Annual Review of Statistics and Its Application

Postprocessing of MCMC

Leah F. South,[1] Marina Riabiz,[2,3] Onur Teymur,[3,4] and Chris J. Oates[3,5]

[1] School of Mathematical Sciences, Queensland University of Technology, Brisbane, Queensland 4000, Australia; email: l1.south@qut.edu.au

[2] Department of Biomedical Engineering, King's College London, London SE1 7EH, United Kingdom

[3] Alan Turing Institute, London NW1 2DB, United Kingdom

[4] School of Mathematics, Statistics and Actuarial Science, University of Kent, Canterbury CT2 7FS, United Kingdom

[5] School of Mathematics, Statistics & Physics, Newcastle University, Newcastle upon Tyne NE1 7RU, United Kingdom

Annu. Rev. Stat. Appl. 2022. 9:529–55

First published as a Review in Advance on November 29, 2021

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

https://doi.org/10.1146/annurev-statistics-040220-091727

Keywords

bias removal, control variates, Markov chain, Monte Carlo, Stein discrepancy, thinning, variance reduction

Abstract

Markov chain Monte Carlo is the engine of modern Bayesian statistics, being used to approximate the posterior and derived quantities of interest. Despite this, the issue of how the output from a Markov chain is postprocessed and reported is often overlooked. Convergence diagnostics can be used to control bias via burn-in removal, but these do not account for (common) situations where a limited computational budget engenders a bias-variance trade-off. The aim of this article is to review state-of-the-art techniques for postprocessing Markov chain output. Our review covers methods based on discrepancy minimization, which directly address the bias-variance trade-off, as well as general-purpose control variate methods for approximating expected quantities of interest.

1. INTRODUCTION

Intractable distribution: a probability distribution whose density function is provided up to an unknown proportionality constant

The Bayesian statistical framework is operational, in the sense that a user first elicits their a priori belief and then updates their belief in light of data, in a way that is (at least in principle) uniquely prescribed. This updating is codified by Bayes' rule, which expresses parameters' posterior probability density as being proportional to the product of a priori probability density and the data likelihood. Certain combinations of a priori belief and likelihood are conjugate, meaning that the posterior can be analytically computed. Outside of the conjugate setting, computational methods are required. The computational challenge, then, is to accurately approximate an intractable probability distribution, meaning a distribution whose density function is available up to proportionality, where the normalization constant is an intractable integral.

The majority of Bayesian analyses produce a posterior that is intractable, as indeed do other statistical frameworks (such as generalized Bayesian inference; Bissiri et al. 2016). There has, accordingly, been extensive research into computational methods for approximating intractable distributions. The focus of this review is on Markov chain Monte Carlo (MCMC) methods, a large class of computational methods which, for several decades now, have been considered among the state of the art. Given an intractable distribution, one can typically find several methods in the MCMC literature that can be applied. However, the effectiveness of a particular method is not always easy to predict. Furthermore, once an MCMC method has been applied, it is not always easy to determine the quality of the approximation produced. Typical situations where these challenges occur include applications of Bayesian statistics in which the parameter space is high dimensional; applications in which the likelihood has high information content, causing the posterior to present multiple modes or concentrate on manifolds; and settings where computational complexity limits the number of evaluations of the likelihood (Brooks et al. 2011).

Postprocessing procedures aim to improve the quality of estimators that are based on MCMC output, to approximate either the probability distribution itself or a derived quantity of interest. The main practical requirement of a postprocessing procedure is that it should be agnostic to the details of the MCMC method. The best-known examples of postprocessing procedures are burn-in removal and thinning. In burn-in removal, one attempts to identify a number of iterations after which the Markov chain can (informally) be said to have converged to the parameters posterior distribution, and then removes the initial part of the output where the chain had not converged. This procedure can reduce bias in the MCMC output by reducing the dependence on how the MCMC was initialized, but it does not consider variance of the resulting estimators, which depends on the sample size and thus may be large if most of the chain is removed. In thinning, every kth iteration is retained and the remainder discarded, in order to reduce the positive correlation between the remaining states and therefore reduce the asymptotic variance of the estimators. This can facilitate compression of MCMC output but does not always improve the approximation that is produced (notoriously, thinning does not lead to an efficiency gain if the samples are only used to estimate the posterior expectation of an inexpensive function).

It is thus notable that postprocessing of MCMC engenders a bias-variance trade-off and yet standard postprocessing procedures do not attempt to address this trade-off.

This review focuses on modern postprocessing techniques that can be applied to MCMC output. Our discussion focuses on MCMC, for which consistency results have been established, but much of what we discuss is amenable to application in other computational methods that produce a collection of representative values as output, such as sequential Monte Carlo (Chopin 2002). To limit scope, our focus is principally on continuous-valued random variables, rather than discrete or categorical variables, but where possible we aim to keep discussion general. The article is structured as follows: Notation is established in Section 1.1, background on Markov chains is recalled in Section 1.2, and a formal problem statement is provided in Section 1.3. Section 2 focuses on the task of approximating the full probability distribution using MCMC output; we recall the standard approaches of burn-in removal and thinning before describing modern and powerful techniques based on discrepancy minimization in detail. In Section 3 we focus on the task of approximating one or more scalar quantities of interest. Control variate methods represent a powerful computational tool in this context, and we discuss the state of the art in control variate methodology in detail. A brief discussion concludes in Section 4.

Markov chain: an ordered sequence of random variables X_n, such that X_{n+1} is conditionally independent of $(X_m)_{m<n}$ given X_n

1.1. Notation

For this article we use $(\Omega, \mathcal{F}, \mathbb{P})$ to denote an underlying probability space on which all random variables are (often implicitly) defined and we let $\mathbb{E}[\,\cdot\,] = \int \cdot \, \mathrm{d}\mathbb{P}$. Conditional probabilities are defined in the standard sense of Kolmogorov (1956) and denoted $\mathbb{P}(F|G)$, $F, G \in \mathcal{F}$. For this article we introduce a measurable space $(\mathcal{X}, \mathcal{B})$ and consider a random variable to be a measurable function $X : \Omega \to \mathcal{X}$, whose distribution P is defined as $P(B) := \mathbb{P}(X \in B)$ for all $B \in \mathcal{B}$, where the conventional shorthand "$X \in B$" is used for the event $\{\omega \in \Omega : X(\omega) \in B\} \in \mathcal{F}$. In the Bayesian context, X represents the parameters of a statistical model and P represents the posterior distribution after data have been assimilated. Let $\mathcal{L}^2(P)$ be the vector space of random variables $f : \mathcal{X} \to \mathbb{R}$ with $\int f^2 \mathrm{d}P < \infty$. Let $\delta(x)$ denote the distribution of the random variable $f(\omega) = x$ for all $\omega \in \Omega$. For a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$, we denote the gradient of f as ∇f where $(\nabla f)(x) := (\partial_{x_1} f(x), \ldots, \partial_{x_d} f(x))^{\mathrm{T}}$. For a differentiable function $f : \mathbb{R}^d \to \mathbb{R}^d$, we denote the divergence of f as $\nabla \cdot f$ where $(\nabla \cdot f)(x) := \partial_{x_1} f_1(x) + \cdots + \partial_{x_d} f_d(x)$, and if f is twice-differentiable, we denote the Laplacian of f as Δf where $\Delta f := \nabla \cdot (\nabla f)$. Natural numbers excluding zero are denoted $\mathbb{N}$ and including zero are denoted $\mathbb{N}_0$. The vector of ones is denoted $\mathbf{1}$, the unit vector $(1, 0, \ldots, 0)^\top$ is denoted $\mathbf{e}_1$, and $\|x\|$ denotes the Euclidean distance $\sqrt{x_1^2 + \cdots + x_d^2}$.

1.2. Markov Chains

A Markov chain is a sequence $(X_n)_{n \in \mathbb{N}}$ of random variables $X_n : \Omega \to \mathcal{X}$ with the property that $X_{n+1} \perp\!\!\!\perp (X_m)_{m<n} \,|\, X_n$, where $X \perp\!\!\!\perp Y \,|\, Z$ indicates that the random variables X and Y are conditionally independent given the random variable Z.

In this article we assume a nonrandom initial state $X_0 \in \mathcal{X}$. To a Markov chain we can associate a sequence of transition kernels $P_n(x, B) := \mathbb{P}(X_n \in B | X_{n-1} = x)$, $x \in \mathcal{X}$, $B \in \mathcal{B}$, so that $P_n(x, B)$ represents the probability that the state X_n of the Markov chain takes a value in the set B, given that the previous state X_{n-1} was equal to x. The chain is said to be time-homogeneous if P_n does not depend on n. Inductively define the nth step transition kernel as $P^n(x, B) := \int P_n(y, B) P^{n-1}(x, \mathrm{d}y)$, $x \in \mathcal{X}$, $B \in \mathcal{B}$, with base case $P^0(x, B) = 1$ if $x \in B$, and 0 if $x \notin B$. That is, $P^n(x, B)$ represents the probability that the state X_n of the Markov chain takes a value in the set B, given that the initial state X_0 was equal to x. A Markov chain is said to be P-invariant if $\int P^n(x, B) \mathrm{d}P(x) = P(B)$ for all n and all $B \in \mathcal{B}$. Intuitively, if one was to randomize the initial state X_0 by sampling it from P, then the state X_n will also have distribution P if the Markov chain is P-invariant.

Loosely speaking, a P-invariant Markov chain might be described as ergodic if $P^n(x, B)$ approximates $P(B)$ in the $n \to \infty$ limit, for all $x \in \mathcal{X}$, $B \in \mathcal{B}$. Several notions of ergodicity exist in the literature, but in this article we focus on a specific notion called V-uniform ergodicity, which will now be defined. For a function $V : \mathcal{X} \to [1, \infty)$, a function $f : \mathcal{X} \to \mathbb{R}$ and a measure Q on $(\mathcal{X}, \mathcal{B})$, we denote $\|f\|_V := \sup_{x \in \mathcal{X}} |f(x)|/V(x)$, $\|Q\|_V := \sup_{\|f\|_V \leq 1} |\int f \mathrm{d}Q|$. A Markov chain is said to be V-uniformly ergodic if there exist constants $R \in [0, \infty)$, $\rho \in [0, 1)$, such that $\|P^n(x, \cdot) - P\|_V \leq$

$RV(x)\rho^n$ for all $n \in \mathbb{N}$ and all $x \in \mathcal{X}$. A comprehensive treatment of Markov chains can be found in the textbook of Meyn & Tweedie (2012).

Markov chain Monte Carlo (MCMC): an MCMC method is an algorithm that, given a distribution Q, constructs a Markov chain that is Q-invariant

MCMC output: a single realization (or sample path) of a Markov chain, of which the first N states are provided

1.3. Problem Statement

Consider an intractable probability distribution P. Our aim is to compute an approximation, either to the distribution P itself (Section 2) or to derived scalar quantities of interest (Section 3). Our starting point is one[1] realization (i.e., based on one random seed $\omega \sim \mathbb{P}$), of a finite portion $(X_n)_{n \leq N}$ of the Markov chain,[2] which we call the MCMC output. It is not assumed that the Markov chain is P-invariant unless stated, and later we discuss how output from an MCMC method that is Q-invariant may nevertheless enable P to be consistently approximated if Q is not too dissimilar to P. All approximations are to be constructed by postprocessing the MCMC output. In other words, we may only consider properties of P defined locally at the states X_n and no further exploration of $\mathcal{X}$ outside this finite set is permitted. In particular, we exclude the trivial solutions of simply running further iterations of MCMC or adopting a different, possibly better MCMC method. This setup is realistic, reflecting the scenario that a practitioner has invested considerable resources into producing MCMC output and wishes to employ postprocessing techniques to extract as much value as possible from their investment.

An important preliminary comment is that the postprocessing techniques described in this article (and, indeed, most MCMC methods) are not parameterization invariant. This means that, if one were to apply an invertible transformation $Y_n = y(X_n)$, then postprocessing of the MCMC output $(Y_n)_{n \leq N}$ can lead to different conclusions compared with if $(X_n)_{n \leq N}$ had been postprocessed. To limit scope we do not discuss parameterization in this article. Instead, following standard practice, we presuppose that one has employed transformation(s), such as centering and scaling (Yu & Meng 2011), that (loosely speaking) promote simplicity, in order that P can be more easily approximated.

2. APPROXIMATION OF THE POSTERIOR DISTRIBUTION

The outcome of an exploratory Bayesian analysis is the posterior distribution itself, expressing a posteriori belief about unknown parameters on the basis of a priori belief and evidence provided by the dataset. To facilitate exploratory Bayesian analysis outside the conjugate setting, it is therefore important that the entire posterior distribution can be accurately approximated. This section studies how MCMC output can be used to produce an approximation to a distribution P of interest. Throughout, we consider approximations of the form

$$\sum_{i=1}^{M} w_i \delta(X_{\pi(i)}), \qquad 1.$$

where $w_1, \ldots, w_M \in \mathbb{R}$ are weights satisfying $\sum_{i=1}^{M} w_i = 1$ and $\pi\colon \{1, \ldots, M\} \to \{1, \ldots, N\}$ is a function that indicates which states from the MCMC output are included in Equation 1. In simple terms, this approximation extracts and reweights a subsequence of length M from the given MCMC output of length N.

[1] In general applications it is common to exploit multi-core central processing units to simulate independent Markov chains in parallel. However, in the most challenging applications (where postprocessing is most important), it is common to have access to only one MCMC output. This article considers postprocessing of one MCMC output, but many of the methods we discuss can be trivially applied to aggregated MCMC output.

