Biometrika (2020), **107**, 1, *pp.* 1–23
Printed in Great Britain

doi: 10.1093/biomet/asz066
Advance Access publication 24 December 2019

The Hastings algorithm at fifty

By D. B. DUNSON

Department of Statistical Science, Duke University, Box 90251, Durham, North Carolina 27707, U.S.A.

dunson@duke.edu

and J. E. JOHNDROW

Department of Statistics, The Wharton School, University of Pennsylvania, 3730 Walnut St, Philadelphia, Pennsylvania 19104, U.S.A.

johndrow@stanford.edu

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024

Summary

In a 1970 *Biometrika* paper, W. K. Hastings developed a broad class of Markov chain algorithms for sampling from probability distributions that are difficult to sample from directly. The algorithm draws a candidate value from a proposal distribution and accepts the candidate with a probability that can be computed using only the unnormalized density of the target distribution, allowing one to sample from distributions known only up to a constant of proportionality. The stationary distribution of the corresponding Markov chain is the target distribution one is attempting to sample from. The Hastings algorithm generalizes the Metropolis algorithm to allow a much broader class of proposal distributions instead of just symmetric cases. An important class of applications for the Hastings algorithm corresponds to sampling from Bayesian posterior distributions, which have densities given by a prior density multiplied by a likelihood function and divided by a normalizing constant equal to the marginal likelihood. The marginal likelihood is typically intractable, presenting a fundamental barrier to implementation in Bayesian statistics. This barrier can be overcome by Markov chain Monte Carlo sampling algorithms. Amazingly, even after 50 years, the majority of algorithms used in practice today involve the Hastings algorithm. This article provides a brief celebration of the continuing impact of this ingenious algorithm on the 50th anniversary of its publication.

Some key words: Bayesian computation; Importance sampling; Markov chain Monte Carlo; Metropolis–Hastings; Posterior sampling; Rejection sampling.

1. Introduction

It is common in many fields to study different characteristics of a probability distribution $f(\cdot)$. We will overload notation in using f to refer to both a probability measure and a density. If $f(x)$ is simple then one can often analytically calculate different features of $f(x)$; for example, these may include the mean, the median, the probability that $f(x)$ assigns to a set of interest S, an interval or region R that is assigned probability $(1-\alpha)$, and so on. However, in many cases $f(x)$ is complicated, and analytically calculating these features may be difficult if not impossible. In such cases one can appeal to numerical integration, but then issues arise in terms of accuracy, stability and ability to scale beyond low-dimensional cases. A broad class of solutions is provided by Monte Carlo algorithms, which estimate features of $f(x)$ on the basis of samples $x_t \sim f$, for

© 2019 Biometrika Trust

$t = 1, \ldots, T$. For example, we can estimate the mean of $f(x)$ as $T^{-1}\sum_{t=1}^{T} x_t$, with the variance of the estimate decreasing at rate T^{-1}.

The key challenge with Monte Carlo methods is how to efficiently generate samples from a target probability distribution $f(x)$. In the univariate case many approaches are available, with the inverse cumulative distribution function algorithm being particularly popular, see Devroye (1986, Ch. 2). However, sampling from arbitrary multivariate distributions remains a challenging problem. Rejection sampling attempts to solve this problem by generating candidate samples from a simpler distribution $g(x)$, and then accepting the draws with probability $f(x)/\{Mg(x)\}$, with M a finite constant chosen so that $f(x) \leqslant Mg(x)$. Key problems with rejection sampling include how to find a $g(x)$ that is easy to sample from, that can be shown to satisfy this property for finite M, and that has a reasonably high acceptance rate. Solving these problems has proven to be effectively intractable. There is a rich literature on improving such approaches by adapting $g(x)$ to be more accurate as sampling proceeds, among other ideas (Robert & Casella, 2013).

