally, we should also point out that while the estimators discussed are unbiased for the log-likelihood, this does not translate into an unbiased estimator for the likelihood itself. Therefore, an approximate correction term is applied to reduce the bias but does not dissolve it, which means that the pseudomarginal theory cannot be applied mutatis mutandis in this case. Therefore, the target distribution of the chain is perturbed, and one must assess the size of the error incurred. The authors produce a bound of the perturbation error and provide empirical evidence that their bound is conservative. Additional details and derivations are provided by Quiroz et al. (2018).

3. MODERN CHALLENGES FOR BAYESIAN COMPUTATION: INTRACTABLE LIKELIHOODS

So far we have looked at the pressure posed by the size of the sample on Bayesian computation. However, there are other hurdles that accompany a massive sample. Often, large data imply more information, which, in order to be used fully, requires a more complex model. As data become richer and modelers more ambitious, the likelihoods tend to get intractable, such as the ones used in population genetics (Pritchard et al. 1999, Beaumont et al. 2002), groundwater studies (Cui et al. 2018), hurricane surges (Plumlee et al. 2021), or climate change scenarios (Oyebamiji et al. 2015).

At first sight, it can be surprising that Bayesian inference can still be conducted when the likelihood is intractable. The likelihood provides a crucial analytical link between any parameter value and the probability of observing a given data set. When such a link is not analytically tractable, it must be inferred from simulations. Central to the latter approach is the ability to sample, given any value of the parameter, pseudodata from the model. To provide an intuition, imagine that infinite computational resources are available. Then one can see that for any $\theta \in \Theta$, it is possible to simulate enough pseudodata sets to approximate at any degree of precision the distribution of the observed data $f(\mathbf{y}_0|\theta)$, essentially filling the void left by the intractability of the likelihood. However, since computational resources are not infinite, ingenious ways are needed to reduce the computational burden. We discuss here two algorithms, approximate Bayesian computation (ABC) and Bayesian synthetic likelihood (BSL), that have gained popularity in the statistical and, more generally, the scientific communities.

3.1. Approximate Bayesian Computation

Our discussion of ABC will be brief, given the recent and excellent reviews of Robert (2014) and Sisson et al. (2018a) and the comprehensive handbook of ABC (Sisson et al. 2018b).

The ABC algorithm was initially proposed as an accept/reject sampler (Tavaré et al. 1997). Given any θ^* sampled from the prior $p(\theta)$, it assumes that is possible to generate pseudodata $\mathbf{y}$ from $f(\mathbf{y}|\theta^*)$. If the pseudodata and the original data are close enough, then the parameter θ^* is an approximate draw from the posterior $\pi(\theta|\mathbf{y}_0)$. We next frame "close enough" and "approximate draw" in precise mathematical terms and provide some justification for our choices.

Given $\epsilon > 0$, a distance $d : \mathbf{R}^p \times \mathbf{R}^p \to \mathbf{R}_+$, and summary statistic $S(\mathbf{y}) \in \mathbf{R}^p$, the ABC algorithm has the following steps:

1. Sample $\theta^* \sim p(\theta)$ and synthetic data $\mathbf{y} \sim f(\mathbf{y}|\theta^*)$.
2. If $d(S(\mathbf{y}), S(\mathbf{y}_0)) \leq \epsilon$ then accept θ^* as a sample from the approximate posterior $\pi_\epsilon(\theta|S(\mathbf{y}_0))$, the marginal (in θ) of the joint distribution

$$\pi_\epsilon(\theta, \mathbf{y}|S(\mathbf{y}_0)) \propto p(\theta) f(\mathbf{y}|\theta) \mathbf{1}_{\{d(S(\mathbf{y}),S(\mathbf{y}_0))<\epsilon\}}. \quad 13.$$

If it is possible to have $\mathbf{y} = \mathbf{y}_0$ (for instance, if $\mathbf{y}_0$ is a discrete random variable with finite support), we can choose $S(\mathbf{y}) = \mathbf{y}$ and $\epsilon = 0$, and then the approximate posterior is the true posterior, i.e., $\pi_\epsilon(\theta|\mathbf{y}_0) = \pi(\theta|\mathbf{y}_0)$. This is easier to see when both θ and $\mathbf{y}$ take discrete values. Then, one can easily see that ABC will draw θ_0 with probability

$$\Pr(\theta = \theta_0) \propto p(\theta_0) \Pr(\mathbf{y} = \mathbf{y}_0|\theta = \theta_0) \propto \pi(\theta_0|\mathbf{y}_0), \quad 14.$$

where Equation 14 holds because of the algorithm's construction with $S(\mathbf{y}) = \mathbf{y}$ and $\epsilon = 0$. The above can be easily extended to the case when S is a sufficient statistic. In general, models with the level of complexity that requires ABC will not have a low-dimensional sufficient statistic so the choice of S is central to the performance of the ABC algorithm (Fearnhead & Prangle 2012, Marin et al. 2014). The accept/reject form of the ABC sampler makes it inefficient when the prior and posterior place most of their mass on different regions of Θ. Recognizing this, Marjoram et al. (2003) proposed an ABC MCMC algorithm that relies on building an MH transition kernel, with state space $\{(\theta, \mathbf{y}) \in \mathbf{R}^d \times \mathcal{X}^n\}$, proposal distribution at iteration t, $q(\theta|\theta_t) \times f(\mathbf{y}|\theta)$, and target

$$\pi_\epsilon(\theta, \mathbf{y}|S(\mathbf{y}_0)) \propto p(\theta) f(\mathbf{y}|\theta) \mathbf{1}_{\{d(S(\mathbf{y}),S(\mathbf{y}_0))<\epsilon\}}, \quad 15.$$

for which the acceptance probability in Equation 4 can be computed exactly, because the intractable terms involving the likelihood, $f(\mathbf{y}|\theta)$, cancel out. Alternatives to this include the pseudomarginal approach of Lee et al. (2012) and the sequential Monte Carlo implementation of Sisson et al. (2007), Lee (2012), and Filippi et al. (2013).

3.2. Bayesian Synthetic Likelihood

Indirect inference was developed in econometrics (Smith 1993, Gourieroux et al. 1993) for complex data models that are intractable but can be sampled from. The central tenet is that a complex model of interest, $f(\mathbf{y}|\theta)$, can be well approximated using a tractable sampling model $g(\mathbf{y}|\phi)$ where $\dim(\phi) > \dim(\theta)$. In other words, the complex model can be approximated by a simpler model whose parameter space is of larger dimension and has a tractable likelihood. This greatly simplifies the original problem since, for instance, Bayesian estimation of θ is possible if one estimates its functional connection with ϕ (see, for instance, Gallant & McCulloch 2009).