[2] To limit scope, so-called adaptive MCMC, which aims to identify a suitable Markov transition kernel on the fly, is not discussed. However, most of our presentation applies also to adaptive MCMC output.

Recall the two categories of postprocessing discussed in Section 1. First, if the chain is constructed so that its asymptotic law converges to P, then excluding the first b points (the burn-in period) from Equation 1 may help to reduce bias due to the choice of the initial state X_0 of the Markov chain. This corresponds to excluding $\{1, \ldots, b\}$ from the image of π, and we discuss standard approaches to this problem in Section 2.1. Second, thinning of MCMC output can be useful when samples are to be used for further computation, especially when the subsequent computation has a high cost. This corresponds to excluding i from the image of π whenever $i \neq 1$ modulo k, and we briefly discuss approaches to thinning in Section 2.3. In both cases, uniform weights $w_i = \frac{1}{M}$ are assumed in Equation 1.

Postprocessing MCMC output: selecting a weighted combination of states from the MCMC output to better represent the posterior distribution P

Burn-in: the first b states of a P-invariant Markov chain, for which the distribution of X_n, $n \leq b$, is deemed to substantially differ from P

2.1. Burn-In Removal

In this section we discuss standard approaches to identification of a burn-in period from given MCMC output, in order to control the bias resulting from an arbitrary choice of initial state X_0 for the Markov chain. Our focus is limited to continuous domains $\mathcal{X} \subseteq \mathbb{R}^d$. Rigorous approaches for selecting a burn-in period b have been proposed by authors including Meyn & Tweedie (1994), Rosenthal (1995), and Roberts & Tweedie (1999) (see also Jones & Hobert 2001). Unfortunately, these often involve conditions that are difficult to establish or, when they hold, they may provide loose bounds on the total variation distance between the law of the Markov chain and the invariant distribution, implying an unreasonably long burn-in period. More recently Biswas et al. (2019) discuss how to estimate such bounds through coupling and multiple MCMC runs, but this is out of the scope considered here, where a single MCMC run has been obtained at moderate to high computing cost. Convergence diagnostics have emerged as a practical solution to the need to test for nonconvergence of MCMC. Their use is limited to reducing bias in MCMC output; they are not designed for the setting that we consider, where the length N of the MCMC output is fixed, and which requires a bias-variance trade-off. Nevertheless, convergence diagnostics constitute the most common means by which MCMC output is postprocessed in modern software packages for MCMC, including WinBUGS (Lunn et al. 2000), JAGS (Plummer 2003), R (R Core Team 2020), Stan (Carpenter et al. 2017), and `PyMC3` (Salvatier et al. 2016).

In this section we recall standard practice for selection of a burn-in period b, and thus (implicitly) in constructing an estimator of the form in Equation 1, focusing on the traditional $\widehat{R}$ statistic of Gelman & Rubin (1992) and Brooks & Gelman (1998). The aim of this section is to describe the general idea and fundamental limitations of convergence diagnostics in the fixed N scenario, rather than presenting the state of the art or providing a comprehensive survey of convergence diagnostics for burn-in removal. We simply recall that the $\widehat{R}$ convergence diagnostic was first introduced by Gelman & Rubin (1992) and subsequently corrected by Brooks & Gelman (1998), and this was then simplified by Gelman et al. (2003). We use the implementation of the Brooks & Gelman (1998) version from the R package `coda` in our experiments, and we focus on the simple expression of Gelman et al. (2003) in the text. Further developments of the $\widehat{R}$ convergence diagnostic include those of Gelman et al. (2013), where the diagnostic test is performed separately on each half of the MCMC output; Vats & Knudson (2018), which revisits a connection between $\widehat{R}$ and effective sample size[3] of quantities of interest estimated from the MCMC output; and Vehtari et al. (2021), which provides more details on such connections, addressing also target distributions

[3] The effective sample size indicates how many independent samples are needed to provide the same amount of information about that quantity of interest as the correlated MCMC output: The higher this value, the lower the loss of information due to correlation in the MCMC output.

Sample mean: the sample mean m_l of an MCMC output $(X_n^l)_{n\leq N}$ is defined as $\frac{1}{N}\sum_{n=1}^{N} X_n^l$

Sample variance: the sample variance s_l^2 of an MCMC output $(X_n^l)_{n\leq N}$ is defined as $\frac{1}{N-1}\sum_{n=1}^{N}(X_n^l - m_l)^2$

with infinite variance, and the case in which the Markov chain is exploring the bulk of the target distribution, but not its tails. A comprehensive survey of convergence diagnostics for MCMC is provided by Roy (2020).

The traditional $\widehat{R}$ statistic of Gelman et al. (2003) is not a postprocessing method in the strict sense set out in Section 1.3 because it is based on $l = 1, \ldots, L$ independent realizations of MCMC output, $(X_n^l)_{n\leq N}$; i.e., X_n^l denotes the random variable X_n evaluated at ω_l, where $\omega_1, \ldots, \omega_L \sim \mathbb{P}$ are independent. For a uni-dimensional target distribution, the traditional $\widehat{R}$ statistic is defined as the square root of the ratio of two estimators of the variance σ^2 of the target:

$$\widehat{R} := \sqrt{\frac{\hat{\sigma}^2}{s^2}},$$

where s^2 is the (arithmetic) mean of the sample variances s_l^2 along the L sample paths $(X_n^l)_{n\leq N}$, and it typically provides an underestimate of σ^2. Meanwhile, $\hat{\sigma}^2$ is constructed as an unbiased overestimate of the target variance

$$\hat{\sigma}^2 := \frac{N-1}{N}s^2 + \frac{1}{L-1}\sum_{l=1}^{L}\left(m_l - \frac{1}{L}\sum_{l'=1}^{L} m_l\right)^2,$$

where m_l is the sample mean from the lth sample path, and where the second term is the sample variance of the sample means from L chains.

For an ergodic Markov chain, $\widehat{R}$ converges to 1 as $N \to \infty$. In practice, it is common to discard a burn-in period of length $b = N$, where N is the smallest integer for which $\widehat{R} < 1 + \delta$, and δ is a suitable threshold.[4] The somewhat arbitrary choice of $\delta = 0.1$ has historically been used (Gelman et al. 2013), and current best practice for traditional $\widehat{R}$ and its extensions advocates $\delta = 0.01$ (Vehtari et al. 2021).

Convergence diagnostics can help to detect situations in which a Markov chain has not converged, and for this purpose they are widely used. Their main drawbacks are that (*a*) such diagnostics do not provide guarantees that the Markov chain has actually converged (existing convergence diagnostics can assess only necessary but not sufficient conditions for convergence), and (*b*) burn-in removal may not be useful in practical settings where the MCMC output has already been obtained and postprocessing is required, as described in Section 1.3. In order to mitigate the first point, Vats & Knudson (2018) and Vehtari et al. (2021) recommend to look at the effective sample size of quantities of interest, if possible combining autocorrelation information from multiple chains, which helps to detect poor convergence in cases of multimodal target distributions. However, this still remains only a necessary, not sufficient, condition for convergence, and it does not help tackling the second point. This section ends with an example to highlight this important second drawback of convergence diagnostics:

> **Example 1 (Burn-in removal lacks a bias-variance trade-off).** The purpose of convergence diagnostics is to detect and avoid bias due to dependence on the arbitrary choice of initial state X_0. However, burn-in removal does not address the bias-variance trade-off that occurs when the MCMC output is fixed. As an extreme illustration of this, consider the MCMC output shown in **Figure 1**. Here, $L = 6$ independent sample paths of total length $N = 10^3$ were produced using random walk Metropolis–Hastings (Metropolis et al. 1953). A simple bivariate target P, whose contour lines are plotted in red, was used, but the Markov chain was not optimized, to simulate

[4] Although seeking the smallest N is not an explicit recommendation in the literature cited, it is clear that one would not want to simulate a Markov chain for longer than required. Thus, in effect, it is standard practice to work with N as small as possible, subject to the diagnostic test being passed.

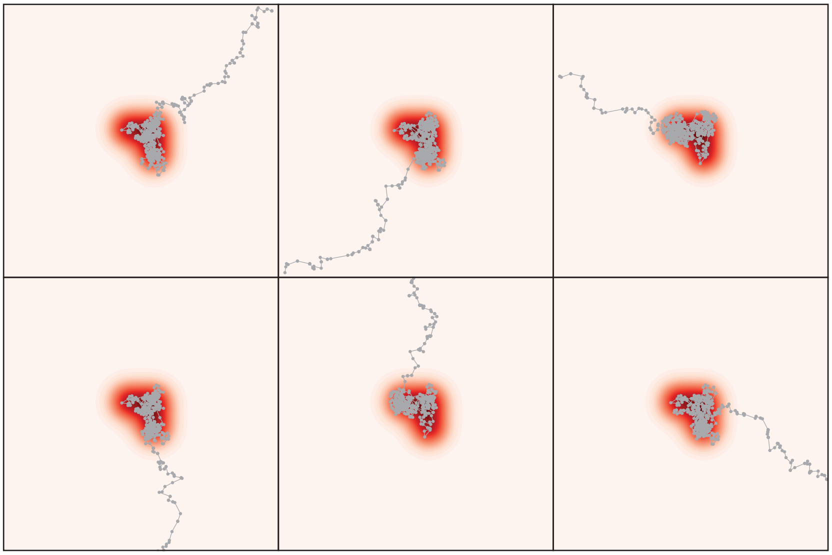

Figure 1

MCMC output. Here we show $L = 6$ independent realizations of MCMC output (*gray lines*) for a particular bivariate distributional target P indicated by the shaded contour plot in the background. In each case a total of $N = 10^3$ iterations of the Markov chain were performed, with the first 500 iterations plotted. Abbreviation: MCMC, Markov chain Monte Carlo.

a challenging sampling context. The initial states X_0^l were overdispersed relative to the target P, requiring the Markov chain to take several steps before the high-probability region is reached. **Figure 2** applies convergence diagnostics to establish whether or not the Markov chains can be said to have converged.

The traditional $\widehat{R}$ statistic of Brooks & Gelman (1998) detects nonconvergence even after all $N = 10^3$ iterations of the MCMC output have been considered, irrespective of whether the diagnostics are applied to each coordinate of the state vector or jointly to both coordinates. This is undesirable from our perspective of postprocessing MCMC output, since it is clear from **Figure 1** that there is useful information in the MCMC output, even if some dependence on X_0 can be detected. In addition, we present in **Figure 2** two of many proposed improvements over Brooks & Gelman (1998): the recent diagnostic due to Vats & Knudson (2018) and also a version of such convergence diagnostic presented in the same work, which can be computed using a single MCMC output. They indicate that the burn-in period has finished, but they leave only a small portion of the chain after the burn-in, when considering the threshold $\delta = 0.01$. All convergence diagnostics were computed using the R packages `coda` (Plummer et al. 2006) and `stableGR` (Knudson & Vats 2020).

The modern MCMC postprocessing techniques presented in Section 2.3 address this bias-variance trade-off, and their use is encouraged in problems where obtaining further MCMC iterations is not practical.

2.2. Fixed Frequency Thinning

As with the classical approaches to burn-in removal discussed in Section 2.1, thinning is often performed on a heuristic basis as the simplest way to achieve compression of MCMC output. In exploratory Bayesian analysis, this is often motivated by the need to reduce storage cost or to make subsequent computation faster (the reader can, for example, think about the case in which the samples obtained from the posterior are used for forward uncertainty propagation in complex

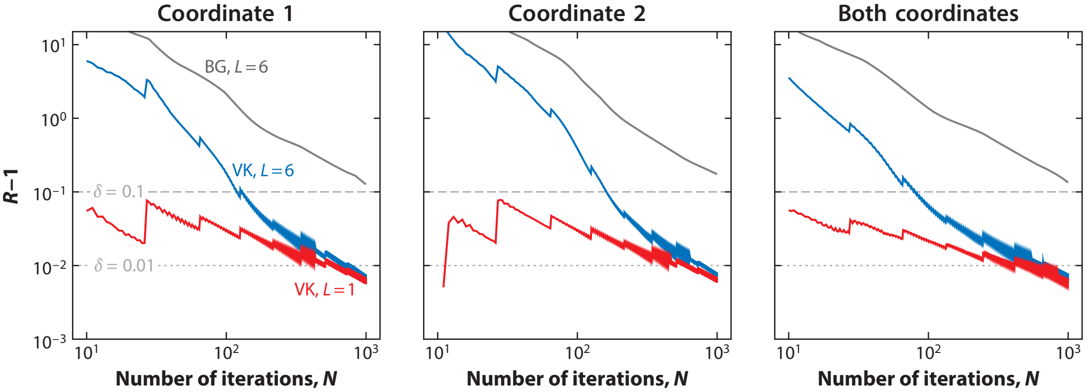

Figure 2

Convergence diagnostics for the Markov chain Monte Carlo (MCMC) output shown in **Figure 1**. Here we show the traditional $\widehat{R}$ statistic of Brooks & Gelman (1998) (BG; *gray solid lines*) and also an autocorrelation-based diagnostic used by Vats & Knudson (2018) (VK; *blue and red solid lines*), each as a function of the number of iterations N of the Markov chain that are considered. The blue lines indicate the Vats & Knudson (2018) diagnostic, and the red lines indicate a version of the diagnostic that can be computed using a single MCMC output. These diagnostics were applied separately to the first and second coordinates of the bivariate state variable (*left, center panels*) and jointly to both coordinates (*right panel*). Dashed lines indicate thresholds at which convergence is deemed to have occurred. In all cases, the traditional $\widehat{R}$ statistic does not fall below the thresholds, indicating that convergence has not occurred.

multi-scale models). However, thinning is traditionally performed with no specific aim to improve the accuracy of the MCMC output.