An alternative is provided by Markov chain Monte Carlo algorithms (Tierney, 1994), which define a Markov chain $\{x_t\}_{t=1}^{T}$ starting at an initial value x_0, and then sequentially apply a transition kernel $\mathcal{K}(x_t; x_{t-1})$ that samples a new value x_t conditionally on x_{t-1}, but independently of $x_0, \ldots, x_{t-2}$. There are several popular books dedicated partly or entirely to this method, including Gamerman & Lopes (2006), Liu (2008), Robert & Casella (2009) and Brooks et al. (2011). If $\mathcal{K}$ is chosen appropriately, then the samples $\{x_t\}$ converge to a stationary distribution corresponding to the target $f(x)$. Because it is not possible in most cases to start the chain from stationarity, the common practice is to discard a number of initial burn-in samples, and then calculate Monte Carlo summaries of $f(x)$ based on the subsequent samples. If the burn-in is selected appropriately, then the retained samples are approximately a dependent sequence from the target distribution. Such algorithms are particularly popular in Bayesian inference where $f(x) = \pi(x)L(x)/\int_{\mathcal{X}} \pi(z)L(z)\,\mathrm{d}z$, with $\pi(x)$ a prior density for x, $L(x)$ a likelihood function and $C = \int_{\mathcal{X}} \pi(z)L(z)\,\mathrm{d}z$ a normalizing constant referred to as the marginal likelihood.

A recipe for defining an appropriate transition kernel $\mathcal{K}$ was provided by Metropolis et al. (1953), who proposed a simple approach building on rejection sampling. One first samples a candidate $\tilde{x}$ for x_t from a proposal density $g(\tilde{x}; x_{t-1})$ that is symmetric: $g(\tilde{x}; x_{t-1}) = g(x_{t-1}; \tilde{x})$. Conditional on the proposal, the next state x_t takes the value $x_t = \tilde{x}$ with probability equal to $\alpha(\tilde{x}_{t-1}, \tilde{x}) = 1 \wedge f(\tilde{x})/f(x_{t-1})$, where $a \wedge b$ is the minimum of a and b, and $x_t = x_{t-1}$ otherwise. Under weak regularity conditions, the resulting samples $\{x_t\}$ converge to the stationary distribution $f(x)$, and hence can be used to obtain Monte Carlo estimates of features of $f(x)$. Although it builds upon rejection sampling, the Metropolis algorithm has the important advantage that there is no need for an accurate global approximation $g(x)$ to $f(x)$, as the proposal density is local as it depends on the previous state x_{t-1}. The most common Metropolis algorithm uses the normal random walk proposal, for which $\tilde{x} \sim N(\tilde{x}; x_{t-1}, \Sigma)$, with Σ a covariance matrix that can be tuned to improve efficiency.

Although the Metropolis algorithm was an enormous advance that was well ahead of its time, the symmetry constraint on the proposal density $g(\tilde{x}; x_{t-1})$ is a major practical limitation greatly restricting flexibility in the design of useful sampling algorithms. Hastings (1970) proposed a remarkable algorithm that provides a simple adjustment for asymmetry. In particular, the only modification over the Metropolis algorithm is to accept $\tilde{x}$ with probability

$$\min\left\{1, \frac{f(\tilde{x})}{f(x_{t-1})} \times \frac{g(x_{t-1}; \tilde{x})}{g(\tilde{x}; x_{t-1})}\right\}, \tag{1}$$

where the first term in the product is the Metropolis acceptance probability and the second term adjusts for asymmetry of g. This outwardly simple algorithm is deceptively rich, and remains

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024

the most popular general-purpose Markov chain Monte Carlo algorithm. For example, popular software packages such as `Stan` (Carpenter et al., 2017) rely on the Hastings algorithm, using proposals obtained by discretized Hamiltonian dynamics (Duane et al., 1987; Neal, 2013). The combination of the Hastings algorithm with Gibbs sampling (Geman & Geman, 1984; Gelfand & Smith, 1990) through the Metropolis-within-Gibbs algorithm (Tierney, 1991; Gilks et al., 1995) leads to an exceptionally flexible framework for computation. The literature continues to generate new special cases at a torrid pace, and there is no sign of the Hastings algorithm decreasing in impact and importance even 50 years after its initial publication. Hastings remains the most common approach in modern Bayesian computation, even with the rise of competitors such as sequential Monte Carlo (Del Moral et al., 2006) and variational approximations (Attias, 1999; Jordan et al., 1999; Wainwright & Jordan, 2008; Blei et al., 2018).