The BSL algorithm (Price et al. 2018) relies on the synthetic likelihood (SL) approximation of Wood (2010), which falls squarely in the class of indirect inference methods. The idea hinges

on the assumption that the conditional distribution $p(S(\mathbf{y})|\theta)$ is well approximated by a multivariate Gaussian $\mathcal{N}(\mu(\theta), \Sigma(\theta))$ whenever $\mathbf{y} \sim f(\mathbf{y}|\theta)$. The SL is defined as $\mathrm{SL}(\theta) = n(S(\mathbf{y}); \mu(\theta), \Sigma(\theta))$, where $n(\cdot; \mu, \Sigma)$ is the density of a multivariate normal with mean μ and variance Σ. One can estimate $\mu(\theta)$ and $\Sigma(\theta)$ numerically for any θ. Given θ, it is enough to repeatedly sample pseudodata $\mathbf{y}_j \sim f(\mathbf{y}|\theta)$ and compute $S(\mathbf{y}_j)$ for $1 \le j \le K$ and then estimate $\hat{\mu}(\theta) = \frac{1}{K}\sum_{j=1}^{K} S(\mathbf{y}_j)$ and $\hat{\Sigma}(\theta) = \mathrm{SamVar}(\{S(\mathbf{y}_j) : 1 \le j \le K\})$, where SamVar is the sample variance of the computed statistics. BSL is then based on the approximation $\pi_{\mathrm{BSL}}(\theta|S(\mathbf{y}_0)) \propto p(\theta)\mathrm{SL}(\theta|S(\mathbf{y}_0))$, which can be explored via MCMC sampling using the following update rule at iteration $t > 0$:

- **PROP**: Generate $\theta^* \sim q(\cdot|\theta_t)$, estimate $\hat{\mu}_{\theta^*}$ and $\hat{\Sigma}_{\theta^*}$ from K pseudodata $\{\mathbf{y}_j \sim f(\mathbf{y}|\theta^*) : 1 \le j \le K\}$, and compute $\mathrm{SL}(\theta^*) = \mathcal{N}(S(\mathbf{y}_0); \hat{\mu}_{\theta^*}, \hat{\Sigma}_{\theta^*})$.
- **ACC**: Set $\theta_{t+1} = \theta^*$ with probability $\alpha = \min(1, \frac{p(\theta^*)\mathrm{SL}(\theta^*)q(\theta_t|\theta^*)}{p(\theta_t)\mathrm{SL}(\theta_t)q(\theta^*|\theta_t)})$ and $\theta_{t+1} = \theta_t$ otherwise.

4. DOUBLE JEOPARDY

The separate treatment of the challenges brought by big data or intractable models is artificial, and we anticipate that, more and more, the two challenges will have to be met simultaneously. Since the use of MCMC within ABC or BSL procedures requires repeated generation of pseudodata of the same size and complexity as the observed ones, it incurs unmanageable computational costs when the data are massive or the data generating procedure is expensive.

Some of the methods described within the first challenge are amenable to being used in combination with ABC or BSL. For instance, DAC strategies can be used for an intractable model if each worker runs a separate ABC MCMC sampler for each batch of data. The obvious caveat is the difficulty of ascertaining the loss of information after the merging stage. Unfortunately, more generalizable methods like those used for subsampling cannot be used within ABC or BSL.

A strategy customized to ABC and BSL samplers with large or complex data is proposed by Levi & Craiu (2022). We describe here a variation of their approach, which combines finite adaptation ideas and presampling of the proposals. Assuming that an MH transition kernel is used to implement ABC MCMC or BSL MCMC, the first B samples are used to tune the proposal distribution. For instance, if a Gaussian proposal is used, then its covariance matrix can be estimated using methods proposed by Haario et al. (2001), Roberts & Rosenthal (2009) or, in the case of multimodal targets, Craiu et al. (2009) or Pompe et al. (2020). The main contribution is to reduce the simulation time via precomputation performed on parallel processors. Specifically, a set of proposals is generated in an embarrassingly parallel procedure. The preprocessed draws are collected in reference set $\mathcal{Z} = \{(\xi_b, s_b = (s_b^{(1)}, \ldots, s_b^{(m)})^{\mathrm{T}}) : 1 \le b \le H\}$, where each parameter value ξ_b generated from the proposal distribution is paired with m pseudodata $\mathbf{w}_b^{(1)}, \ldots, \mathbf{w}_b^{(m)} \overset{\text{i.i.d.}}{\sim} f(\mathbf{w}|\xi_b)$ and the statistic $s_b^{(j)} = S(\mathbf{w}_b^{(j)})$ calculated for all $1 \le j \le m$. Note that the set $\mathcal{Z}$ is generated independently of the chain.

We illustrate here the use of $\mathcal{Z}$ to run the approximate BSL (ABSL) sampler proposed by Levi & Craiu (2022). If the chain's proposal at the tth iteration, θ^*, is identical to one element, say $\xi_b \in \mathcal{Z}$, and m is large, then we would not need to generate $\mathbf{y}_1, \ldots, \mathbf{y}_m \sim f(\mathbf{y}|\theta^*)$ since we already have the corresponding pseudodata statistics vectors, s_b, which can be used to estimate $\mu(\theta^*)$ and $\Sigma(\theta^*)$ and thus $\mathrm{SL}(\theta^*)$. While the intuition is attractive, it is impractical to faithfully implement it. For instance, using a large value for m when creating $\mathcal{Z}$ might still be too costly, and an exact match with an element in the reference set remains unattainable when the parameter space is continuous. However, if $\mathcal{Z}$ contains enough ξ values that are close enough to θ^*, one can still use them for estimating $\mathrm{SL}(\theta^*)$. Levi & Craiu (2022) build the reference set with $m = 1$ and propose

the use of K-nearest-neighbors (kNN) estimators for $\mu(\theta^*)$ and $\Sigma(\theta^*)$:

$$\tilde{\mu}(\theta^*) = \frac{\sum_{b=1}^{H}[W_b(\theta^*)\frac{1}{m}\sum_{j=1}^{m} s_b^{(j)}]}{\sum_{b=1}^{H} W_b(\theta^*)} \text{ and}$$

$$\tilde{\Sigma}(\theta^*) = \frac{\sum_{b=1}^{H}[W_b(\theta^*)\frac{1}{m}\sum_{j=1}^{m}(s_b^{(j)} - \hat{\mu}_{\theta^*})(s_b^{(j)} - \hat{\mu}_{\theta^*})^{\mathrm{T}}]}{\sum_{b=1}^{H} W_b(\theta^*)}, \quad 16.$$

where $W_b(\theta^*) = 1$ or $W_b(\theta^*) = 1 - \|\xi_b - \theta^*\|/\|\xi^* - \theta^*\|$ and $\xi^* = \max_{\xi\in\mathcal{Z}} \|\xi - \theta^*\|$, i.e., is the point in $\mathcal{Z}$ that is furthest away from θ^*. If H is large, it is likely that most of its elements will contribute little or not at all to the estimators in Equation 16. Instead of summing over all the H elements in $\mathcal{Z}$, it is then advisable to use only the K closest ξs to θ^*, where K is user defined and depends on the available computational power. In our numerical experiments we have used $W_b = 1$ for all $1 \le b \le K$ and $K = \lfloor\sqrt{H}\rfloor$. Clearly, the estimators in Equation 16 are consistent due to the properties of kNN estimators, but they are not unbiased, so pseudomarginal arguments cannot be invoked to justify the approach. Validity is demonstrated theoretically by showing that the perturbation induced when using the modified transition kernel can be controlled using the user-specified tuning parameters of the sampler (for details, see Levi & Craiu 2022, section 6).

A similar approach is used by Levi & Craiu (2022) for the ABC MCMC chain that targets the marginal posterior density of θ resulting from Equation 15, $\pi(\theta|S(\mathbf{y}_0)) \propto p(\theta)\Pr(d(S(\mathbf{y}), S(\mathbf{y}_0)) < \epsilon|\theta)$. Instead of using an unbiased estimator for $\Pr(d(S(\mathbf{y}), S(\mathbf{y}_0)) < \epsilon|\theta)$, which would require multiple pseudodata generated from $f(\mathbf{y}|\theta)$, they construct the kNN-based estimator from the collection $\mathcal{Z}$.