The most common approach to thinning is to subsample with a fixed frequency from the chain (retain every kth sample and discard the rest), which can be an effective strategy to reduce autocorrelation in the MCMC output. Systematic approaches for determining k do exist, and the most well-known is based on the autocorrelation estimator of Geyer (1992), which can be computed using the R package `LaplacesDemon` (Statisticat LLC 2021). This method estimates a sequence of fixed-lag autocorrelations in the Markov chain and then thresholds this sequence to give a k that results in a subsample that is close to uncorrelated. This procedure is most useful in exploratory Bayesian analysis, where a set of such samples are themselves required, rather than as an attempt to improve an estimator. Owen (2017) provides a discussion on the statistical efficiency of this approach.

More sophisticated approaches to compress MCMC have been explored by authors including Paige et al. (2016) and Mak & Joseph (2018). In both cases, the authors aimed to construct an approximation of the form in Equation 1 with $M \ll N$, such that Equation 1 provides an accurate approximation to the discrete distribution $\frac{1}{N}\sum_{n=1}^{N}\delta(X_n)$ supported on the original MCMC output. Although these can provide effective compression of MCMC output, if the Markov chain has not converged then the compressed output will be biased. In the next section, we discuss an approach that is simultaneously capable of thinning and debiasing MCMC output and is applicable even in cases where the Markov chain is not P-invariant.

Discrepancy: a discrepancy D is a nonnegative function where $D(P, Q)$ is interpreted as the dissimilarity between measures P and Q

2.3. Discrepancy Minimization

Here, we discuss modern and powerful approaches to postprocessing of MCMC that aim to directly address the bias-variance trade-off just described. The approach we explore casts the choice of π in Equation 1 as an optimization problem. The key idea is to identify an appropriate quantification of the discrepancy between the discrete distribution Q_M in Equation 1 and the distributional target P, and then to select both the weights w_i and the index sequence π so that this discrepancy is minimized. A discrepancy is defined as a bivariate function D such that $D(P, P) = 0$ for all distributions P, and $D(P, Q) > 0$ for all $P \neq Q$. There are an infinitude of functions D that satisfy these relations, so for a discrepancy to be useful we typically require several other properties. An important property, which we do not discuss further in this article since it is often satisfied, is that $D(P, Q_M) \to 0$ whenever Q_M converges to P in an appropriate sense. Another important property, which is the converse of the property just described and which we do discuss, is called convergence control, where $D(P, Q_M) \to 0$ implies that Q_M converges to P in a sense that must be specified.

As an aside, we note that many related strands of work exploit discrepancy to approximate a distributional target P by a discrete distribution Q^N. For example, in quasi–Monte Carlo, an explicit construction $Q^N = \frac{1}{N}\sum_{n=1}^{N}\delta(x_n)$ is sought to approximate P in such a way that a discrepancy $D(P, Q^N)$ is provably asymptotically minimized (Hickernell 1998, Dick & Pillichshammer 2010). In a different direction, other researchers have implicitly constructed point sets by performing direct minimization of $D(P, Q^N)$ over the high-dimensional joint space $(x_1, \ldots, x_N) \in \mathcal{X}^N$ (see the surveys by Briol et al. 2017, 2019; Oettershagen 2017; Pronzato & Zhigljavsky 2020). In both of these methods the goal is to find a compressed representation of P, and as a starting point it is assumed that P is known in full. This is not the case when one has to postprocess MCMC output. An approach considered by Kyriazopoulou-Panagiotopoulou et al. (2008) and Kontoyiannis & Meyn (2008) is to adjust estimates based on the discrepancy between the expectation under P and the expectation under Q^N of a reference function, although this discrepancy is not convergence-determining for a finite number of reference functions. To overcome these problems, a specialized family of discrepancies are required, and these are introduced next.

Convergence control: a discrepancy D is said to have convergence control if $D(P, Q_M) \to 0$ implies Q_M converges to P in a sense that must be specified

Stein discrepancy: a discrepancy D such that $D(P, Q)$ can be computed when P is an intractable distribution and Q has a finite support

Stein thinning: an algorithm that thins MCMC output such that the Stein discrepancy between resulting empirical distribution and target distribution is minimized

2.3.1. Stein discrepancy minimization.

Our aim is to select an appropriate discrepancy D for postprocessing of MCMC. To this end, we focus on Stein discrepancy, and in particular a kernel Stein discrepancy constructed for the case where the domain $\mathcal{X}$ is $\mathbb{R}^d$ (see the sidebars titled Stein Discrepancy, Tail Condition for Stein Discrepancy, and Kernel Stein Discrepancy for definitions and detail). The main computational requirement when using Stein discrepancy is the evaluation of the gradient $\nabla \log p$ along the MCMC sample path, where p is a density function for P. Note that gradient-based samplers, such as the Metropolis-adjusted Langevin algorithm (Roberts & Stramer 2002) or Hamiltonian Monte Carlo (Duane et al. 1987), produce the required evaluations as a byproduct. Stein discrepancy is particularly well suited to postprocessing of such MCMC output since, under appropriate technical assumptions, it (*a*) allows explicit computation of $D(P, Q_M)$ and (*b*) provides convergence control, meaning in this context that $D(P, Q_M) \to 0$ implies Q_M converges weakly to P.

The optimization problem we are interested in thus reduces to the problem of identifying weights w_i and an index sequence π for which the kernel Stein discrepancy $D(P, Q_M)$ is minimized when Q_M is the discrete distribution in Equation 1. Approaches based on Stein discrepancy minimization include black-box importance sampling (Liu & Lee 2017, Hodgkinson et al. 2020), Stein points (Chen et al. 2018, 2019), and Stein thinning (Riabiz et al. 2021). In what follows we describe the Stein thinning approach of Riabiz et al. (2021), illustrated in **Figure 3** in the setting with equal weights $w_i = \frac{1}{M}$, and defer to Liu & Lee (2017), Hodgkinson et al. (2020), and Riabiz et al. (2021) for discussion of the case in which weights are not equal.

Combinatorial optimization to elicit an index sequence π for which the kernel Stein discrepancy $D(P, Q_M)$ is minimized presents some technical challenges, which we defer discussion of until Section 2.3.2. Here we describe the simple, sequential approach called Stein thinning that was explored by Riabiz et al. (2021). This involves constructing π in a sequential, greedy manner, in

STEIN DISCREPANCY

A Stein discrepancy is a discrepancy of the form

$$D(P,Q) = \sup_{f \in \mathcal{F}_P} \left| \int f \mathrm{d}Q \right|, \tag{2.}$$

where $\mathcal{F}_P$ is a set of functions chosen to satisfy $\int f \mathrm{d}P = 0$. For a sufficiently large set $\mathcal{F}_P$, it is possible to have $D(P, Q) = 0$ imply $P = Q$. One way of achieving this is by taking $\mathcal{F}_P$ to be the set of functions of the form $f(x) = h(x) - \int h \mathrm{d}P$, with h ranging over a measure-determining set $\mathcal{H}$ (such a D is recognized as an integral probability metric; see Müller 1997). However, for intractable P the presence of the integral $\int h \mathrm{d}P$ renders this choice impractical.

Building on Stein (1972), the recent work of Gorham & Mackey (2015) proposed an alternative approach that can be applied provided that P admits a positive and differentiable density on $\mathcal{X} = \mathbb{R}^d$. Let the set $\mathcal{F}_P$ be composed of functions of the form

$$f(x) = (\mathcal{A}_P h)(x) := (\nabla \cdot h)(x) + (\nabla \log p)(x) \cdot h(x), \tag{3.}$$

where h ranges over a sufficiently large set $\mathcal{H}$ of differentiable functions $h : \mathbb{R}^d \to \mathbb{R}^d$. The differential operator $\mathcal{A}_P$ is called a Stein operator, and the set $\mathcal{F}_P$ is called a Stein set. Under a particular tail condition on h (see the sidebar titled Tail Condition for Stein Discrepancy) it can be shown that $\int f \mathrm{d}P = 0$. With further regularity assumptions, it can be shown that such a Stein discrepancy can enjoy either Wasserstein convergence control (see Gorham & Mackey 2015, theorem 2) or weak convergence control (see Gorham & Mackey 2017, theorem 8), depending on how the set $\mathcal{H}$ is selected. Alternative Stein operators are also possible (see Gorham et al. 2019).

which at iteration $1 \leq j \leq M$, an index $\pi(j)$ is selected according to

$$\pi(j) \in \operatorname*{argmin}_{i \in \{1,\ldots,N\}} D\left(P, \frac{1}{j} \left[\delta(X_i) + \sum_{j'=1}^{j-1} \delta(X_{\pi(j')}) \right] \right) \tag{5.}$$

TAIL CONDITION FOR STEIN DISCREPANCY

To construct a Stein discrepancy, we require a set of functions $h : \mathbb{R}^d \to \mathbb{R}^d$ for which the Stein identity $\int (\mathcal{A}_P h) \mathrm{d}P = 0$ holds, with $\mathcal{A}_P$ defined in Equation 3. This can be formulated as a tail condition on h. The main idea is to recognize $\mathcal{A}_P h$ as a divergence operator and exploit the divergence theorem over a ball $B(r)$ of radius $r > 0$, centered at the origin in $\mathbb{R}^d$:

$$\int (\mathcal{A}_P h) \mathrm{d}P = \int \frac{1}{p} \nabla \cdot (ph) \mathrm{d}P = \int \nabla \cdot (ph) \mathrm{d}V = \lim_{r \to \infty} \oint_{B(r)} ph \cdot n \mathrm{d}\sigma.$$

Here, $\mathrm{d}V$ denotes the volume element in $B(r)$, $\mathrm{d}\sigma$ denotes the surface area element on the boundary of $B(r)$, and n denotes the unit normal to the boundary of $B(r)$. In order for this final term to vanish, it is sufficient that $\|ph \cdot n\|$ vanishes uniformly with respect to the surface area $\oint_{B(r)} \mathrm{d}\sigma$, which is $O(r^d)$. Thus, if $h : \mathbb{R}^d \to \mathbb{R}^d$ and $\log p : \mathbb{R}^d \to \mathbb{R}$ are both continuously differentiable and the tail condition

$$\|h(x)\| \leq C \|x\|^{-\delta} p(x)^{-1} \tag{4.}$$

is satisfied for some $C \in \mathbb{R}$, some $\delta > d - 1$, and all $x \in \mathbb{R}^d$ outside of a bounded set, then the Stein identity is satisfied (South et al. 2021).

KERNEL STEIN DISCREPANCY

To facilitate computation of the supremum in Equation 2, one can specialize to a particular form of Stein discrepancy called kernel Stein discrepancy. A kernel is a symmetric, positive-definite function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. A kernel k reproduces a Hilbert space, denoted $\mathcal{H}(k)$, whose inner product is denoted $\langle \cdot, \cdot \rangle_{\mathcal{H}(k)}$. This means that the elements of $\mathcal{H}(k)$ are functions $f : \mathcal{X} \to \mathbb{R}$, and it holds that (*a*) $k(\cdot, x) \in \mathcal{H}(k)$ for all $x \in \mathcal{X}$ and (*b*) $\langle f, k(\cdot, x) \rangle_{\mathcal{H}(k)} = f(x)$ for all $x \in \mathcal{X}$, $f \in \mathcal{H}(k)$. For example, the Gaussian kernel $k(x, y) = \exp(-(x-y)^2)$ reproduces a Hilbert space that contains functions of the form $f(x) = \sum_{i=1}^{m} w_i \exp(-(x - y_i)^2)$ for all $w_i \in \mathbb{R}$, $y_i \in \mathbb{R}$, $m \in \mathbb{N}$, as well as certain limits of such functions (Berlinet & Thomas-Agnan 2011).

The main observation here is that if we take the set $\mathcal{H} := \{h : \mathbb{R}^d \to \mathbb{R}^d : \sum_{i=1}^{d} \langle h_i, h_i \rangle_{\mathcal{H}(k)} \leq 1\}$, then the supremum in Equation 2 can be exactly evaluated. Let $\mathcal{H}(k)^d := \mathcal{H}(k) \times \cdots \times \mathcal{H}(k)$ denote the Cartesian product; i.e., the elements of $\mathcal{H}(k)^d$ are functions $h : \mathcal{X} \to \mathbb{R}^d$ with components $h_i \in \mathcal{H}(k)$. Then, Oates et al. (2017, theorem 1) showed that the set of functions of the form $\mathcal{A}_P h$, $h \in \mathcal{H}(k)^d$ is a Hilbert space reproduced by the kernel

$$k_P(x, y) := \nabla_x \cdot \nabla_y k(x, y) + \nabla_x k(x, y) \cdot u(y) + \nabla_y k(x, y) \cdot u(x) + k(x, y) u(x) \cdot u(y),$$

where $u(x) := \nabla \log p(x)$. Assuming that, for each $\mathcal{A}_P h$, $h \in \mathcal{H}(k)^d$, the tail condition Equation 4 is satisfied, then following Liu et al. (2016) and Chwialkowski et al. (2016), one can show that

$$D\left(P, \sum_{i=1}^{n} w_i \delta(x_i)\right)^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j k_P(x_i, x_j)$$

for all $w_i \in \mathbb{R}$, $x_i \in \mathbb{R}^d$, $n \in \mathbb{N}$. The kernel Stein discrepancy in this equation can therefore be exactly computed whenever the gradient $\nabla \log p$ can be evaluated. Furthermore, under certain conditions, the kernel Stein discrepancy provides weak convergence control (Gorham & Mackey 2017, theorem 8).

or, equivalently, using the explicit form of kernel Stein discrepancy in the second equation in the sidebar titled Kernel Stein Discrepancy,

$$\pi(j) \in \operatorname*{argmin}_{i \in \{1, \ldots, N\}} \quad \frac{k_P(X_i, X_i)}{2} + \sum_{j'=1}^{j-1} k_P(X_{\pi(j')}, X_i). \qquad 6.$$

This procedure has computational complexity $\mathcal{O}(NM^2)$, or possibly less (since it is possible for a state to be repeatedly selected and the relevant quantities to be cached).