The Metropolis algorithm is ranked among the 10 most important algorithms of the 20th century (Dongarra & Sullivan, 2000). It emerged as a solution to problems in physics that were of particular interest during the construction of the hydrogen bomb in the late 1940s and early 1950s. The development of the algorithm is intertwined with the history of computing, in being one of the first programs written at Los Alamos for an early computer with the unwieldy name ‘mathematical analyzer, numerical integrator, and computer’. While greatly expanding the scope of the Metropolis algorithm, Hastings’ additional contribution was to bring this class of algorithms to the attention of the statistics community. Interestingly, Hastings’ paper did not mention applications to Bayesian statistics, which is one of the most prominent application areas. Indeed, it is no exaggeration that the Metropolis–Hastings algorithm, and its extension to the Metropolis-within-Gibbs sampler, transformed Bayesian statistics from a theoretical curiosity, for which computation was largely infeasible outside of toy models, to its modern place as the inferential paradigm of choice in many applications where complex hierarchical models are desirable.

There have been a variety of review articles on the Hastings algorithm, including some aimed at practitioners (Chib & Greenberg, 1995) or theoreticians (Diaconis & Saloff-Coste, 1998), and some providing a historical perspective (Hitchcock, 2003). There are also many books on theory and practice in Bayesian statistics, including detailed descriptions of a variety of important special cases (Gamerman & Lopes, 2006; Liu, 2008; Robert & Casella, 2010; Brooks et al., 2011). The goal of this article is not to provide a comprehensive review of the literature building on Hastings’ 1970 paper, but instead to celebrate his remarkable article on the 50th anniversary of its publication. This celebration will take the form of highlighting some key advances building on the original work, while focusing on recent and ongoing areas of research.

We pay particular attention to issues of scalability, which have become increasingly important in recent years amidst concerns over the ability of Markov chain algorithms to compete with optimization for speed on large and complex datasets. These issues are fundamentally bound up with questions about the ongoing relevance of statistical inference and uncertainty quantification in modern applications. It is perhaps no surprise that we come down firmly in favour of the ongoing importance of Markov chain Monte Carlo and uncertainty quantification, particularly given the growing concerns about reproducibility of science.

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024

2. Some key historical developments

2.1. *Overview*

The amazing generality of the Hastings algorithm comes with a price. In particular, although one can conceptually use any proposal g when implementing the algorithm, the key question is how to choose a good proposal having high computational efficiency? Computational efficiency is controlled by two factors: (i) the computational cost per iteration of the sampler, and (ii) the mixing rate of the Markov chain $\{x_t\}$. Factor (i) is dependent on the cost of sampling a proposal from g and calculating the acceptance probability (1). Factor (ii) arises because the samples are not independent, but are generated from a Markov chain, so that x_t and $x_{t+\Delta}$ are correlated. For a slow mixing chain, the correlation between x_t and $x_{t+\Delta}$ will tend to decrease slowly in the lag Δ, and T samples $\{x_t\}_{t=1}^{T}$ may contribute much less information about summaries of interest than would be available in T independent samples. This discrepancy from independent sampling is typically quantified via the effective sample size, T_{ESS}. The following subsections detail some key historical developments building on the Hastings algorithm motivated by computational efficiency, generalizability and reducing the need for tuning of the proposal distribution to achieve rapid mixing.