In the next section, we compare numerically the methods discussed so far using a couple of examples.

5. NUMERICAL EXPERIMENTS

In this section we present the performance of the discussed algorithms on two models: logistic regression and stochastic volatility. We compare the accuracy and computational efficiency of the described methods with a couple of benchmark MCMC algorithms that are widely known to perform very well in these cases. Specifically, we measure the performance of the methods presented in this article against the Polya–Gamma (PG) sampler (Polson et al. 2013) for the logistic regression, and the sequential Monte Carlo ABC (ABC SMC) (Sisson et al. 2007, Lee 2012) for the stochastic volatility model. The former is customized for logistic regression, and for the latter, the length of ϵ sequence is set at 15.

5.1. Description of the Simulation Settings

The following variations of the algorithms described in previous sections are implemented.

- **PG_DAC_J**: DAC algorithm with PG sampler that follows the setup of Scott et al. (2016) using Equation 5. The samples from each batch are combined proportionally to the inverse covariance matrices. J denotes the number of batches.
- **RW_SS**: Subsampling using the technique of Quiroz et al. (2018) with a random walk (RW) transition kernel. There are four variations corresponding to pairing parameter or data expansion with random or correlated index selection.
  - **RW_SS_P_R_m**: Parameter expansion and random index selection
  - **RW_SS_D_R_K_m**: Data expansion and random index selection

- **RW_SS_P_C_m**: Parameter expansion and correlated index selection; correlation ρ set at $\rho = 0.9999$
- **RW_SS_D_C_K_m**: Data expansion and correlated index selection; correlation ρ set at $\rho = 0.9999$

Note that m and K indicate the number of observations that will be evaluated with the actual log-likelihood and the number of clusters, respectively.

- **RW_CO_K_f**: Coreset method for logistic regression proposed by Huggins et al. (2016). The number of clusters and proportion of nonzero weights out of N are specified by K and f, respectively. Note that the radius R is calculated from the average sum of squared distances within each cluster, as suggested by Huggins et al. (2016). The coreset is used with an RWM sampling algorithm.
- **RW_ABC**: ABC MCMC algorithm using an RWM transition kernel for target Equation 15. Only one pseudodata set is generated for each proposal θ^*.
- **RW_AABC**: Approximate ABC MCMC algorithm proposed by Levi & Craiu (2022). Proposals from the history of the chain are used to estimate the likelihood using the kNN approach with uniform weights. Only one pseudodata set is generated at every iteration.
- **RW_BSL_m**: BSL MCMC algorithm with an RWM transition kernel. The distribution of the summary statistics is approximated by a Gaussian. The mean and covariance of the distribution are estimated by generating m pseudodata sets at each proposed θ^*.
- **RW_ABSL**: Approximate BSL MCMC algorithm proposed by Levi & Craiu (2022). Past results are used to estimate the mean and covariance of the summary statistics distribution using the kNN approach with uniform weights. Only one pseudodata set is generated at every iteration. More detail is provided in the **Supplemental Appendix**.

Supplemental Material >

With the exception of PG and ABC SMC, all the approximate samplers rely on an RWM kernel with Gaussian proposals to ensure consistency and comparability. The RWM kernels used here benefit from a finite-adaptation strategy, in which the covariance of the proposal is modified using the method of Haario et al. (2001), during the first B iterations that make up the burn-in period, and is kept fixed after that. The ABC, AABC, and ABC SMC samplers depend on the threshold ϵ and the ingredients needed to compute the distance d in Equation 15, and are determined following preliminary simulations, as detailed by Levi & Craiu (2022). Additional details about each sampling design are provided in the **Supplemental Appendix**.

For standard MCMC samplers, performance comparison is often reported in terms of the effective sample size (ESS) per second of CPU time, denoted ESS/cpu. The ESS is interpreted as the number of independent samples that would yield the same variance of the Monte Carlo estimator. A higher ESS value indicates a more efficient MCMC sampling algorithm, since it has been directly linked with the algorithm's computational uncertainty (e.g., Gong & Flegal 2016, Vats et al. 2019). The CPU time directly measures the computational cost in seconds, so ESS/cpu can be interpreted as a sampler's speed of generating information about the target.

All the samplers discussed here target a distribution different than the posterior of interest. Thus, in order to fully compare these samplers, one must consider the errors incurred because of this shift. Therefore, in addition to metrics designed to measure the efficiency of a regular MCMC sampler, such as ESS/cpu, we also use $R = 50$ independent replicates to produce estimates of Monte Carlo bias and variance. This led us to two measures of efficiency that are used to convey the performance of each method: the root mean square error (RMSE) and the ESS/cpu.

To fix the notions, let $\theta_{(t)}^{rs}$ represent the posterior samples from replicate $1 \le r \le R$, iteration $B \le t \le M$ (only draws obtained after burn-in are retained) and parameter component $1 \le s \le d$.

Similarly, $\tilde{\theta}^{rs}_{(t)}$ are posterior draws from the benchmark chain (only draws obtained after the burn-in period are retained). We also let θ^{s}_{true} denote the true parameter value that was used to generate the data. The following quantities are used for comparing computational efficiency:

$$\text{Bias}^2 = \text{Mean}_s\left(\left(\text{Mean}_{tr}(\theta^{rs}_{(t)}) - \theta^{s}_{\text{true}}\right)^2\right),$$

$$\text{VAR} = \text{Mean}_s(\text{Var}_r(\text{Mean}_t(\theta^{rs}_{(t)}))), \quad \text{and}$$

$$\text{RMSE} = \sqrt{\text{Bias}^2 + \text{VAR}},$$

where $\text{Mean}_t(a^{st})$ is defined as the average of $\{a^{st}\}$ over index t and, similarly, $\text{Var}_t(a^{st})$ and $\text{Cov}_t(a^{st})$ denote the sample variance and covariance, respectively.

Using the `coda` library in R, we compute ESS for each replicate and parameter's component ESS^{rs}. Letting CPU^{r} denote the total CPU time used for producing the MCMC samples in replicate r, we define ESS/cpu as:

$$\text{ESS/cpu} = \text{Mean}_{rs}(ESS^{rs}/CPU^{r}).$$

Note that we consider the average over all parameters and replicates. Generally, a sampler with a higher ESS/cpu is preferred because it yields a higher amount of information per unit of time. The ABC SMC sampler produces independent draws so its ESS is equal to the number of particles.

Finally, in order to frame the comparison in terms of unit-free measures, we report the performance relative to the benchmark samplers. This means that once we compute the RMSE for, say, method A, RMSE_A, we report instead $\text{RMSE}_A/\text{RMSE}_{\text{Bench}}$, where the denominator is the benchmark sampler's RMSE. Similarly, we also report the relative ESS/cpu performance.

5.2. Logistic Regression

This set of simulations contains the standard setting for the logistic regression model. The design $N \times d$ matrix X is generated by simulating each variable independently from Unif(0, 1). The leftmost column is a column of 1s (intercept). For $i = 1, \ldots, N$, Y_i is Bernoulli with $\Pr(Y_i = 1) = \text{logistic}(x_i \cdot \theta_{\text{true}})$. We considered two values for the sample size, N: 1,000 and 10,000, and two sets of parameters:

- $d = 2$ with the true parameter of $\theta_{\text{true}} = (-2, 2)$ and
- $d = 10$ with the true parameter of $\theta_{\text{true}} = (-2, 2, -3, 4, 1, 2, -3, -4, 2, 1)/3$.