The main conceptual advantages of Stein thinning and related algorithms, compared with the standard postprocessing techniques described in Sections 2.1 and 2.2, are that (*a*) they directly address the bias-variance trade-off, (*b*) they can correct for systematic bias in the MCMC output, and (*c*) they can automatically identify and remove a burn-in period. The main practical limitation of Stein thinning and related algorithms is that there are certain pathologies of Stein discrepancy, which occur when either (*a*) P has distant high-probability regions or (*b*) P is high-dimensional (e.g., $d > 100$), either of which can lead to poor approximations when M is small (see Wenliang 2020). To illustrate the potential advantages of Stein thinning, we now present a special case of theorem 3 of Riabiz et al. (2021), which describes conditions under which the sequence generated using the Stein thinning algorithm in Equation 6 produces a discrete approximation Q_M that converges almost surely to P. Note in particular that the result does not assume that the Markov chain is P-invariant.

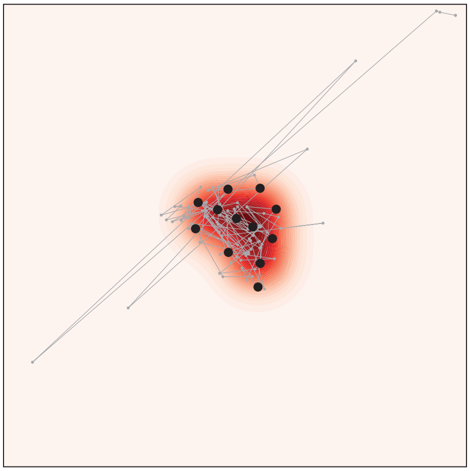

Figure 3

Postprocessing of MCMC output via Stein thinning. Here, a Markov chain sample path (*gray line*) is postprocessed to select $M = 12$ representative states (*black circles*), such that the discrete measure supported on these representative states provides an accurate approximation to the same distributional target P as in **Figure 1**, indicated by the shaded contour plot in the background.

> **Theorem 1 (Bias correction for MCMC).** Let P and P' be probability distributions with positive and continuous densities p and p' on $\mathbb{R}^d$. Assume that the tails of P are distantly dissipative (a relaxation of log concavity; see Gorham et al. 2019) and that p is continuously differentiable on $\mathbb{R}^d$. Consider a P'-invariant, time-homogeneous Markov chain $(X_i)_{i\in\mathbb{N}}$ generated using a V-uniformly ergodic transition kernel, where $V(x) = \frac{p(x)}{p'(x)}\sqrt{d + \|\nabla \log p\|^2}$. Suppose that, for some $\gamma > 0$, the following moment condition is satisfied:
>
> $$\sup_{i\in\mathbb{N}} \mathbb{E}\left[\exp\left(\gamma \max\left(1, \frac{p(X_i)}{p'(X_i)}\right)^2 \left(d + \|\nabla \log p(X_i)\|^2\right)\right)\right] < \infty. \qquad 7.$$
>
> Let π be an index sequence of length m produced by Equation 6 applied to the MCMC output $(X_n)_{n\leq N}$, where k_P in the first equation in the sidebar titled Kernel Stein Discrepancy is based on the inverse multi-quadric kernel $k(x, y) = (1 + \|x - y\|^2)^{-1/2}$. If $M \leq N$ and the growth of N is limited to at most $\log(N) = O(M^{\beta/2})$ for some $\beta < 1$, then, $\mathbb{P}$-almost surely, $Q_M = \frac{1}{M}\sum_{j=1}^{M} \delta(X_{\pi(j)})$ converges weakly to P as $M, N \to \infty$.

This result, and the related results in Liu & Lee (2017) and Hodgkinson et al. (2020), weaken or remove the requirement to design Markov chains that are exactly P-invariant (see also Gramacy et al. 2010, Radivojević & Akhmatskaya 2020). Less formally, this result suggests that one may not need to run a Markov chain to convergence in order for its output to be useful. That said, the moment condition in Equation 7 imposes a requirement that P' cannot be too dissimilar to P (informally, the Markov chain must explore the high-density regions of P, albeit not necessarily with the same frequencies as would be expected if the chain was P-invariant). This is a recent line of research, so it remains to be seen whether these advances in the postprocessing of MCMC output will in turn influence the design of algorithms for MCMC. Software to perform Stein thinning, including packages for R, Python, and MATLAB, is available at **http://stein-thinning.org/**.

2.3.2. Extensions to Stein discrepancy minimization. The Stein thinning algorithm that we just described in Equation 6 is myopic, in that it selects the index of the single best state $\pi(j)$ at each iteration without consideration of whether this makes subsequent choices better or worse

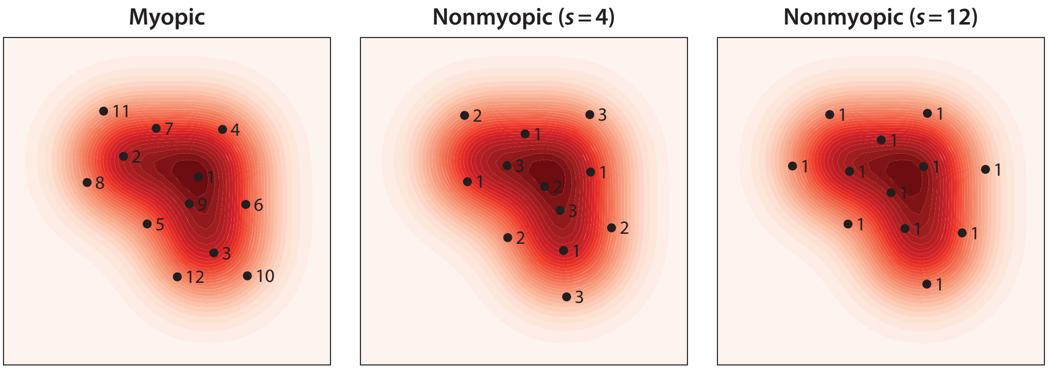

Figure 4

Extensions to Stein thinning. Here, the sample path (*gray line* in **Figure 3**) is postprocessed to select $M = 12$ representative states (*black circles*) to provide an approximation to the distributional target P, indicated by the shaded contour plot in the background. (*Left*) Myopic selection of states (look-ahead horizon $s = 1$). (*Center*) Nonmyopic selection of states, with a look-ahead horizon of $s = 4$. (*Right*) Nonmyopic selection of states, with a look-ahead horizon of $s = 12$. Integers indicate the iteration of the algorithm in which a given state was selected.

overall. This myopia can make the optimization statistically inefficient, as observed in **Figure 4** (*left panel*). Specifically, we see that the choice of the first state as the sample closest to the global mode of the distribution means that all possible choices for the second state will (temporarily) significantly worsen the overall approximation. In a less extreme fashion, this can also be seen for state 6, and state 8, by similar symmetry observations. A second shortcoming of the algorithm we just presented is that it requires scanning through the entire MCMC output of size N at each iteration, which can lead to an unacceptable computational cost.

To ameliorate these shortcomings, at least to an extent, the Stein thinning algorithm can be generalized to both nonmyopic and mini-batch settings, as described by Teymur et al. (2021). The first of these extensions involves selecting multiple points simultaneously, while the mini-batch extension considers, at each iteration, selecting points from a random subset of the samples along the MCMC path. Teymur et al. (2021) showed that these two extensions are synergistic, in that nonmyopic optimization is most useful in the mini-batch context. These extensions of Stein thinning are now described. Let $B \ll N$ be a mini-batch size and let $(X_b^j)_{1\le b\le B, 1\le j\le M}$ be the collection of mini-batches, each of size B and to be used at iteration j. For example, the mini-batches could be chosen uniformly, with or without replacement, from the MCMC output $(X_n)_{n\le N}$. Let $s \in \mathbb{N}$ be the look-ahead horizon, meaning the number of points to be simultaneously selected (the algorithm in Equation 6 corresponds to $s = 1$). Then, we choose a vector $\pi(j, \cdot)$ of s indices to be used at iteration j by performing the optimization

$$\pi(j,\cdot) \in \underset{S\in\{1,\ldots,B\}^s}{\operatorname{argmin}} \left[\frac{1}{2}\sum_{b,b'\in S} k_P(X_b^j, X_{b'}^j) + \sum_{j'=1}^{j-1}\sum_{b=1}^{s}\sum_{b'\in S} k_P\left(X^{j'}_{\pi(j',b)}, X^j_{b'}\right)\right], \qquad 8.$$

which we have obtained using the explicit form of kernel Stein discrepancy in the second equation in the sidebar titled Kernel Stein Discrepancy, in a similar manner to how we obtained Equation 6. Run for M iterations, this algorithm selects a representative set of sM states, potentially with some states selected more than once. It is possible to apply a similar theoretical analysis to that in Section 2.3.1 to the generalized algorithm in Equation 8 (see Teymur et al. 2021).

Implementation of this nonmyopic algorithm requires solving the optimization problem in Equation 8. This is a potentially challenging combinatorial optimization problem and is only

Nonmyopic: an optimization algorithm is nonmyopic if it looks further than a single step ahead when deciding the best course of action at a given iteration

Mini-batch: an approach in which the set of candidate samples is parceled up and the selection algorithm is applied separately to subsets

Integer quadratic program: an optimization problem in which the objective function is quadratic and where the solutions are constrained to be integer-valued

NP-hard: a class of problems that, informally, cannot be solved in polynomial time

tractable when the batch size B is small (e.g., $B \leq 1{,}000$). In order to solve it, one can represent the indices $S \subset \{1, \ldots, B\}^s$ of the s points to be selected at iteration j as a vector $v^j \in \mathbb{N}_0^s := \{0, \ldots, s\}^B$ whose ith element indicates the number of copies of X_i^j that are selected from the jth mini-batch, where v^j is constrained to satisfy $\sum_{i=1}^B v_i^j = s$. It is then an algebraic exercise to recast an optimal index sequence $\pi(j, \cdot)$ as the solution to a constrained integer quadratic program (e.g., Wolsey 2020):

$$\begin{gathered}\operatorname*{argmin}_{v^j \in \mathbb{N}_0^s} \frac{1}{2} v^{j\top} K_P^j v^j + c^{j\top} v^j \quad \text{such that} \quad \mathbf{1}^\top v^j = s, \\ [K_P^j]_{i,i'} := k_P(X_i^j, X_{i'}^j), \quad c_i^j := \sum_{i'=1}^{i-1} \sum_{j'=1}^{s} k_P(X_{\pi(i',j')}, X_j).\end{gathered} \tag{9}$$

Depending on the values of B, M, and N, and the way in which the mini-batches are selected, it may be advantageous to store and reuse kernel calculations from iteration to iteration. In general, however, we assume that the matrix K_P^j and vector c^j are recalculated for each batch, giving the algorithm an overall complexity of $\mathcal{O}(M^2 s^2 B^s)$. This apparently daunting computational complexity can nevertheless be advantageous if N is very large and $B \ll N$. Teymur et al. (2021) recommends a ratio $s/B \approx 10$, though this is expected to be problem dependent. Finding the exact solution of this type of optimization problem is NP-hard, where NP stands for nondeterministic polynomial time.[5] However, a good feasible solution may still be useful. Indeed, the iterative nature of the overall algorithm allows it to compensate, to a degree, for suboptimal selection of states at a given iteration through its selection of states in future iterations. Fortunately, good solutions can readily be obtained using any of a number of packaged discrete optimization routines, such as the commercial software Gurobi, MOSEK, and MATLAB's Optimization Toolbox, or numerous open-source equivalents.

2.4. Summary

This completes our review of postprocessing strategies for MCMC, when the aim is to accurately approximate the distributional target P itself. Given that convergence diagnostics and thinning are well-known techniques, we deliberately focused on their shortcomings in this review. Then, we described recent methodology that aims to directly address the bias-variance trade-off that occurs when postprocessing MCMC output. This trade-off is fundamental to many important and challenging applications of MCMC, in which there is a practical limit to the computational budget. To limit scope, we do not discuss alternative classes of algorithm, such as unbiased Monte Carlo (Jacob et al. 2020), for which a bias-variance trade-off is systematically avoided. Finally, we argued that recent developments in Stein discrepancy have the potential to substantially impact both applications of, and research into, MCMC.

3. APPROXIMATION OF POSTERIOR EXPECTATIONS

In contrast to exploratory Bayesian analyses, several applications of Bayesian statistics require just a finite number of scalar posterior quantities of interest. For example, in a decision-making context, Bayes' rule may take an explicit and simple form, such as the mean of the posterior, or perhaps a median, or a higher moment (Berger 2013). To proceed, one can first run MCMC, followed

[5] Without the cardinality constraint $\mathbf{1}^\top v^j = s$, this integer quadratic program is equivalent to the celebrated Max-Cut problem, and with this constraint to the related cardinality constrained k-partition problem (Rendl 2016).

by suitable postprocessing as described in Section 2, to obtain an approximation to the posterior from which quantities of interest can be extracted. However, approximating the full posterior may incur unnecessary computational effort. In such circumstances it is natural to seek to focus computational resources on approximating just the quantities of interest.