2.2. *Gibbs sampling*

One of the key developments in the history of Markov chain Monte Carlo is the Gibbs sampler (Besag, 1974; Geman & Geman, 1984; Gelfand & Smith, 1990; Casella & George, 1992). Supposing that $x = (x^j, j = 1, \ldots, p)$ is p-dimensional, each iteration of the Gibbs sampler consists of p substeps that update each of the elements of x sequentially from their conditional probability distribution under $f(x)$, holding all other elements fixed at their current value. In particular, the jth substep draws x_t^j from the conditional probability distribution of x^j given $x^l = x_t^l$ for $l = 1, \ldots, j-1$ and $x^l = x_{t-1}^l$ for $l = j+1, \ldots, p$. This conditional distribution can be derived from $f(x)$ as $f(x^j \mid x^{-j}) = f(x)/f(x^{-j})$, where $x^{-j} = (x^1, \ldots, x^{j-1}, x^{j+1}, \ldots, x^p)$ denotes the vector x, excluding the jth element.

There are several appealing practical properties of the Gibbs sampler relative to generic Hastings algorithms: (i) in the multivariate case, it can be challenging to construct good proposals g when p is not small, but Gibbs solves this problem by sampling directly from the conditional distributions of each of the elements of x, one at a time; (ii) the resulting algorithm is free of tuning parameters and there is no need to tweak certain algorithmic parameters, such as the proposal covariance Σ, to obtain good performance; (iii) in many cases, even when the multivariate density $f(x)$ is highly complex, the univariate conditional distributions have simple forms that are easy to sample from. This is particularly the case in many Bayesian hierarchical modelling applications. Even when the conditionals are not members of a family for which direct sampling algorithms are available, because each of the full conditionals is univariate, many more options are available for creating efficient sampling algorithms.

2.3. *Metropolis-within-Gibbs and other extensions*

The Gibbs sampler has the limitation that one must sample from the conditional probability distribution of each element of x sequentially. In many cases one or more of these conditionals may not be in a form that is convenient for sampling. To solve this problem one can consider a broader class of Metropolis-within-Gibbs algorithms that replace sampling directly from the conditional distribution $f(x^j \mid x^{-j}) = f(x)/f(x^{-j})$ with a Hastings step targeting $f(x^j \mid x^{-j})$. In particular, in the jth substep we generate a proposal $\tilde{x}^j$ for the jth element of x and accept with probability (1), calculated treating all other elements x^{-j} as fixed at their current values. The Gibbs sampler corresponds to the special case of Metropolis-within-Gibbs, in which the proposal density for each $\tilde{x}^j$ corresponds to the conditional distribution $f(x^j \mid x^{-j})$, as in this case the acceptance probability is one. In practice, one can use different types of proposals for each element of x, for example using Gibbs steps for some and normal random walk steps for others, leading to a large class of possible algorithms.

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024

In practice, Gibbs sampling is not necessarily more efficient than Metropolis–Hastings algorithms that update all the elements of x simultaneously, since updating elements one at a time can lead to slow mixing when there is a high degree of dependence between x^j and x^l in $f(x)$. One approach to improve mixing is to reparameterize so that one samples $z = h(x)$ instead of x, with the reparameterizaton function h chosen so that the elements of z have relatively low dependence. One class of such reparameterizations is referred to as preconditioning, with the idea being that if one can obtain an approximation to $\Sigma = \text{cov}_f(x)$, then $z = \Sigma^{-1/2}x$ would provide a good correlation-reducing reparameterization.

Another effective strategy for improving mixing of the Markov chain is to update highly correlated elements of x together within blocks (Amit & Grenander, 1991). This motivates a blocked version of Metropolis-within-Gibbs, which first chooses blocks $B_1, \ldots, B_k$ such that $\bigcup_{h=1}^{k} B_h = \{1, \ldots, p\}$ and $B_h \bigcap B_l = \emptyset$ for all $h \neq l$. One then updates $x^{B_h} = \{x^l : l \in B_h\}$ jointly from a Metropolis–Hastings or Gibbs substep for $h = 1, \ldots, k$. In some cases it is advantageous to choose a blocking scheme that is not a partition. While blocking often improves convergence of the Markov chain, it can also worsen it (Roberts & Sahu, 1997).