We set the prior distribution to be $p(\theta) \sim \mathcal{N}(0, 4I_d)$, where $\theta \in R^d$ and I_d is $d \times d$ identity matrix. All the samplers are run for $M = 55{,}000$ iterations with burn-in set at $B = 15{,}000$.

For the DAC sampler we consider three values for the number of batches $J = 2, 3, 5$. We set $K = 4$ for the coreset method and compare four values for the fraction $f = 0.5, 0.1, 0.05, 0.01$. Finally, the values of the tuning parameters for the subsampling method are also variable—specifically, the number of data clusters, $K \in \{10, 50\}$, and the size of the subsample, $m \in \{20, 100\}$. Generally, K and m will depend on the sample size N and parameter dimension d. The recommendation is to select larger values for data expansion than for parameter expansion. In addition, using correlated indices requires smaller values for these hyperparameters. **Figures 1** and **2** present the simulation results for the scenarios with $N = 10{,}000$ and $d \in \{2, 10\}$. In the **Supplemental Appendix**, we include two additional scenarios: $N = 1{,}000, d = 2$ and $N = 10{,}000, d = 2$. The height of the bars represents the value of the relative measure, and we add the dashed line at 1 to make it easier to separate performance improvements from degradations.

Supplemental Material >

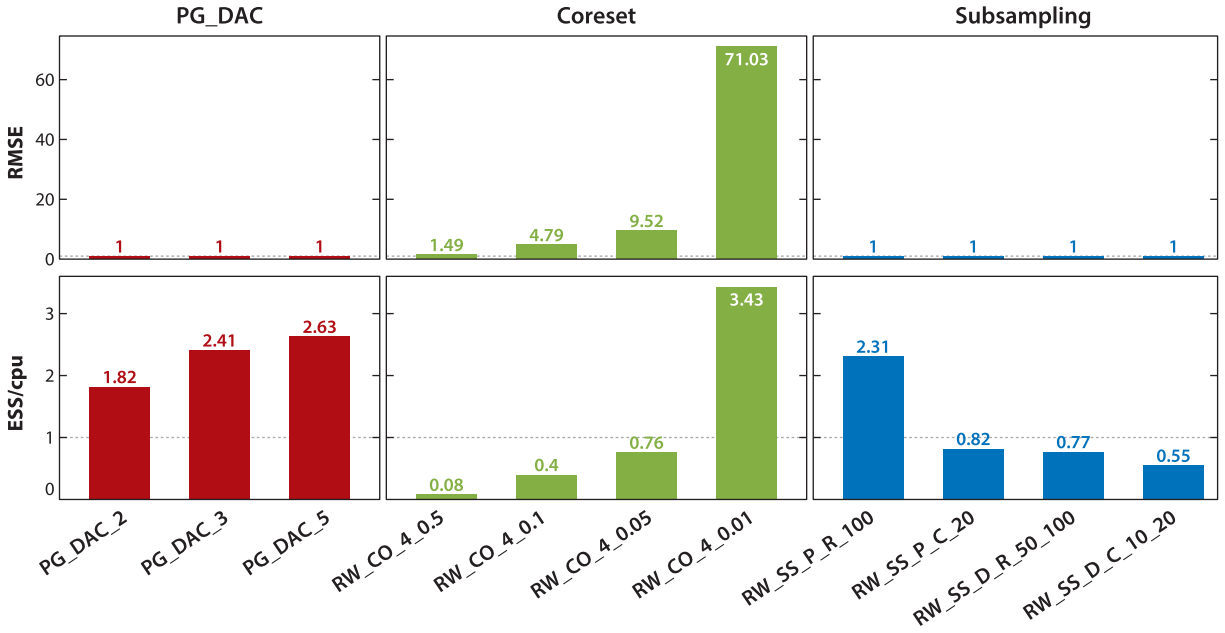

Figure 1

Logistic model: Relative RMSE and relative ESS/cpu when $N = 10{,}000$ and $d = 2$ for DAC-based samplers, coreset-based samplers, and subsampling-based samplers. Abbreviations: C, correlated index expansion; CO, coreset; D, data expansion; DAC, divide and conquer; ESS, effective sample size; ESS/cpu, ESS per second of central processing unit time; P, parameter expansion; PG, Polya–Gamma; R, random index selection; RMSE, root mean square error; RW, random walk; SS, subsampling.

From **Figures 1** and **2** a few lessons emerge. Combining PG with DAC produces good results, likely because the Gaussian approximation is accurate for such a large sample. The ESS/cpu grows with the number of batches.

The performance of the coreset-based algorithm yields a relatively high RMSE. To shed some light on this performance, we can recover from Huggins et al. (2016) the discrepancy, ϵ, between the original likelihood and the coreset one, as a function of the coreset size, mean sensitivity, and parameter dimension for $\delta = 0.10$.

Table 1 shows the average (over 50 replicates) discrepancy ϵ for different values of the sample size N, parameter dimension d, and data fraction f divided by the average maximum value of the full data likelihood. It is not surprising that ϵ increases as the fraction (i.e., the coreset size) decreases, but we also can see that the discrepancy is generally quite large, and this explains the poor performance of the sampler. The ESS/cpu measure beats PG only when using 1% of the samples, but this comes at the expense of a vastly inflated RMSE.

Overall, subsampling techniques show good results with very high ESS/cpu without sacrificing the accuracy of the posterior, when $d = 2$. The logistic posterior tends to be unimodal, so the parameter expansion methodology is more suitable, and clearly a larger concentration is achieved for $d = 2$ than for $d = 10$. The deterioration of the performance is clearly visible for $d = 10$, although the method still controls the RMSE at the PG level. Since the data do not exhibit any clusters, it is not surprising that the data expansion techniques are not competitive to the parameter expansion ones.

Note that the computational time for the calculation of the log-likelihood can be significantly reduced using the vectorization trick available in R. This method allows much faster calculation