Control variates are a classical technique for reducing the variance of Monte Carlo estimators, which are used in a wide range of applications, including stochastic gradient-based optimization (Wang et al. 2013, Grathwohl et al. 2018) and as part of MCMC methods themselves (Baker et al. 2019). In this section we review the use of control variates as a postprocessing technique for MCMC. We show that modern control variates, unlike their classical counterparts, can facilitate bias removal as well as variance reduction. In Section 3.1 the control variate technique is presented at a general level; then, in Section 3.2, we present specific control variates techniques and explain how these can be used to postprocess MCMC.

3.1. Monte Carlo Estimators

For the purposes of this article, a Monte Carlo estimator is a map $\mu : \Omega \times \mathcal{L}^2(P) \to \mathbb{R}$ whose output $\mu(\omega, f)$ depends on ω only via dependence on a collection of random variables $X_1(\omega), \dots, X_n(\omega)$. The output, $\mu(\omega, f)$, is interpreted as an approximation to the integral $\int f \mathrm{d}P$, which we consider to be a scalar quantity of interest. Our focus is on Monte Carlo estimators that are based on MCMC output, with the standard example being the estimator

$$\mu(\omega, f) = \frac{1}{N} \sum_{i=1}^{N} f(X_i(\omega)), \qquad 10.$$

which takes an average of f over the states $(X_n)_{n \leq N}$ in the MCMC output. Such an estimator is said to be consistent if, for all $f \in \mathcal{L}^2(P)$, the random variable in Equation 10 converges in probability to $\int f \mathrm{d}P$ as $N \to \infty$. The asymptotics of Equation 10 are well studied in the setting where the Markov chain is P-invariant (Meyn & Tweedie 2012). Improved approximations can be obtained using the methods described in Section 2. For example, postprocessed MCMC output of the form in Equation 1 can be used to provide a Monte Carlo estimator

$$\mu(\omega, f) = \sum_{i=1}^{M} w_i f(X_{\pi(i)}(\omega)). \qquad 11.$$

In the setting where the Markov chain is not P-invariant, Equation 10 will be asymptotically biased in general, but Equation 11 may yet be consistent, as explained in Section 2.3, and therefore Equation 11 may be preferred. In the presence of several consistent estimators, it is natural to ask which estimator should be preferred; this question can be rigorously formulated in terms of the mean square error of the estimators and the answer will be f-dependent in general. For convenience, we leave the ω argument implicit in the remainder of this section.

3.1.1. Selecting a Monte Carlo estimator.

The mean square error of a Monte Carlo estimator μ is defined as

$$\mathrm{MSE}(\mu(f)) := \mathbb{E}\left[\left(\mu(f) - \int f \mathrm{d}P\right)^2\right]. \qquad 12.$$

Presented with a collection $\{\mu^\theta\}_{\theta \in \Theta}$ of Monte Carlo estimators, say indexed by $\theta \in \Theta$, we would like to select an estimator for which $\mathrm{MSE}(\mu^\theta(f))$ is minimized. Let us assume that the mean square error can itself be consistently estimated based on the MCMC output, i.e., we have available an

GENERAL RECIPE TO SELECT A MONTE CARLO ESTIMATOR

1. Identify a collection of Monte Carlo estimators $\mu^\theta, \theta \in \Theta$.
2. For each estimator, compute $\widehat{\text{MSE}}(\mu^\theta(f))$.
3. Select $\hat{\theta}$ such that $\theta \mapsto \widehat{\text{MSE}}(\mu^\theta(f))$ is minimized.

estimator $\widehat{\text{MSE}}(\mu(f))$. Then, a general recipe to select a Monte Carlo estimator appears in the sidebar titled General Recipe to Select a Monte Carlo Estimator.

There are at least three possible shortcomings with this general recipe, which will be discussed. First, it is not clear how one should identify an appropriate set of Monte Carlo estimators; control variates provide an elegant solution that we discuss next in Section 3.1.2. Second, it may be challenging to identify a suitable estimator for the mean squared $\widehat{\text{MSE}}$, since the underlying MCMC method may be complicated. Options for this are discussed in Section 3.1.3. Third, estimation error in $\widehat{\text{MSE}}$ presents a challenge when there are many Monte Carlo estimators being compared, since with more estimators there is a greater chance of selecting a poor estimator due to bad luck. A solution to this problem requires that the size of the set of candidate estimators is controlled in some way commensurate with the error in $\widehat{\text{MSE}}$. Several solutions are discussed in Section 3.2, including restricting the size of this set through the use of explicit finite-dimensional bases, and through coupling the size of Θ to the size N of the MCMC output.

We emphasize that, compared with the techniques reviewed in Section 2, the selection of Monte Carlo estimators remains as much an art as a science. Theoretical analyses are available on some aspects of the general recipe just outlined and will be highlighted, but to our knowledge there does not yet exist a theoretical treatment that is broadly applicable in the MCMC context.

3.1.2. Constructing Monte Carlo estimators using control variates. An element $g \in \mathcal{L}^2(P)$ is said to be a control variate (for P) if $\int g \mathrm{d}P = 0$. Clearly any finite linear combination of control variates is also a control variate, and we use $\mathcal{G}$ to denote a linear subspace of $\mathcal{L}^2(P)$ whose elements are control variates. The power of control variates is that they enable one to take a single Monte Carlo estimator, such as Equation 10, and from this generate a possibly large collection of Monte Carlo estimators. Indeed, armed with a consistent Monte Carlo estimator μ and a set of control variates $\mathcal{G}$, one can consider Monte Carlo estimators of the form $\mu^\theta(f) := \theta_1 + \mu(f - \theta_1 - \theta_2)$ where $\theta_1 \in \mathbb{R}, \theta_2 \in \mathcal{G}, \Theta = \mathbb{R} \times \mathcal{G}$. The consistency of μ is automatically inherited by each μ^θ.

Up to this point we have not discussed how control variates can be found in practice. Many approaches for developing control variates in the context of Markov chain sampling are based on approximating the solution $\hat{f}$ to the typically intractable Poisson equation

$$\hat{f} - K\hat{f} = f - \mathbb{E}[f], \qquad 13.$$

where K is the one-step-ahead prediction operator $K\hat{f} = E[\hat{f}(X^{(n+1)})|X^{(n)} = x]$. In this setting, one could evaluate $\mathbb{E}[f]$ exactly by evaluating $f + K\hat{f} - \hat{f}$. Andradóttir et al. (1993) propose numerical algorithms to approximate this solution in the context of finite state spaces. Henderson (1997) approximates the solution for specific Markov samplers, focusing on continuous-time processes and applications in stochastic network theory. This was extended in Dellaportas & Kontoyiannis (2012) for reversible Markov chains where K is tractable for some basis functions. A method to approximate the solution to the Poisson equation by discretizing the state space for geometrically ergodic Metropolis–Hastings chains is introduced by Mijatović & Vogrinc (2018).

Control variate: a square-integrable function whose expectation is 0

Control variates have also been built for independent Metropolis–Hastings samplers (Atchadé & Perron 2005) and for general Metropolis–Hastings samplers (Hammer & Tjelmeland 2008), although the latter approach requires an extension of the state space to include proposals.

The aforementioned control variates are sampler specific or require adjustments to the MCMC algorithm. Section 3.2 describes sampler-agnostic control variates that are applicable when $\nabla \log p$ or an unbiased estimate is available.

3.1.3. Proxies for mean square error.

The problem of estimating the mean square error of a Monte Carlo estimator is difficult, due to the fact that both the dependence between the states $(X_n)_{n \leq N}$ in MCMC output and the way that these states are combined in the Monte Carlo estimator μ can be arbitrarily complicated. Readers are directed to, for example, Flegal & Jones (2010) for strategies that can be used to estimate the mean square error of the Monte Carlo estimator in Equation 10. To promote generality, here we consider simple and generic proxies for mean square error that are easily computed, and much of what we recommend is based on empirical evidence only.

A simple proxy for mean square error can be obtained by considering Equation 11 in the idealized setting where $X_i \sim P$, for which it follows

$$\mathrm{MSE}\left(\sum_{i=1}^{M} w_i f(X_{\pi(i)})\right) \leq \frac{1}{M} \int \left(f - \int f \mathrm{d}P\right)^2 \mathrm{d}P =: \frac{\mathrm{Var}(f)}{M}, \qquad 14.$$

with equality when the X_i are independent. The variance $\mathrm{Var}(f)$ can be estimated using the empirical variance

$$\widehat{\mathrm{Var}}(f) := \sum_{i=1}^{M} w_i \left(f(X_{\pi(i)}) - \sum_{j=1}^{M} w_j f(X_{\pi(j)}) \right)^2, \qquad 15.$$

evaluated using MCMC output. Empirical variance minimization for constructing control variates was studied by Belomestny et al. (2021) for the case where the X_i are independent. For nonindependent X_i, arising as MCMC output, a more involved proxy based on spectral approximation of the asymptotic variance was studied by Brosse et al. (2019) and Belomestny et al. (2020a,b), representing probably the most successful attempt to date to provide theory for control variates for postprocessing MCMC output. A popular and simple upper bound on Equation 15 is the least squares estimator

$$\widehat{\mathrm{LS}}(f) := \sum_{i=1}^{M} w_i f(X_{\pi(i)})^2. \qquad 16.$$

An empirical comparison of empirical variance and least squares estimators for the selection of control variates by Si et al. (2020) reported that, perhaps surprisingly, the least squares estimator performed best. That is, one selects $\hat{\theta} \in \mathbb{R} \times \mathcal{G}$ in order that $\theta \mapsto \widehat{\mathrm{LS}}(f - \theta_1 - \theta_2)$ is minimized. The scalar integral of interest is then estimated as

$$\int f \mathrm{d}P \approx \mu^{\hat{\theta}}(f) = \hat{\theta}_1 + \underbrace{\sum_{i=1}^{M} w_i \left(f(X_{\pi(i)}) - \hat{\theta}_1 - \hat{\theta}_2(X_{\pi(i)}) \right)}_{=0}, \qquad 17.$$

where we have used the defining optimality property of $\hat{\theta}_2$ to conclude that the summation in Equation 17 is zero. One can equivalently describe this estimator as the result of first solving the weighted least squares regression problem

$$f(x_i) = \theta_1 + \theta_2(x_i) + \epsilon_i \qquad 18.$$

for the intercept $\theta_1 \in \mathbb{R}$ and the predictor $\theta_2 \in \mathcal{G}$, where the dataset consists of the (random) covariates $x_i = X_{\pi(i)}$ and independent errors $\epsilon_i \sim \mathcal{N}(0, w_i^{-1})$, $i = 1, \dots, M$, then reporting the fitted intercept $\hat{\theta}_1$ as an approximation to the integral of interest. Next we address the question of how a set $\mathcal{G}$ of control variates can actually be constructed.

3.2. Gradient-Based Control Variates

Perhaps the main challenge in the application of control variates is identifying a suitable linear subspace $\mathcal{G}$. The elements of $\mathcal{G}$ should (*a*) have known expectation under P, (*b*) be easy to compute, and (*c*) offer an improvement on a Monte Carlo estimator μ that would otherwise have been used, in the sense that $\mathrm{MSE}(\mu^{\theta}(f)) < \mathrm{MSE}(\mu(f))$ for some $\theta \in \mathbb{R} \times \mathcal{G}$. In this section we discuss gradient-based control variates that often meet these requirements, focusing on domains $\mathcal{X} = \mathbb{R}^d$ for some $d \in \mathbb{N}$. These gradient-based control variates are constructed using mathematical tools similar to those exploited in Section 2. The construction of control variates for non-Euclidean domains is discussed by Barp et al. (2021) for closed manifolds, while the general case, including discrete domains, remains underdeveloped.

Recall the operator $\mathcal{A}_P$ defined in Equation 3; i.e., $\mathcal{A}_P h = \nabla \cdot h + \nabla \log p \cdot h$ where $h : \mathbb{R}^d \to \mathbb{R}^d$. It was shown (in the sidebar titled Tail Condition for Stein Discrepancy) that $\int \mathcal{A}_P h \, \mathrm{d}P = 0$ under an appropriate tail condition on h; it therefore is natural to consider a linear subspace of $\mathcal{L}^2(P)$ consisting of control variates of the form $\mathcal{G} = \mathcal{A}_P \Phi = \{\mathcal{A}_P \phi : \phi \in \Phi\}$, where $\Phi := \{\phi : \mathbb{R}^d \to \mathbb{R}^d\}$ is a linear space of functions for which the aforementioned tail condition is satisfied. The form for gradient-based control variates described here can be traced to the physics literature (Assaraf & Caffarel 1999, 2003) and was first brought to bear on MCMC in Mira et al. (2013). As discussed in Section 2, the required gradients are produced as a byproduct when gradient-based samplers, such as the Metropolis-adjusted Langevin algorithm (Roberts & Stramer 2002) or Hamiltonian Monte Carlo (Duane et al. 1987), are used, making the combination of gradient-based sampling and gradient-based control variates particularly appealing (Papamarkou et al. 2014).

It remains to be discussed how the set Φ of differentiable vector fields $\phi : \mathbb{R}^d \to \mathbb{R}^d$ can be selected. In what follows, we review some of the main choices that previous researchers have considered.