One of the appealing attributes of a Gibbs sampler is the avoidance of tuning. This has motivated a rich literature developing Gibbs samplers for particular models. One possibility is to choose a conditionally conjugate prior density $\pi(x)$, so that conditional posterior distributions have forms that are simple to sample from. An alternative is to use data augmentation (Tanner & Wong, 1987). One starts by introducing a joint density $w(y)$, with $y = (z, x)$, $z \in \mathcal{Z}$ are auxiliary variables introduced for algorithmic purposes, and $w(y)$ is chosen so that $f(x) = \int_{\mathcal{Z}} w(z, x)\, \mathrm{d}z$. One then draws samples $\{y_t = (z_t, x_t)\}$ having stationary distribution $w(y)$, and the sampled $\{x_t\}$ values have stationary distribution $f(x)$. Although one increases dimensionality through incorporating z, if this is done carefully it can lead to a data augmentation Gibbs sampler that has simple steps for updating each of the elements of y_t in blocks from their respective conditional distributions. This is a common strategy for generalized linear models, where the idea is to introduce augmented variables so that the full conditional of the regression coefficients β have a multivariate normal distribution. These include data augmentation samplers for probit (Albert & Chib, 1993) and logistic regression models (Holmes & Held, 2006; Frühwirth-Schnatter & Frühwirth, 2010; Polson et al., 2013).

Slice samplers (Swendsen & Wang, 1987; Neal, 2003) represent another clever use of auxiliary variables to enable Gibbs sampling. Suppose we wish to sample from a density

$$f(x) = f_0(x) \prod_{j=1}^{J} f_j(x).$$

Slice sampling augments the state variable with an auxiliary variable U so that the joint density of (X, U) has the form $f_0(x) \prod_{j=1}^{J} \mathbf{1}\{u_j \leqslant f_j(x)\}$. One then defines a Gibbs sampler that alternates sampling $u_j \mid x \overset{\text{ind}}{\sim} \text{Un}\{0, f_j(x)\}$ with sampling $x \mid u$. Slice sampling is a common way to handle Gibbs sampling steps in Bayesian statistics for which the chosen prior is not conditionally conjugate. The elliptical slice sampler (Murray et al., 2010) is a generalization that is popular in the Gaussian process literature and has been successful for multivariate Gaussian likelihoods with general prior structures (Hahn et al., 2019). The slice sampler has been shown to converge to the target at an exponential rate under very general conditions (Roberts & Rosenthal, 1999).

2.4. *Adaptive algorithms*

Adaptive rejection sampling is an accelerated version of ordinary rejection for sampling from log-concave, univariate densities (Gilks & Wild, 1992). Like many generic rejection sampling

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024

algorithms, it uses piecewise linear upper and lower bounds on the density to construct good proposals and achieve high acceptance rates at low computational cost. In adaptive rejection sampling the log-concavity of the target is used to create the bounding functions using derivatives. The algorithm begins with a set of k initial points at which the derivative is computed, leading to a piecewise linear bounding function with $k + 1$ pieces. The set of points and, consequently, the number of linear pieces in the bounding functions, is gradually expanded as the algorithm runs to quickly build up an accurate piecewise linear approximation to the target density. The algorithm is often efficient, but is restricted to univariate densities. It is therefore natural to use adaptive rejection sampling within a Gibbs sampler where each coordinate is updated conditional on all the others (Gilks & Wild, 1992).

Adaptive Metropolis is a non-Markovian version of the ordinary Metropolis algorithm in which the entire history of the chain is used to gradually refine the proposal distribution (Haario et al., 2001). It is related to the Robbins–Monro stochastic control algorithm, and is most often used in the context of multivariate Gaussian random walk proposals $\tilde{x} \sim N(x_t, \Sigma_t)$, with the proposal covariance being time dependent. An initial guess at a good proposal covariance Σ_0 is made, and the algorithm is run for an initial t_0 steps using proposal $N(x_t, \Sigma_0)$. At this point adaptation begins, with $\Sigma_t = s_p \operatorname{cov}(x_0, \ldots, x_{t-1}) + s_p \epsilon I_p$ for a dimension-dependent scaling parameter s_p which is usually proportional to the inverse of the dimension p^{-1}, and ϵ a small tuning constant. The algorithm can have dramatically improved performance relative to isotropic random walk with optimal scaling $s_p = (2.4)^2/p$ for multidimensional targets. Although the algorithm is not a Markov chain, it still can be shown to give valid Monte Carlo estimates with respect to the target (Atchadé & Rosenthal, 2005) so long as the algorithm satisfies certain technical conditions, for example the containment and diminishing adaptation conditions of Roberts & Rosenthal (2007).