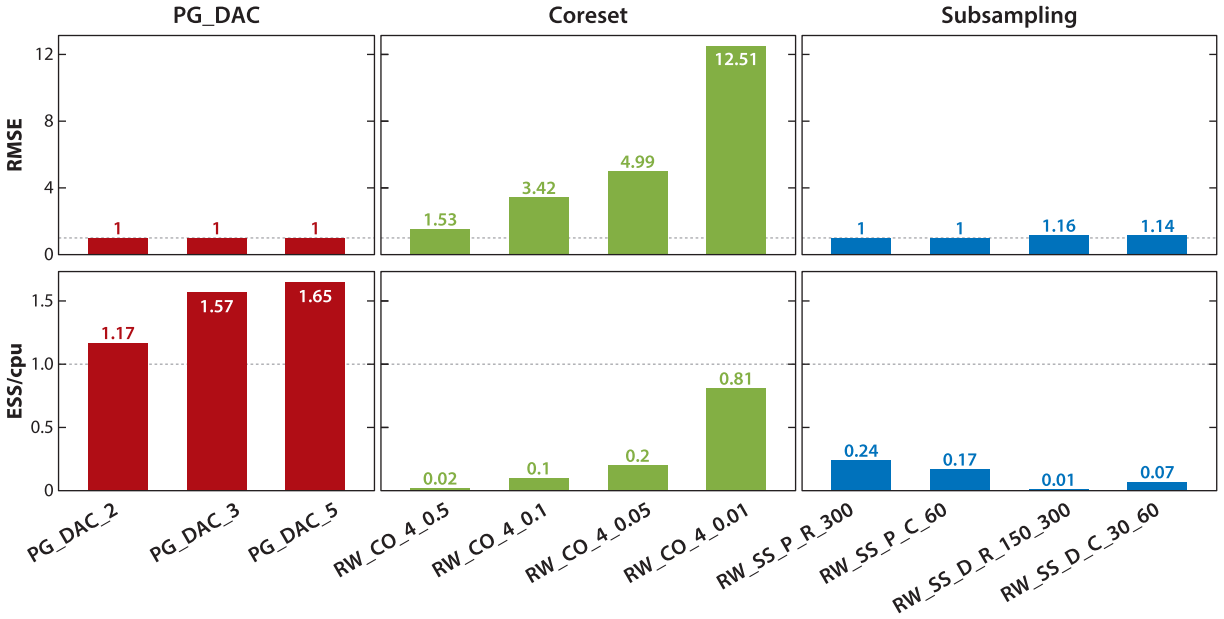

Figure 2

Logistic model: Relative RMSE and relative ESS/cpu when $N = 10{,}000$ and $d = 10$ for DAC-based samplers, coreset-based samplers, and subsampling-based samplers. Abbreviations: C, correlated index expansion; CO, coreset; D, data expansion; DAC, divide and conquer; ESS, effective sample size; ESS/cpu, ESS per second of central processing unit time; P, parameter expansion; PG, Polya–Gamma; R, random index selection; RMSE, root mean square error; RW, random walk; SS, subsampling.

by executing operations on the entire vectors of data instead of using a for loop that goes through all the N records one by one. This technique enabled us to increase the sample size to 100,000. The comparison of the samplers using the vectorization-induced speed-up can be found in the **Supplemental Appendix**.

Supplemental Material >

5.2.1. German credit data. This concerns data with a sample size that is not exceedingly large, but the dimension of the parameter is higher than we have considered so far. Specifically, the German credit data consist of 1,000 records and 49 predictors including the intercept (for more information, see Biswas et al. 2019). Most predictors are dummy variables taking only 0 and 1 values. The target/response is binary, with 70% of them being cases, so the response variable is quite balanced. Logistic regression is implemented to predict $\Pr(Y = 1)$ from the set of features.

Table 1 Coreset (logistic model): relative average discrepancy ϵ for different values of sample size N, parameter dimension d, and data fraction f

Fraction	ϵ			
	$N = 1{,}000$		$N = 10{,}000$	
	$d = 2$	$d = 10$	$d = 2$	$d = 10$
$f = 0.50$	3.217	3.602	0.685	1.160
$f = 0.10$	7.845	9.717	3.570	3.154
$f = 0.05$	11.992	13.972	6.138	4.461
$f = 0.01$	28.595	31.663	16.383	10.091

The numbers represent the average discrepancy divided by the average maximum value of the full data likelihood.

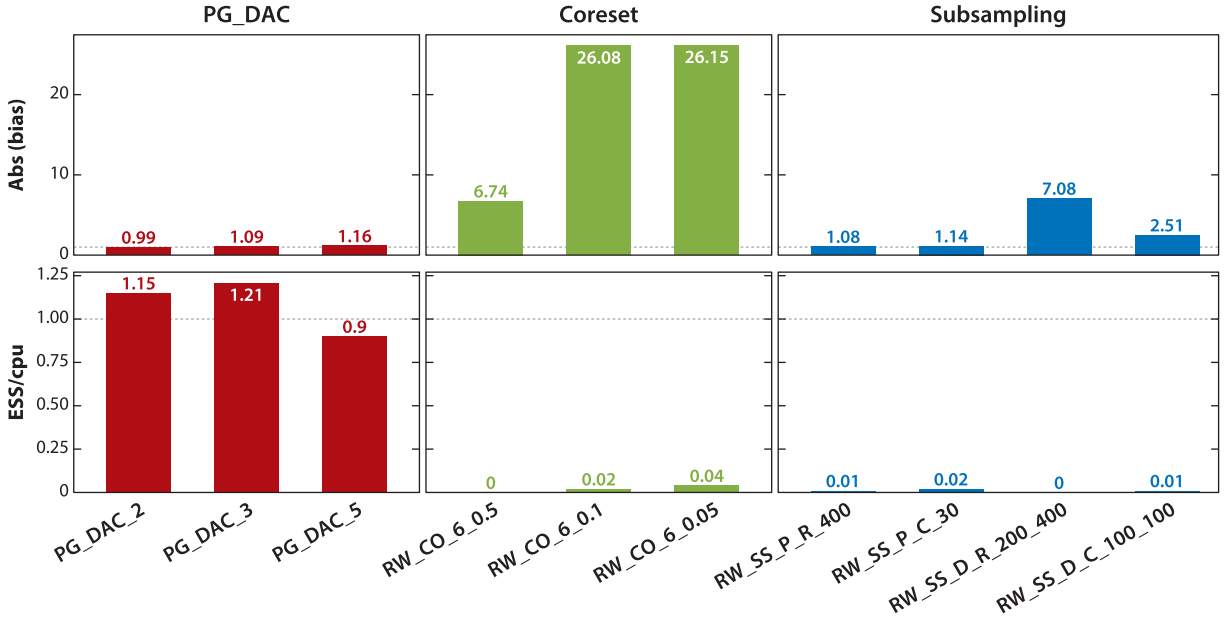

Figure 3

German credit data: Relative |bias| and relative ESS/cpu for DAC-based samplers, coreset-based samplers, and subsampling-based samplers. Abbreviations: C, correlated index expansion; CO, coreset; D, data expansion; DAC, divide and conquer; ESS, effective sample size; ESS/cpu, ESS per second of central processing unit time; P, parameter expansion; PG, Polya–Gamma; R, random index selection; RW, random walk; SS, subsampling.

Before fitting the model, we transform all the quantitative features by subtracting the minimum value and dividing by the range so their values are in the [0, 1] interval.

All the samplers are run for $N = 100{,}000$ iterations, burn-in is $B = 50{,}000$, and adaptation occurs every 500 chain updates. The performance of the samplers is presented in **Figure 3**. Note that the absolute values of the biases reported in the top row panels are calculated with respect to the maximum likelihood estimates, as the true parameter values are not known. We refer the reader to the **Supplemental Appendix** for additional metrics and details.

Supplemental Material >

The results are similar to the ones obtained in the previous subsection, but some additional elements emerge. We can see that with five batches, the DAC approach loses a bit in terms of bias and even more on the ESS/cpu side. None of the subsampling-based methods (including coreset-based) can compete with the PG sampler, likely because the signal-to-noise ratio is altered too much when implementing any of these methods. We should also recognize that PG is a Gibbs sampler, which, unlike RWM samplers, will move at every iteration. This makes a bigger difference when the parameter space has large dimensions since then the RWM chain often gets stuck, especially if the posterior exhibits strong dependence.

Based on these numerical experiments, we conclude that with a very large sample size, the first choice would be to use a DAC technique as long as the Gaussian approximation is likely to be accurate. The latter assessment will have to take into account the number of parameters and the nature of the model and data. If the Gaussian approximation is unsuitable, the subsampling methods can be used. The user will need to decide if the posterior is likely to be concentrated, so that they can use a parameter expansion, or if the data exhibit multiple clusters, in which case a data expansion is needed. In the latter case, an exploratory analysis is recommended to determine

reasonable values for the number of centroids, K. The size of the subsample m is typically decided based on the computational power available at the time of the analysis—we recommend using the largest possible value that can be handled by the system.