3.2.1. Finite-dimensional basis.

Perhaps the simplest choice for Φ is the linear span of a finite set $\{\phi_1, \dots, \phi_J\}$. There is clearly much flexibility in the choice of the vector fields ϕ_j, but a popular choice is to use the gradients of monomials. Specifically, the so-called zero-variance control variates (ZVCV) of Assaraf & Caffarel (1999, 2003) and Mira et al. (2013) set Φ to be gradients of the class of rth order polynomials, $\Phi = \mathrm{span}\{\nabla \mathbf{x}^{\alpha} : \alpha \in \mathbb{N}_0^d, 0 < |\alpha| \leq r\}$ where $r \in \mathbb{N}$, $\mathbf{x}^{\alpha} = \prod_{i=1}^{d} x_i^{\alpha_i}$ and $|\alpha| = \sum_{i=1}^{d} |\alpha_i|$. The number of basis functions is therefore $J = \binom{d+r}{d}$, and the associated set $\mathcal{G}$ of control variates contains elements of the form

$$\mathcal{A}_P(\nabla \mathbf{x}^{\alpha}) = \sum_{j=1}^{d} \alpha_j \left[(\alpha_j - 1) x_j^{\alpha_j - 2} + x_j^{\alpha_j - 1} \nabla_{x_j} \log p(x) \right] \prod_{i \neq j} x_i^{\alpha_i}. \qquad 19.$$

Having identified Φ, we can aim to select an optimal control variate from $\mathcal{G}$ using one of the proxies for mean square error discussed in Section 3.1.3. Suppose that $J < M$ and consider the least squares proxy in Equation 16. In what follows we consider a Monte Carlo estimator of the form in Equation 11, which of course contains, as a special case, the vanilla Monte Carlo estimator in Equation 10. Then, we solve the regression problem in Equation 18 to obtain a fitted regression model

$$\hat{f}(x) = \hat{\theta}_1 + \sum_{j=1}^{J} \hat{\theta}_{2,j} \mathcal{A}_P \phi_j(x), \qquad 20.$$

where we collect the regression coefficients together into a vector $\mathbf{c} = (\hat{\theta}_1, \hat{\theta}_{2,1}, \dots, \hat{\theta}_{2,J})^{\mathrm{T}} \in \mathbb{R}^{J+1}$. For completeness we now provide an explicit formula for the coefficient vector $\mathbf{c}$ in the fitted model. Let

Semiexact: a Monte Carlo estimator is semiexact if it is exact on a linear subspace of $\mathcal{L}^2(P)$

$$\mathbf{f} = \begin{bmatrix} f(X_{\pi(1)}) \\ \vdots \\ f(X_{\pi(M)}) \end{bmatrix}, \quad W = \begin{bmatrix} w_1 & & \\ & \ddots & \\ & & w_M \end{bmatrix}, \quad \Phi = \begin{bmatrix} 1 & \mathcal{A}_P\phi_1(X_{\pi(1)}) & \cdots & \mathcal{A}_P\phi_J(X_{\pi(1)}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \mathcal{A}_P\phi_1(X_{\pi(M)}) & \cdots & \mathcal{A}_P\phi_J(X_{\pi(M)}) \end{bmatrix}.$$

Then standard calculations show that selecting $\mathbf{c}$ to minimize $\widehat{\mathrm{LS}}(f - \hat{f})$ leads to the estimated coefficients being $\hat{\mathbf{c}} = (\Phi^\top W \Phi)^{-1} \Phi^\top W \mathbf{f}$. The integral $\int f \mathrm{d}P$ of interest is approximated by $\hat{\theta}_1$, the first component of $\mathbf{c}$. The Monte Carlo estimator so obtained will be denoted $\mu^{\mathrm{ZVCV}}(f) = \hat{\mathbf{c}}^{\mathrm{T}} \mathbf{e}_1 = \hat{\theta}_1$ in the following.

An intriguing property of gradient-based control variates with finite-dimensional bases is that, under many of the proxies for mean square error that we discussed, the resulting Monte Carlo estimators are semiexact, in the sense that $\mathrm{MSE}(\mu^{\mathrm{ZVCV}}(f)) = 0$ when $f \in \mathrm{span}\{1\} \oplus \mathcal{A}_P\Phi$.

Recalling that for a Gaussian P, the gradient $\nabla \log p$ is a first order polynomial, semiexactness in this case carries the interpretation of being exact for polynomials up to a certain order when Φ consists of gradients of monomials, in a similar way to how Gaussian cubature methods are constructed. This explains the "zero variance" nomenclature used by Assaraf & Caffarel (1999, 2003) and Mira et al. (2013).

The main problem with using a finite-dimensional basis is that the regression problem is typically misspecified, since $f \notin \mathrm{span}\{1\} \oplus \mathcal{A}_P\Phi$ for most functions f of interest. This limits the variance reduction that can be achieved. To improve convergence rates, one could consider increasing the size of Φ with increasing M, in the spirit of Portier & Segers (2018) and South et al. (2018), or using an infinite-dimensional basis with regularization, as described next.

3.2.2. Infinite-dimensional basis.

Oates et al. (2017) extended the gradient-based control variates of Assaraf & Caffarel (1999, 2003) and Mira et al. (2013) to an infinite-dimensional linear subspace of $\mathcal{L}^2(P)$. This was achieved by taking $\Phi = \mathcal{H}(k)^d$ to be a Cartesian product of reproducing kernel Hilbert spaces $\mathcal{H}(k)$ of sufficiently regular functions (see the sidebar titled Kernel Stein Discrepancy in Section 2 for background). The resulting set of control variates is $\mathcal{G} = \mathcal{A}_P\Phi = \mathcal{H}(k_P)$, which is again a reproducing kernel Hilbert space with reproducing kernel $k_P(x, y)$ defined in the first equation in the sidebar titled Kernel Stein Discrepancy. The resulting method was referred to as control functionals (CF), a nonparametric (or functional) generalization of existing control variates.

The major challenge associated with an infinite-dimensional set $\mathcal{G}$ of control variates is overfitting; there may be infinitely many $\theta \in \mathbb{R} \times \mathcal{G}$ for which $\widehat{\mathrm{MSE}}(\mu^\theta(f)) = 0$, yet in reality $\mathrm{MSE}(\mu^\theta(f))$ may be arbitrarily large. Consider, for instance, the least squares proxy

$$\widehat{\mathrm{LS}}(f - \theta_1 - \theta_2) = \sum_{i=1}^{M} w_i \left(f(X_{\pi(i)}) - \theta_1 - \theta_2(X_{\pi(i)}) \right)^2, \qquad 21.$$

which can be driven to zero by taking θ_2 to interpolate $f - \theta_1$ at the nodes $X_{\pi(i)}$, $i = 1, \dots, M$. Constraining θ_2 at a finite set of locations does not constrain what θ_2 may do outside this finite set and is therefore not sufficient to provide control on $\mathrm{MSE}(\mu^{\theta}(f))$. The methodological contribution of Oates et al. (2017) was to select, among the set of $\theta \in \mathbb{R} \times \mathcal{G}$ for which Equation 21 is minimized, an element with minimal seminorm, where the seminorm on $\mathbb{R} \times \mathcal{G}$ was defined as $|\theta|^2 = \langle \theta_2, \theta_2 \rangle_{\mathcal{H}(k_P)}$. Under regularity assumptions, it can be shown that there exists a unique such element $\theta \in \mathbb{R} \times \mathcal{G}$. Moreover, there is a closed-form solution to this optimization problem that leads to the estimator $\mu^{\mathrm{CF}}(f) = (\mathbf{1}^{\top} K_P^{-1} \mathbf{1})^{-1} (\mathbf{1}^{\top} K_P^{-1} \mathbf{f})$, where $[K_P]_{i,j} = k_P(X_{\pi(i)}, X_{\pi(j)})$. Note that we may, without loss of generality, assume that the $X_{\pi(i)}$ are distinct in Equation 11, since otherwise we could consider smaller M and modify the weights accordingly. This ensures that the matrix K_P is nonsingular whenever k_P is a genuine reproducing kernel. An interesting feature, and possible weakness, of CF is that the Monte Carlo estimator obtained does not depend on the weights w_i appearing in Equation 11.

Super-$\sqrt{M}$ convergence: the property of having a convergence rate that is $o(M^{-1/2})$

The performance of CF is heavily dependent on the choice of the kernel k. A common choice is to use a radial kernel k, such that $k(x, y)$ depends only on $\|x - y\|$, with examples including the Gaussian, Matérn, and rational quadratic kernels (Rasmussen 2003). Typically such kernels will be parametric, with a small number of parameters ℓ that must be specified. Oates et al. (2017) recommended using cross-validation to select kernel parameters ℓ, wherein a subset of the $\{X_{\pi(i)}, i \in I_{\mathrm{train}}\}$ are used construct the Monte Carlo estimator $\mu^{\theta}(f)$ where $\theta = \hat{\theta}^{\ell} \in \mathbb{R} \times \mathcal{G}$ and performance of this Monte Carlo estimator associated with ℓ is measured by the sum of squared errors $E_{\ell} := \sum_{i \in I_{\mathrm{test}}} w_i (f(X_{\pi(i)}) - \hat{\theta}_1^{\ell} - \hat{\theta}_2^{\ell}(X_{\pi(i)}))^2$, where $I_{\mathrm{test}} = \{1, \ldots, M\} \backslash I_{\mathrm{train}}$. One then selects the kernel parameters ℓ for which E_{ℓ} is minimized.

Under regularity assumptions, CF have theoretical advantages over ZVCV. Oates et al. (2019) and Barp et al. (2021) used results from scattered data approximation (Wendland 2004) to prove that, in the uniformly weighted case (i.e., $w_i = \frac{1}{M}$), the expected error $\mathbb{E}[|\mu^{\hat{\theta}}(f) - \int f \mathrm{d}P|]$ converges at a rate $O(M^{-s/d} \log(M)^{-s/d})$, where here s is the number of (weak) derivatives of the function f whose integral is sought. This indicates that the use of CF for postprocessing MCMC output can actually improve the convergence rate of the estimator compared with standard MCMC, provided that the smoothness s of f is commensurate with the dimension d of the domain on which it is defined (i.e., $s > \frac{d}{2}$). Thus, CF are an example of a method that offers super-$\sqrt{M}$ convergence. The main weakness of CF is that their performance can be inferior to ZVCV when the dimension d is high relative to the size M of the dataset; next, we discuss how this weakness can be addressed.

3.2.3. Mixed basis.

To address the poor performance of CF relative to CV in the high-dimensional context, South et al. (2021) generalized the approaches discussed in Sections 3.2.1 and 3.2.2, to consider functional approximations of the form

$$\hat{f}(x) = \hat{\theta}_1 + \hat{\theta}_2(x) + \sum_{j=1}^{J} \hat{\theta}'_{2,j} \mathcal{A}_P \phi_j(x), \qquad 22.$$

where the parameters $\hat{\theta}$, consisting of $\hat{\theta}_1 \in \mathbb{R}$, $\hat{\theta}_2 \in \mathcal{H}(k_P)$ and $\hat{\theta}'_2 \in \mathbb{R}^J$, are about to be specified. Notice that one recovers the same form of approximation used in ZVCV, i.e., Equation 20, as the special case where $\hat{\theta}_2 = 0$. Similarly, one can show that the same form as CF is recovered when $\hat{\theta}'_2 = 0$. Thus, Equation 22 represents a strict generalization of ZVCV and CF, and one may hope to obtain the best of both worlds in terms of the superior performance of ZVCV in high dimensions and the super-$\sqrt{M}$ convergence of CF. The performance of this hybrid approach depends on how the parameters $\hat{\theta}$ are selected. Following Sard (1949), South et al. (2021) propose to select $\hat{\theta}$ such that the following properties are satisfied:

1. $\hat{f} = f$ for all $f \in \mathrm{span}\{1\} \oplus \mathcal{A}_P \Phi$, where $\Phi = \mathrm{span}\{\phi_1, \ldots, \phi_J\}$

Table 1 Properties of the control variate methods we have discussed

Method	Complexity	Semiexact	Bias-correcting	Super-$\sqrt{M}$
Vanilla MCMC	$O(Nd)$	No	No	No
ZVCV + $\widehat{\mathrm{LS}}$	$O(Nd + Md^{2r} + d^{3r})$	Yes	No	No
CF	$O(Nd + M^3 + M^2 d)$	No	Yes	Yes (e.g., Barp et al. 2021)
SECF	$O(Nd + M^3 + d^{3r})$	Yes	Yes	Yes (conjectured)

Abbreviations: CF, control functionals; LS, least squares; MCMC, Markov chain Monte Carlo; SECF, semiexact control functional; ZVCV, zero-variance control variates.

2. $\widehat{\mathrm{LS}}(f - \hat{f}) = 0$
3. $\hat{\theta}_2$ minimizes $\theta_2 \mapsto \langle \theta_2, \theta_2 \rangle_{\mathcal{H}(k_P)}$ subject to the first two properties being satisfied

The first property is to ensure semiexactness, and the second is an interpolation requirement. The third property amounts to minimizing the seminorm $|\theta| = \langle \theta_2, \theta_2 \rangle_{\mathcal{H}(k_P)}$ and serves to ensure uniqueness of $\hat{\theta}$ and to penalize complexity, similarly to CF. This method is referred to as a semiexact control functional (SECF), and the closed-form solution for the estimator is $\mu^{\mathrm{SECF}}(f) = \mathbf{e}_1^{\top}(\Phi^{\top} K_P^{-1} \Phi)^{-1} \Phi^{\top} K_P^{-1} \mathbf{f}$. If there are parameters ℓ in the kernel k that must be specified, then cross-validation can be applied in a similar way to that described in Section 3.2.2. Similarly to CF, a possible weakness of this hybrid approach is that the Monte Carlo estimator obtained does not depend on the weights w_i appearing in Equation 11.