5.3. Stochastic Volatility

When analyzing stationary time series, it is frequently observed that there are periods of high and low volatility, a phenomenon known as volatility clustering (see, for example, Lux & Marchesi 2000). One way to model such behavior is through a stochastic volatility model, where variances of the observed time series depend on hidden states that themselves form a stationary time series. We work with the following model, which is indexed by parameter $\theta = (\theta_1, \theta_2, \theta_3)$:

$$\begin{aligned} &x_1 \sim \mathcal{N}(0, 1/(1-\theta_1^2)); \quad v_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1); \quad w_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1), \quad i = \{1,\dots,N\}; \\ &x_i = \theta_1 x_{i-1} + v_i, \quad i = \{2,\dots,N\}; \\ &y_i = \sqrt{\exp[\theta_2 + \exp(\theta_3)x_i]}\, w_i, \quad i = \{1,\dots,N\}. \end{aligned} \tag{17.}$$

Only data $\mathbf{y} = (y_1, \dots, y_N)$ are observed, and $(x_1, \dots, x_N)$ are latent/hidden states. The parameter $\theta_1 \in (-1, 1)$ controls the auto-correlation of hidden states, while θ_2 and θ_3 are unrestricted and relate to the hidden states' influence on the variability of the observed series. Given a hidden state, the distribution of the observed variable is Gaussian. We introduce the following priors, independently for each parameter:

$$\begin{aligned} \theta_1 &\sim \text{Unif}[0,1], \\ \theta_2 &\sim \mathcal{N}(0,1), \text{ and} \\ \theta_3 &\sim \mathcal{N}(0,1). \end{aligned} \tag{18.}$$

We set the true parameters to $\theta_{\text{true}} = (0.95, -2, -1)$ and consider three lengths of the time series, $N = 100$, 500 and 1,000. Note that the model does not admit a closed form log-likelihood but allows simulations of pseudodata sets. Therefore, for this model, we only consider simulation-based ABC samplers: AABC, BSL, ABSL, and the benchmark ABC SMC. The summary statistic used for all the samplers is $S(\mathbf{y}) \in \mathbf{R}^6$ and has the following components:

1. Average of $\mathbf{y}^2$
2. Standard deviation of $\mathbf{y}^2$
3. Sum of the first 5 auto-correlations of $\mathbf{y}^2$
4. Sum of the first 5 auto-correlations of binary series $\{\mathbf{1}_{\{y_i^2 < \text{quantile}(\mathbf{y}^2, 0.1)\}}\}_{i=1}^N$
5. Sum of the first 5 auto-correlations of binary series $\{\mathbf{1}_{\{y_i^2 < \text{quantile}(\mathbf{y}^2, 0.5)\}}\}_{i=1}^N$
6. Sum of the first 5 auto-correlations of binary series $\{\mathbf{1}_{\{y_i^2 < \text{quantile}(\mathbf{y}^2, 0.9)\}}\}_{i=1}^N$

The quantile($\mathbf{y}$, τ) is defined as the τth quantile of the sequence $\mathbf{y}$. We focus here on $\mathbf{y}^2$ and its auto-correlations because the model parameters only affect its variability; the auto-correlation of $\mathbf{y}$ is zero for any lag. Components 4–6 have been considered because the auto-correlations of those binary series, defined under different quantiles, are useful in characterizing a time series (Dette et al. 2015, Schmitt et al. 2015). The ABC, AABC, BSL, and ABSL samplers are run for $M =$ 55,000 iterations. The burn-in period is of length $B = 15{,}000$, with adaptation taking place every other 200 iterations.

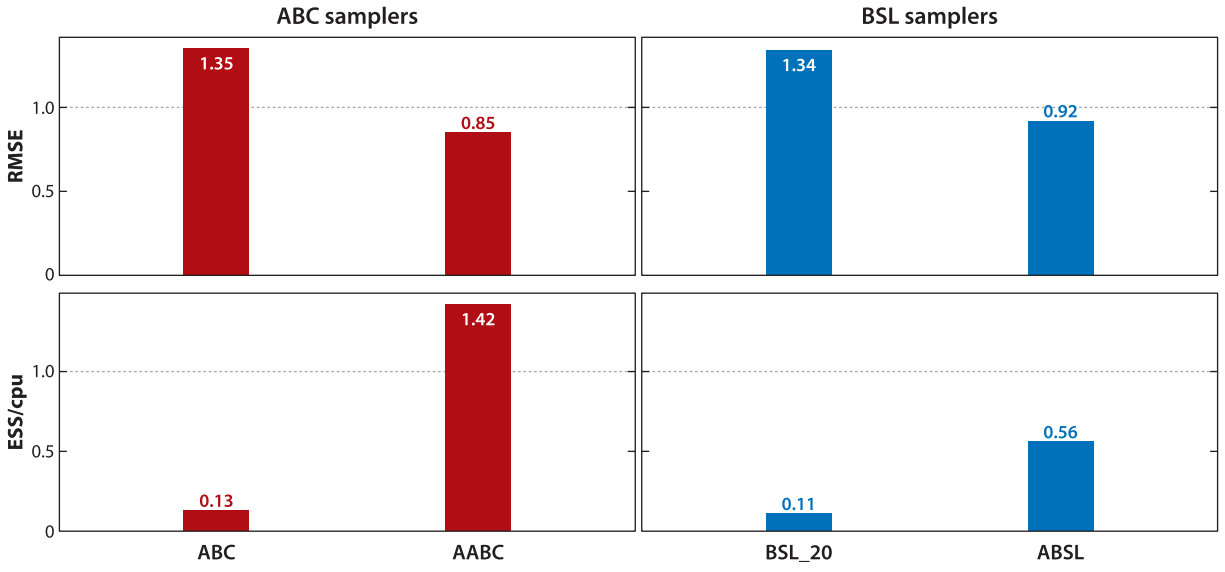

Figure 4

Stochastic volatility model: Relative RMSE and ESS/cpu when $N = 500$ for ABC-based samplers and BSL-based samplers. Abbreviations: ABC, approximate Bayesian computation; BSL, Bayesian synthetic likelihood; ESS/cpu, effective sample size per second of central processing unit time; RMSE, root mean square error.

Figures 4 and **5** present the simulation results when, respectively, $N = 500$ and $N = 1{,}000$. The ABC and BSL samplers exhibit loss in terms of both RMSE and ESS/cpu when compared to the benchmark. The BSL is more costly since we generate 20 pseudodata sets at each iteration. Not surprisingly, using precomputation designs reduces the CPU time, so we see a bump in efficiency for AABC and ABSL. Less obvious is the reduction in RMSE, which is due to the increase in the number of pseudodata one can use while still saving computational time and the higher acceptance rate. These findings mirror those of Levi & Craiu (2022), and we refer the reader to that article for more in-depth explanations. In this example, ABC-based samplers outperform BSL ones. The likely reason is that the Gaussian approximation on which BSL relies is not accurate for this choice of the summary statistic, $S(\mathbf{y})$.

Overall, we find reasons for cautious optimism in these numerical results. They show that careful and controlled injection of noise in the transition kernel can bring real practical benefits.

6. CONCLUSION AND FUTURE DIRECTIONS

The Bayesian computational community finds itself at an inflection point. Traditional MCMC computation is no longer tenable for complex problems. The new ideas and developments discussed here significantly reduce the computational costs or bypass the intractability of the likelihood but introduce additional layers of approximation. The latter requires a careful theoretical analysis to make sure that incurred errors are realistically controllable via tuning parameters.

Complex models are often defined using high-dimensional parameters. MCMC methods efficiently sample high-dimensional spaces as long as there are no bottlenecks or regions of small probability that the chain has difficulty traversing. Adaptive MCMC methods (Andrieu & Thoms 2008, Hoffman & Gelman 2014, Yang et al. 2019, Pompe et al. 2020) have been proven effective for sampling in high-dimensional spaces with unfriendly geometries. Injecting adaptive ideas

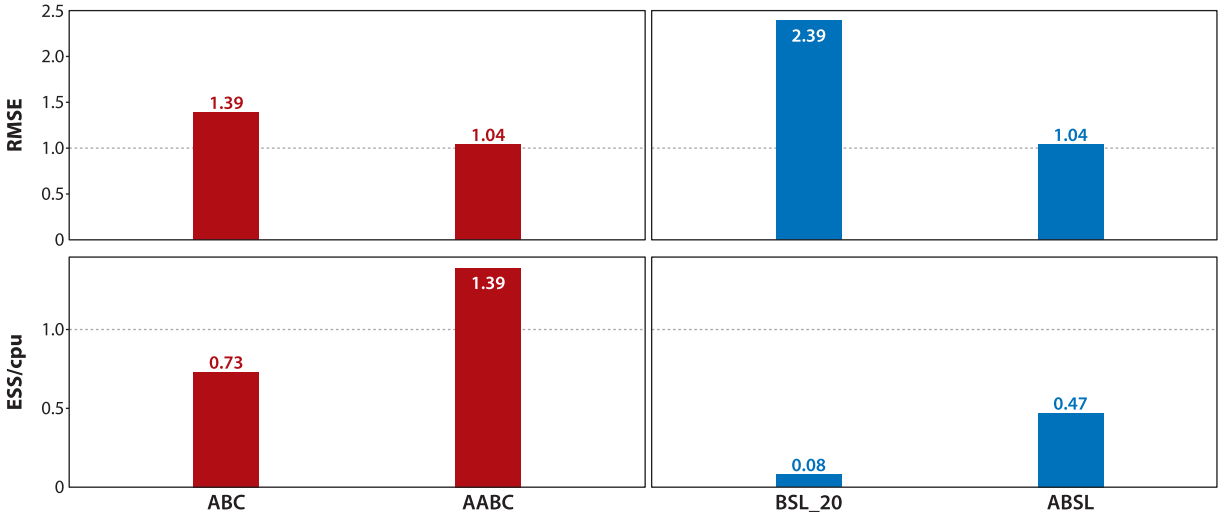

Figure 5

Stochastic volatility model: Relative RMSE and ESS/cpu when $N = 1{,}000$ for ABC-based samplers and BSL-based samplers. Abbreviations: ABC, approximate Bayesian computation; BSL, Bayesian synthetic likelihood; ESS/cpu, effective sample size per second of central processing unit time; RMSE, root mean square error.

into the world of sampling with intractable targets is hindered by stringent conditions that need to be satisfied by an adaptive transition kernel, e.g., the containment condition (Bai et al. 2011, Łatuszyński & Rosenthal 2014). Some inroads have been made into eliminating the latter by Craiu et al. (2015) and Rosenthal & Yang (2018), so we expect to see more adaptive designs permeating in pseudodata-generation-type samplers.

Constraints on article length and considerations of subject matter consistency have prevented us from discussing methods that do not rely on MCMC sampling to perform Bayesian inference, such as variational Bayes (Blei et al. 2017) or integrated nested Laplace approximation (Rue et al. 2017). These are active research threads that continue to develop rapidly under the impetus provided by the expansion of data science and the explosive growth of machine learning methods and other computationally demanding domains of information processing. Creative intertwining of most of the ideas and methods mentioned in this article will likely continue well into the future, but we believe that entirely new perspectives are also necessary in order to create the automatization of computation that is required if widespread use of Bayesian methods is to be seen in the twenty-first century.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

This research has been funded by the Natural Sciences and Engineering Research Council (NSERC) of Canada. The authors thank Nancy Reid for the invitation to write this article, Alicia Carriquiry for guidance in defining the article's scope, and an anonymous reviewer for a number of suggestions that have led to important improvements.

LITERATURE CITED

Andrieu C, Roberts GO. 2009. The pseudo-marginal approach for efficient Monte Carlo computations. *Ann. Stat.* 37(2):697–725

Andrieu C, Thoms J. 2008. A tutorial on adaptive MCMC. *Stat. Comput.* 18:343–73

Andrieu C, Vihola M. 2015. Convergence properties of pseudo-marginal Markov chain Monte Carlo algorithms. *Ann. Appl. Probab.* 25(2):1030–77

Bai Y, Roberts GO, Rosenthal JS. 2011. On the containment condition for adaptive Markov chain Monte Carlo algorithms. *Adv. Appl. Stat.* 21(1):1–54

Bardenet R, Doucet A, Holmes C. 2014. Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach. *PMLR* 32(1):405–13

Bardenet R, Doucet A, Holmes C. 2017. On Markov chain Monte Carlo methods for tall data. *J. Mach. Learn. Res.* 18(1):1515–57

Beaumont MA, Zhang W, Balding DJ. 2002. Approximate Bayesian computation in population genetics. *Genetics* 162(4):2025–35

Biswas N, Jacob PE, Vanetti P. 2019. Estimating convergence of Markov chains with L-lag couplings. In *NIPS'19: Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 7389–99. N.p.: NeurIPS

Blei DM, Kucukelbir A, McAuliffe JD. 2017. Variational inference: a review for statisticians. *J. Am. Stat. Assoc.* 112(518):859–77

Campbell T, Broderick T. 2019. Automated scalable Bayesian inference via Hilbert coresets. *J. Mach. Learn. Res.* 20(1):551–88

Changye W, Robert CP. 2019. Parallelising MCMC via random forests. arXiv:1911.09698 [stat.CO]

Chipman HA, George EI, McCulloch RE. 2010. BART: Bayesian additive regression trees. *Ann. Appl. Stat.* 4(1):266–98

Craiu RV, Gray L, Łatuszyński K, Madras N, Roberts GO, Rosenthal JS. 2015. Stability of adversarial Markov chains, with an application to adaptive MCMC algorithms. *Ann. Appl. Probab.* 25(6):3592–623

Craiu RV, Meng XL. 2005. Multiprocess parallel antithetic coupling for backward and forward Markov chain Monte Carlo. *Ann. Stat.* 33(2):661–97

Craiu RV, Rosenthal JS, Yang C. 2009. Learn from thy neighbor: parallel-chain adaptive and regional MCMC. *J. Am. Stat. Assoc.* 104:1454–66

Cui T, Peeters L, Pagendam D, Pickett T, Jin H, et al. 2018. Emulator-enabled approximate Bayesian computation (ABC) and uncertainty analysis for computationally expensive groundwater models. *J. Hydrol.* 564:191–207

Deligiannidis G, Doucet A, Pitt MK. 2018. The correlated pseudomarginal method. *J. R. Stat. Soc. Ser. B* 80(5):839–70

Dette H, Hallin M, Kley T, Volgushev S. 2015. Of copulas, quantiles, ranks and spectra: an l_1-approach to spectral analysis. *Bernoulli* 21(2):781–831

Entezari R, Craiu RV, Rosenthal JS. 2018. Likelihood inflating sampling algorithm. *Can. J. Stat.* 46(1):147–75

Fearnhead P, Prangle D. 2012. Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation. *J. R. Stat. Soc. Ser. B* 74(3):419–74