South et al. (2021) demonstrated that such a hybrid approach can indeed enjoy the advantages of both ZVCV and CF; we illustrate this below in Section 3.2.4.1. Open-source software is available for ZVCV, CF, and SECF, via the `ZVCV` package (South 2020) on the comprehensive R archive network (CRAN). The required input for this package is a set of M samples and the associated evaluations of $f(\cdot)$ and $\nabla \log p(\cdot)$.

3.2.4. Practical considerations.

Earlier we alluded to the construction of control variates being more an art than a science; here, we provide practical recommendations based on our personal experience using control variates to postprocess MCMC.

3.2.4.1. Choosing a control variate method.

Choosing between various control variate methods, like ZVCV, CF, and SECF, is nontrivial. Cross-validation approaches are computationally expensive and prone to incorrect decisions due to the need to reduce the sample size in each fold. It would therefore be helpful to have an understanding of the theoretical properties of different methods. Unfortunately, such theoretical analyses are underdeveloped at present. Specifically, the theory that does exist tends to involve assumptions that are difficult to verify in practice, if they hold at all. **Table 1** summarizes the current state of knowledge for the methods that we have discussed.

The positive entries in **Table 1** should be interpreted as there being (possibly strong) theoretical assumptions under which the result has been established. The fact that CF and SECF are bias-correcting will not come as a surprise to the reader in light of the discussion in Section 2.3.1. A perhaps more useful approach to selection of a control variate method is to explore their empirical performance in the context of a synthetic test-bed.

Bias-correcting: capable of removing asymptotic bias in certain biased MCMC algorithms

Example 2. Here we compare the performance of different control variate methods on a simple toy example that aims to represent the (relatively common) situation in which P is approximately Gaussian, which may hold in applications for which there is a Bernstein-von-Mises limit. For illustrative purposes, we use a 1-dimensional unit Gaussian distribution with density

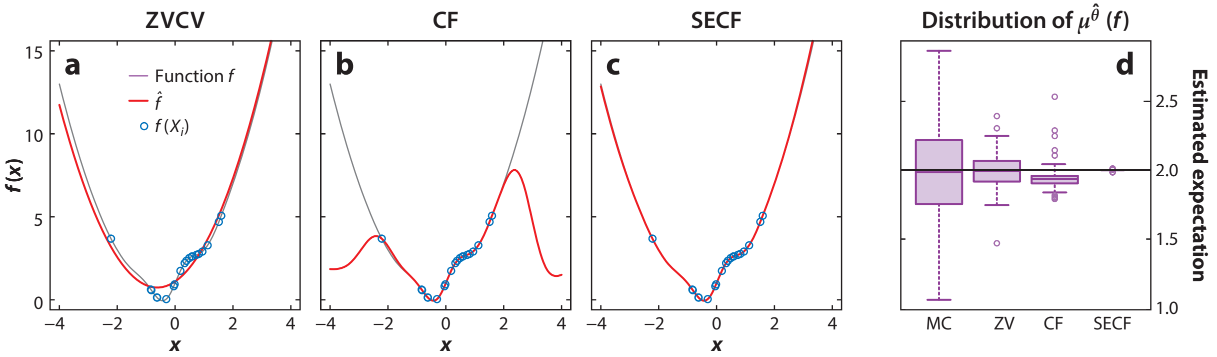

Figure 5

Gradient-based control variates in a toy example. Panels *a–c* show the function f of interest (*gray line*) along with the values $f(X_i)$ computed at the random locations $(X_i)_{i \le N}$, $N = 20$ (*blue circles*). These data are used to construct approximations $\hat{f}$ (*red line*) to f, in each of the methods zero-variance control variates, CF, and SECF. Panel *d* shows boxplots of 100 independent estimates for the integral $\int f \mathrm{d}P$ of interest, with the true value of the integral shown as a thick black line. Abbreviations: CF, control functionals; MC, Monte Carlo; SECF, semiexact control functional; ZVCV, zero-variance control variates.

$p(x) = (2\pi)^{-1/2}\exp(-x^2/2)$ and we estimate the posterior expectation of $f(x) = 1 + x + x^2 + \sin(\pi x)\exp(-x^2)$, for which one can verify $\int f \mathrm{d}P = 2$. This function f was chosen because the combination of complex behavior near $x = 0$ and polynomial behavior in the tails presents challenges for both the parametric and nonparametric methods. For simplicity we consider an idealized MCMC algorithm that samples X_i independently from P, and we consider the vanilla Monte Carlo (MC) estimator $\frac{1}{N}\sum_{i=1}^{N} f(X_i)$ as our starting point; i.e., we seek to reduce the variance of this Monte Carlo estimator using a control variate method.

The results are shown in **Figure 5**. Here, the approximating function $\hat{f}$ for ZVCV is a second-order polynomial,[6] which provides a poor approximation in the region where there are data but provides a good approximation in the tail (Section 3.2.4.1). For CF,[7] the interpolant $\hat{f}$ performs well in regions where there are data, less so in the tail (Section 3.2.4.1). In contrast, SECF[8] is seen to enjoy the best of both worlds, behaving like CF in the region of the data and like ZVCV in the tail (Section 3.2.4.1). Examining the sampling distribution of these estimators through repeated simulation, we observe a remarkable increase in accuracy as a result of post-processing the MCMC output (Section 3.2.4.1). Although the total computing time for the 100 repeated simulations increases from approximately 0.02 seconds for vanilla Monte Carlo integration to 0.11 seconds for ZVCV, 0.17 seconds for CF, and 0.19 seconds for SECF, all three control variate methods improve upon the vanilla Monte Carlo estimate in terms of the overall efficiency measured by the product of mean square error and computing time.

3.2.4.2. Computational cost. For many problems the benefit provided by control variates is not justified when the computational cost of implementation is taken into account (see **Table 1**). However, when the cost of obtaining MCMC output, or the cost of evaluating f on MCMC output, is sufficiently high then control variates can be a useful tool. For borderline cases, Si et al. (2020) demonstrated the use of stochastic gradient descent to speed up the optimization

[6] ZVCV was implemented with a polynomial basis of order $r = 2$, fit using $\widehat{\mathrm{LS}}$. The form of $\hat{f}$ as a second order polynomial can be derived using Equation 19 and $\nabla \log p = -x$.

[7] CF were implemented with a Gaussian kernel $k(x, y) = \exp(-x^2/\lambda^2)$ where λ is selected from $10^{\{-2,-1,0,1,2\}}$ using 3-fold cross-validation.

[8] SECF was implemented with $r = 2$ and a Gaussian kernel $k(x, y) = \exp(-x^2/\lambda^2)$ where λ was selected in the same way as CF.

in step 3 of the general recipe to select a Monte Carlo estimator. A reduced-cost SECF method, based on a low-rank Nyström approximation, was also proposed by South et al. (2021).

3.2.4.3. Curse of dimension. The gradient-based control variates that we discussed suffer from a curse of dimension, which is most evident in kernel methods like CF. However, the regression perspective in Equation 18 suggests that, by analogy with high-dimensional regression modeling (Bühlmann & Van De Geer 2011), it may be possible to construct control variates for functions f whose effective dimension is small, despite a high ambient dimension of $\mathcal{X}$. Additional regularization can be introduced to this effect (South et al. 2018, Wan et al. 2019), with positive results reported for $d \leq 100$. For even larger d, it may be sensible to pursue nonlinear approximation (DeVore 1998), where the basis Φ is restricted to allow dependence only on a subset of the parameters (so-called a priori regularization in South et al. 2018).

3.3. Summary

This section focused on the application of gradient-based control variates to approximate an integral of interest based on output from MCMC. Applications to other sampling algorithms, such as population MCMC (Oates et al. 2016), stochastic gradient Langevin dynamics (Baker et al. 2019), sequential Monte Carlo (South et al. 2018), and unbiased MCMC with couplings (South et al. 2019) have also been considered, and much of our discussion applies unchanged. Applications to estimation of the normalizing constant of the posterior have also been considered in the population MCMC and sequential Monte Carlo sampler settings (Oates et al. 2016, South et al. 2018). Again, the extension is straightforward and consists of applying the ideas from this section to improve multiple expectations. The `ZVCV` package (South 2020) on CRAN provides functions to apply ZVCV and CF to two estimators of the normalizing constant.

A current weakness of control variate methodology is that it is underdeveloped from a theoretical perspective; our focus was on sets of control variates that form linear subspaces of $\mathcal{L}^2(P)$, for which some limited theoretical understanding has been achieved, but more sophisticated sets of control variates have also been empirically considered. For example, Wan et al. (2019) and Si et al. (2020) proposed using the gradients of neural networks for the set Φ. A neural network is parameterized by a collection of weights and biases, which are jointly estimated using stochastic gradient descent applied to a proxy for mean square error, as discussed in Section 3.1.3. These authors found empirically that this approach can lead to improved performance over methods like ZVCV and CF in the high-dimensional context. In light of the anticipated technical complexity required to analyze such sophisticated control variate methods, we expect that empirical assessment will continue to be the primary means through which control variate methodology is developed and assessed.

4. DISCUSSION

MCMC has become a core part of most graduate programs in statistics due to its effectiveness in enabling Bayesian analyses to be performed. Perhaps understandably, these programs focus on the design and validity of algorithms, emphasizing the elegant probabilistic arguments that are often involved. However, this leaves little or no time to discuss postprocessing of MCMC output. In fact, our impression is that many professional users of MCMC are also not aware of this aspect, beyond convergence diagnostics and burn-in removal. Through writing this review, we hope greater attention may be given to this underappreciated but important practical side of MCMC. In particular, the topic is receiving considerable attention from computational researchers at the time of writing, and we extend an invitation to the interested reader to explore further into the recent works cited.

DISCLOSURE STATEMENT

Aside from being authors of some of the literature that was discussed, 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

M.R., O.T., and C.J.O. were supported by the Lloyd's Register Foundation program on data-centric engineering at the Alan Turing Institute, United Kingdom. M.R. was supported by the British Heart Foundation—Alan Turing Institute cardiovascular data science award (BHF; SP/18/6/33805) and by the Wellcome/EPSRC Centre for Medical Engineering (WT203148/Z/16/Z). For the purpose of open access, the authors have applied a CC BY public copyright license to any Author Accepted Manuscript version arising from this submission. The authors thank Matt Graham, Aki Vehtari, Ioannis Kontoyiannis, Pierre Jacob and an anonymous reviewer for helpful comments.

LITERATURE CITED

Andradóttir S, Heyman DP, Ott TJ. 1993. Variance reduction through smoothing and control variates for Markov chain simulations. *ACM Trans. Model. Comput. Simul.* 3(3):167–89

Assaraf R, Caffarel M. 1999. Zero-variance principle for Monte Carlo algorithms. *Phys. Rev. Lett.* 83(23):4682–85

Assaraf R, Caffarel M. 2003. Zero-variance zero-bias principle for observables in quantum Monte Carlo: application to forces. *J. Chem. Phys.* 119(20):10536–52

Atchadé YF, Perron F. 2005. Improving on the independent Metropolis-Hastings algorithm. *Stat. Sin.* 15(1):3–18

Baker J, Fearnhead P, Fox EB, Nemeth C. 2019. Control variates for stochastic gradient MCMC. *Stat. Comput.* 29(3):599–615

Barp A, Oates CJ, Porcu E, Girolami M. 2021. A Riemann–Stein kernel method. *Bernoulli*. In press

Belomestny D, Iosipoi L, Moulines É, Naumov A, Samsonov S. 2020a. Variance reduction for dependent sequences with applications to stochastic gradient MCMC. arXiv:2008.06858 [math.ST]

Belomestny D, Iosipoi L, Moulines É, Naumov A, Samsonov S. 2020b. Variance reduction for Markov chains with application to MCMC. *Stat. Comput.* 30(4):973–97

Belomestny D, Iosipoi L, Paris Q, Zhivotovskiy N. 2021. Empirical variance minimization with applications in variance reduction and optimal control. *Bernoulli*. In press

Berger JO. 2013. *Statistical Decision Theory and Bayesian Analysis*. New York: Springer

Berlinet A, Thomas-Agnan C. 2011. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. New York: Springer

Bissiri PG, Holmes CC, Walker SG. 2016. A general framework for updating belief distributions. *J. R. Stat. Soc. Ser. B* 78(5):1103–30

Biswas N, Jacob PE, Vanetti P. 2019. Estimating convergence of Markov chains with *L*-lag couplings. In *Advances in Neural Information Processing Systems 32*, ed. H Wallach, H Larochelle, A Beygelzimer, F d'Alché-Buc, E Fox, R Garnett. Red Hook, NY: Curran

Briol FX, Oates CJ, Cockayne J, Chen WY, Girolami M. 2017. On the sampling problem for kernel quadrature. In *Proceedings of the 34th International Conference on Machine Learning*, ed. D Precup, YW Teh, pp. 586–95. N.p.: PMLR

Briol FX, Oates C, Girolami M, Osborne M, Sejdinovic D. 2019. Probabilistic integration: a role in statistical computation? *Stat. Sci.* 34(1):1–22

Brooks S, Gelman A, Jones G, Meng XL, eds. 2011. *Handbook of Markov Chain Monte Carlo*. Boca Raton, FL: Chapman and Hall/CRC