Fieller E, Hartley H. 1954. Sampling with control variables. *Biometrika* 41(3/4):494–501

Filippi S, Barnes CP, Cornebise J, Stumpf MP. 2013. On optimality of kernels for approximate Bayesian computation using sequential Monte Carlo. *Stat. Appl. Genet. Mol. Biol.* 12(1):87–107

Frigessi A, Gåsemyr J, Rue H. 2000. Antithetic coupling of two Gibbs sampler chains. *Ann. Stat.* 28:1128–49

Gallant AR, McCulloch RE. 2009. On the determination of general scientific models with application to asset pricing. *J. Am. Stat. Assoc.* 104(485):117–31

Gong L, Flegal JM. 2016. A practical sequential stopping rule for high-dimensional Markov chain Monte Carlo. *J. Comput. Graph. Stat.* 25(3):684–700

Gourieroux C, Monfort A, Renault E. 1993. Indirect inference. *J. Appl. Econom.* 8(S1):S85–118

Haario H, Saksman E, Tamminen J. 2001. An adaptive Metropolis algorithm. *Bernoulli* 7:223–42

Hastings WK. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57(1):97–109

Hoffman MD, Gelman A. 2014. The no-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.* 15(1):1593–623
Huggins J, Campbell T, Broderick T. 2016. Coresets for scalable Bayesian logistic regression. In *NIPS'16: Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 4087–95. N.p.: NeurIPS
Korattikara A, Chen Y, Welling M. 2014. Austerity in MCMC land: cutting the Metropolis-Hastings budget. *PMLR* 32(1):181–89
Łatuszyński K, Rosenthal JS. 2014. The containment condition and adapfail algorithms. *J. Appl. Probab.* 51(4):1189–95
Lee A. 2012. On the choice of MCMC kernels for approximate Bayesian computation with SMC samplers. In *Proceedings of the 2012 Winter Simulation Conference (WSC)*. New York: IEEE
Lee A, Andrieu C, Doucet A. 2012. Discussion of constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation. *J. R. Stat. Soc. Ser. B* 74(3):449–50
Levi E, Craiu RV. 2022. Finding our way in the dark: approximate MCMC for approximate Bayesian methods. *Bayesian Anal.* 17(1):193–221
Lux T, Marchesi M. 2000. Volatility clustering in financial markets: a microsimulation of interacting agents. *Int. J. Theor. Appl. Finance* 3(04):675–702
Manousakas D, Xu Z, Mascolo C, Campbell T. 2020. Bayesian pseudocoresets. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, pp. 14950–60. N.p.: NeurIPS
Marin JM, Pillai NS, Robert CP, Rousseau J. 2014. Relevant statistics for Bayesian model choice. *J. R. Stat. Soc. Ser. B* 76(5):833–59
Marjoram P, Molitor J, Plagnol V, Tavaré S. 2003. Markov chain Monte Carlo without likelihoods. *PNAS* 100(26):15324–28
Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. 1953. Equation of state calculations by fast computing machines. *J. Chem. Phys.* 21(6):1087–92
Metropolis N, Ulam S. 1949. Monte Carlo method. *J. Am. Stat. Assoc.* 44(247):335–41
Neiswanger W, Wang C, Xing E. 2013. Asymptotically exact, embarrassingly parallel MCMC. arXiv:1311.4780 [stat.ML]
Nemeth C, Sherlock C. 2018. Merging MCMC subposteriors through Gaussian-process approximations. *Bayesian Anal.* 13(2):507–30
Oyebamiji OK, Edwards NR, Holden PB, Garthwaite PH, Schaphoff S, Gerten D. 2015. Emulating global climate change impacts on crop yields. *Stat. Model.* 15(6):499–525
Plumlee M, Asher TG, Chang W, Bilskie MV. 2021. High-fidelity hurricane surge forecasting using emulation and sequential experiments. *Ann. Appl. Stat.* 15(1):460–80
Polson NG, Scott JG, Windle J. 2013. Bayesian inference for logistic models using Pólya–Gamma latent variables. *J. Am. Stat. Assoc.* 108(504):1339–49
Pompe E, Holmes C, Łatuszyński K. 2020. A framework for adaptive MCMC targeting multimodal distributions. *Ann. Stat.* 48(5):2930–52
Pratola MT. 2016. Efficient Metropolis–Hastings proposal mechanisms for Bayesian regression tree models. *Bayesian Anal.* 11(3):885–911
Price LF, Drovandi CC, Lee A, Nott DJ. 2018. Bayesian synthetic likelihood. *J. Comput. Graph. Stat.* 27(1):1–11
Pritchard JK, Seielstad MT, Perez-Lezaun A, Feldman MW. 1999. Population growth of human Y chromosomes: a study of Y chromosome microsatellites. *Mol. Biol. Evol.* 16(12):1791–98
Quiroz M, Kohn R, Villani M, Tran MN. 2018. Speeding up MCMC by efficient data subsampling. *J. Am. Stat. Assoc.*
Robert CP. 2014. Bayesian computational tools. *Annu. Rev. Stat. Appl.* 1:153–77
Roberts GO, Rosenthal JS. 2009. Examples of adaptive MCMC. *J. Comput. Graph. Stat.* 18:349–67
Rosenthal JS, Yang J. 2018. Ergodicity of combocontinuous adaptive MCMC algorithms. *Methodol. Comput. Appl. Probab.* 20(2):535–51
Rue H, Riebler A, Sorbye S, Illian J, Simpson D, Lindgren F. 2017. Bayesian computing with INLA: a review. *Annu. Rev. Stat. Appl.* 4:395–421
Schmitt TA, Schäfer R, Dette H, Guhr T. 2015. Quantile correlations: uncovering temporal dependencies in financial time series. *Int. J. Theor. Appl. Finance* 18(7):1550044

Scott SL, Blocker AW, Bonassi FV, Chipman HA, George EI, McCulloch RE. 2016. Bayes and big data: the consensus Monte Carlo algorithm. *Int. J. Manag. Sci. Eng. Manag.* 11(2):78–88

Sisson SA, Fan Y, Beaumont M. 2018a. Overview of approximate Bayesian computation. arXiv:1802.09720 [stat.CO]

Sisson SA, Fan Y, Beaumont M. 2018b. *Handbook of Approximate Bayesian Computation*. Boca Raton, FL: Chapman and Hall/CRC

Sisson SA, Fan Y, Tanaka MM. 2007. Sequential Monte Carlo without likelihoods. *PNAS* 104(6):1760–65

Smith AA Jr. 1993. Estimating nonlinear time-series models using simulated vector autoregressions. *J. Appl. Econom.* 8(S1):S63–84

Tavaré S, Balding DJ, Griffiths RC, Donnelly P. 1997. Inferring coalescence times from DNA sequence data. *Genetics* 145(2):505–18

Vats D, Flegal JM, Jones GL. 2019. Multivariate output analysis for Markov chain Monte Carlo. *Biometrika* 106(2):321–37

Wang X, Dunson DB. 2013. Parallelizing MCMC via Weierstrass sampler. arXiv:1312.4605 [stat.CO]

Wood SN. 2010. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature* 466(7310):1102

Yang J, Levi E, Craiu RV, Rosenthal JS. 2019. Adaptive component-wise multiple-try Metropolis sampling. *J. Comput. Graph. Stat.* 28(2):276–89

Annual Review of Statistics and Its Application

Volume 10, 2023

Contents

Errata

An online log of corrections to *Annual Review of Statistics and Its Application* articles may be found at http://www.annualreviews.org/errata/statistics