Brooks SP, Gelman A. 1998. General methods for monitoring convergence of iterative simulations. *J. Comput. Graph. Stat.* 7(4):434–55

Brosse N, Durmus A, Meyn S, Moulines E, Radhakrishnan A. 2019. Diffusion approximations and control variates for MCMC. arXiv:1808.01665 [stat.ME]

Bühlmann P, Van De Geer S. 2011. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. New York: Springer

Carpenter B, Gelman A, Hoffman MD, Lee D, Goodrich B, et al. 2017. Stan: A probabilistic programming language. *J. Stat. Softw.* 76(1)

Chen WY, Barp A, Briol FX, Gorham J, Girolami M, et al. 2019. Stein point Markov chain Monte Carlo. In *Proceedings of the 36th International Conference on Machine Learning*, ed. K Chaudhuri, R Salakhutdinov, pp. 1011–21. N.p.: PMLR

Chen WY, Mackey L, Gorham J, Briol FX, Oates CJ. 2018. Stein points. In *Proceedings of the 35th International Conference on Machine Learning*, ed. J Dy, A Krause, pp. 844–53. N.p.: PMLR

Chopin N. 2002. A sequential particle filter method for static models. *Biometrika* 89(3):539–52

Chwialkowski K, Strathmann H, Gretton A. 2016. A kernel test of goodness of fit. In *Proceedings of The 33rd International Conference on Machine Learning*, ed. MF Balcan, KQ Weinberger, pp. 2606–15. N.p.: PMLR

Dellaportas P, Kontoyiannis I. 2012. Control variates for estimation based on reversible Markov chain Monte Carlo samplers. *J. R. Stat. Soc. Ser. B* 74(1):133–61

DeVore RA. 1998. Nonlinear approximation. *Acta Numer.* 7:51–150

Dick J, Pillichshammer F. 2010. *Digital Nets and Sequences: Discrepancy Theory and Quasi–Monte Carlo Integration*. Cambridge, UK: Cambridge Univ. Press

Duane S, Kennedy AD, Pendleton BJ, Roweth D. 1987. Hybrid Monte Carlo. *Phys. Lett. B* 195(2):216–22

Flegal JM, Jones GL. 2010. Batch means and spectral variance estimators in Markov chain Monte Carlo. *Ann. Stat.* 38(2):1034–70

Gelman A, Carlin JB, Stern HS, Dunson DB, Vehtari A, Rubin DB. 2013. *Bayesian Data Analysis*. Boca Raton, FL: Chapman and Hall/CRC. 3rd ed.

Gelman A, Carlin JB, Stern HS, Rubin DB. 2003. *Bayesian Data Analysis*. Boca Raton, FL: Chapman and Hall/CRC. 2nd ed.

Gelman A, Rubin DB. 1992. Inference from iterative simulation using multiple sequences. *Stat. Sci.* 7(4):457–72

Geyer CJ. 1992. Practical Markov chain Monte Carlo. *Stat. Sci.* 7(4):473–83

Gorham J, Duncan AB, Vollmer SJ, Mackey L. 2019. Measuring sample quality with diffusions. *Ann. Appl. Probab.* 29(5):2884–928

Gorham J, Mackey L. 2015. Measuring sample quality with Stein's method. In *Advances in Neural Information Processing Systems 28*, ed. C Cortes, N Lawrence, D Lee, M Sugiyama, R Garnett, pp. 226–34. Cambridge, MA: MIT Press

Gorham J, Mackey L. 2017. Measuring sample quality with kernels. In *Proceedings of the 34th International Conference on Machine Learning*, ed. D Precup, YW Teh, pp. 1292–301. N.p.: PMLR

Gramacy R, Samworth R, King R. 2010. Importance tempering. *Stat. Comput.* 20(1):1–7

Grathwohl W, Choi D, Wu Y, Roeder G, Duvenaud D. 2018. *Backpropagation through the void: optimizing control variates for black-box gradient estimation.* Presented at the Sixth International Conference on Learning Representations, Vancouver, Canada, Apr. 30–May 3

Hammer H, Tjelmeland H. 2008. Control variates for the Metropolis–Hastings algorithm. *Scand. J. Stat.* 35(3):400–14

Henderson SG. 1997. *Variance reduction via an approximating Markov process*. Ph.D. Thesis, Stanford Univ., Stanford, CA

Hickernell F. 1998. A generalized discrepancy and quadrature error bound. *Math. Comput.* 67(221):299–322

Hodgkinson L, Salomone R, Roosta F. 2020. The reproducing Stein kernel approach for post-hoc corrected sampling. arXiv:2001.09266 [math.ST]

Jacob PE, O'Leary J, Atchadé YF. 2020. Unbiased Markov chain Monte Carlo methods with couplings. *J. R. Stat. Soc. Ser. B* 82(3):543–600

Jones GL, Hobert JP. 2001. Honest exploration of intractable probability distributions via Markov chain Monte Carlo. *Stat. Sci.* 16(4):312–34

Knudson C, Vats D. 2020. `stableGR:` a stable Gelman-Rubin diagnostic for Markov chain Monte Carlo. *R Package*, version 1.0. **https://cran.r-project.org/package=stableGR**

Kolmogorov AN. 1956. *Foundations of the Theory of Probability*. New York: Chelsea

Kontoyiannis I, Meyn SP. 2008. Computable exponential bounds for screened estimation and simulation. *Ann. Appl. Probab.* 18(4):1491–518

Kyriazopoulou-Panagiotopoulou S, Kontoyiannis I, Meyn SP. 2008. Control variates as screening functions. In *Proceedings of the 3rd International Conference on Performance Evaluation Methodologies and Tools*, ed. J Baras, C Courcoubetis, pp. 1–9. Brussels: ICST

Liu Q, Lee J. 2017. Black-box importance sampling. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ed. A Singh, J Zhu, pp. 952–61. N.p.: PMLR

Liu Q, Lee J, Jordan M. 2016. A kernelized Stein discrepancy for goodness-of-fit tests. In *Proceedings of the 33rd International Conference on Machine Learning*, ed. MF Balcan, KQ Weinberger, pp. 276–84. N.p.: PMLR

Lunn DJ, Thomas A, Best N, Spiegelhalter D. 2000. WinBUGS – a Bayesian modelling framework: concepts, structure, and extensibility. *Stat. Comput.* 10(4):325–37

Mak S, Joseph VR. 2018. Support points. *Ann. Stat.* 46(6A):2562–92

Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E. 1953. Equations of state calculations by fast computing machines. *J. Chem. Phys.* 12(6):1087–92

Meyn SP, Tweedie RL. 1994. Computable bounds for geometric convergence rates of Markov chains. *Ann. Appl. Probab.* 4(4):981–1011

Meyn SP, Tweedie RL. 2012. *Markov Chains and Stochastic Stability*. New York: Springer

Mijatović A, Vogrinc J. 2018. On the Poisson equation for Metropolis–Hastings chains. *Bernoulli* 24(3):2401–28

Mira A, Solgi R, Imparato D. 2013. Zero variance Markov chain Monte Carlo for Bayesian estimators. *Stat. Comput.* 23(5):653–62

Müller A. 1997. Integral probability metrics and their generating classes of functions. *Adv. Appl. Probab.* 29(2):429–43

Oates CJ, Cockayne J, Briol FX, Girolami M. 2019. Convergence rates for a class of estimators based on Stein's method. *Bernoulli* 25(2):1141–59

Oates CJ, Girolami M, Chopin N. 2017. Control functionals for Monte Carlo integration. *J. R. Stat. Soc. Ser. B* 79(3):695–718

Oates CJ, Papamarkou T, Girolami M. 2016. The controlled thermodynamic integral for Bayesian model evidence evaluation. *J. Am. Stat. Assoc.* 111(514):634–45

Oettershagen J. 2017. *Construction of optimal cubature algorithms with applications to econometrics and uncertainty quantification*. Ph.D. Thesis, Univ. Bonn

Owen AB. 2017. Statistically efficient thinning of a Markov chain sampler. *J. Comput. Graph. Stat.* 26(3):738–44

Paige B, Sejdinovic D, Wood FD. 2016. Super-sampling with a reservoir. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, ed. A Ihler, D Janzing, pp. 567–76. Arlington, VA: AUAI

Papamarkou T, Mira A, Girolami M. 2014. Zero variance differential geometric Markov chain Monte Carlo algorithms. *Bayesian Anal.* 9(1):97–128

Plummer M. 2003. JAGS: a program for analysis of Bayesian graphical models using Gibbs sampling. In *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*, ed. K Hornik, F Leisch, A Zeileis. Vienna: DSC

Plummer M, Best N, Cowles K, Vines K. 2006. CODA: Convergence diagnosis and output analysis for MCMC. *R News* 6(1):7–11

Portier F, Segers J. 2018. Monte Carlo integration with a growing number of control variates. arXiv:1801.01797 [math.ST]

Pronzato L, Zhigljavsky A. 2020. Bayesian quadrature, energy minimization, and space-filling design. *SIAM/ASA J. Uncertainty Quantificat.* 8(3):959–1011

R Core Team. 2020. R: a language and environment for statistical computing. *Statistical Software*, R Found. Stat. Comput., Vienna

Radivojević T, Akhmatskaya E. 2020. Modified Hamiltonian Monte Carlo for Bayesian inference. *Stat. Comput.* 30(2):377–404

Rasmussen CE. 2003. Gaussian processes in machine learning. In *Summer School on Machine Learning*, ed. O Bousquet, U von Luxburg, G Rätsch, pp. 63–71. New York: Springer

Rendl F. 2016. Semidefinite relaxations for partitioning, assignment and ordering problems. *Ann. Operat. Res.* 240:119–40

Riabiz M, Chen W, Cockayne J, Swietach P, Niederer SA, et al. 2021. Optimal thinning of MCMC output. *J. R. Stat. Soc. Ser. B.* In press

Roberts GO, Stramer O. 2002. Langevin diffusions and Metropolis-Hastings algorithms. *Methodol. Comput. Appl. Probab.* 4(4):337–57

Roberts GO, Tweedie RL. 1999. Bounds on regeneration times and convergence rates for Markov chains. *Stochastic Proc. Appl.* 80(2):211–29

Rosenthal JS. 1995. Minorization conditions and convergence rates for Markov chain Monte Carlo. *J. Am. Stat. Assoc.* 90(430):558–66

Roy V. 2020. Convergence diagnostics for Markov chain Monte Carlo. *Annu. Rev. Stat. Appl.* 7:387–412

Salvatier J, Wiecki TV, Fonnesbeck C. 2016. Probabilistic programming in Python using PyMC3. *PeerJ Comput. Sci.* 2:e55

Sard A. 1949. Best approximate integration formulas; best approximation formulas. *Am. J. Math.* 71(1):80–91

Si S, Oates CJ, Duncan AB, Carin L, Briol FX. 2020. Scalable control variates for Monte Carlo methods via stochastic optimization. arXiv:2006.07487 [stat.ML]

South LF. 2020. `ZVCV`: Zero-variance control variates. *R Package*, version 2.1.0. **https://cran.r-project.org/package=ZVCV**

South LF, Karvonen T, Nemeth C, Girolami M, Oates CJ. 2021. Semi-exact control functionals from Sard's method. *Biometrika* 2021:asab036

South LF, Nemeth C, Oates CJ. 2019. Discussion of "Unbiased Markov chain Monte Carlo with couplings" by Pierre E. Jacob, John O'Leary and Yves F. Atchadé. arXiv:1912.10496 [stat.ME]

South LF, Oates CJ, Mira A, Drovandi C. 2018. Regularised zero-variance control variates for high-dimensional variance reduction. arXiv:1811.05073 [stat.CO]

Statisticat LLC. 2021. LaplacesDemon: complete environment for Bayesian inference. *R Package*, version 16.1.6. **https://cran.r-project.org/web/packages/LaplacesDemon/index.html**

Stein C. 1972. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the 6th Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 2, ed. LM Le Cam, J Neyman, EL Scott, pp. 583–602. Berkeley: Univ. Calif. Press

Teymur O, Gorham J, Riabiz M, Oates CJ. 2021. Optimal quantisation of probability measures using maximum mean discrepancy. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, ed. A Banerjee, K Fukumizu, pp. 1027–35. N.p.: PMLR

Vats D, Knudson C. 2018. Revisiting the Gelman-Rubin diagnostic. arXiv:1812.09384 [stat.CO]

Vehtari A, Gelman A, Simpson D, Carpenter B, Bürkner PC. 2021. Rank-normalization, folding, and localization: an improved for assessing convergence of MCMC. *Bayesian Anal.* 16(2):667–718

Wan R, Zhong M, Xiong H, Zhu Z. 2019. Neural control variates for variance reduction. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, ed. K Myszkowski, pp. 1–9. New York: ACM

Wang C, Chen X, Smola AJ, Xing EP. 2013. Variance reduction for stochastic gradient optimization. In *Advances in Neural Information Processing Systems 26*, ed. CJC Burges, L Bottou, M Welling, Z Ghahramani, KQ Weinberger, pp. 181–89. Red Hook, NY: Curran

Wendland H. 2004. *Scattered Data Approximation*. Cambridge, UK: Cambridge Univ. Press

Wenliang LK. 2020. Blindness of score-based methods to isolated components and mixing proportions. arXiv:2008.10087 [stat.ML]

Wolsey LA. 2020. *Integer Programming*. New York: Wiley. 2nd ed.

Yu Y, Meng XL. 2011. To center or not to center: That is not the question—an ancillarity–sufficiency interweaving strategy (ASIS) for boosting MCMC efficiency. *J. Comput. Graph. Stat.* 20(3):531–70