2.5. *Gradient-based algorithms*

In recent years the focus has been increasingly moving away from Gibbs samplers and back towards the use of classical Hastings algorithms, but with better design of proposals. The key disadvantage of most Metropolis-within-Gibbs algorithms is the need to develop algorithms on a case-by-case basis. Ideally one would have general algorithms for automatically sampling from broad classes of distributions; one application is in probabilistic programming software for routine implementation of Bayesian inference. Perhaps the first serious attempt at such a package was `BUGS`, and its later instantiation `WinBUGS`, which relied largely on Gibbs sampling, using adaptive rejection sampling when possible. The disadvantages of this approach include: (i) relying on one-at-a-time updating, as without automated approaches for correlation reduction such as preconditioning, there is substantial risk of slow mixing and corresponding inefficiency; (ii) adaptive rejection sampling can be expensive, involving many sampling steps to produce a single sample from a conditional distribution; and (iii) often, adaptive rejection sampling cannot be used, so one requires some other Metropolis–Hastings step that may need tuning.

In light of these issues, recent Bayesian probabilistic programming software, including `Stan` (Carpenter et al., 2017), `PyMC` (Salvatier et al., 2016) and `Nimble` (de Valpine et al., 2017), has focused largely on gradient-based Markov chain Monte Carlo algorithms. The most popular gradient-based algorithms are the Metropolis-adjusted Langevin algorithm (Roberts & Tweedie, 1996) and Hamiltonian Monte Carlo (Duane et al., 1987; Neal, 2013). These algorithms are similar in both their reliance on gradients to construct proposals and their use of differential equations to harness this gradient information efficiently.

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024

The Metropolis-adjusted Langevin algorithm uses discrete-time approximations of the overdamped Langevin diffusion, which is defined by the stochastic differential equation

$$\dot{x} = \nabla \log f(x) + \sqrt{2}\dot{w}, \tag{2}$$

where w is a Brownian motion on $\mathbb{R}^p$. Langevin dynamics have invariant distribution f, so that as $t \to \infty$, the distribution of x_t evolving according to (2) converges to f. Thus, if it were possible to exactly simulate from (2), one could use sample paths after some appropriate burn-in to estimate expectations with respect to f. Although exact simulation is not possible, a discrete approximation can be constructed using the Euler method with step size ϵ as

$$x_t = x_{t-1} + \epsilon\nabla \log f(x_{t-1}) + (2\epsilon)^{1/2}\xi_t, \quad \xi_t \overset{\text{iid}}{\sim} N(0, I_p). \tag{3}$$

The discrete-time dynamics no longer preserve f, and so x_t is used as a proposal in a Hastings step. When $\epsilon \approx 0$ (3) is close to (2) and the proposal is accepted with high probability, but rarely proposes to move far from the current state. Conversely, when ϵ is large the proposals will tend to be far from the current state, but will be frequently rejected. Thus, one seeks to find values of ϵ that avoid slow mixing due to either small step sizes or low acceptance probabilities. The scaling limit literature indicates that the optimal acceptance probability is approximately 0.57, and that the mixing time is of order $p^{1/3}$ (Pillai et al., 2012), which compares favourably to random walk Metropolis, which has mixing time scaling linearly in dimension.

Hamiltonian Monte Carlo algorithms also rely on a differential equation to generate proposals, but through an approximating Hamiltonian instead of Langevin dynamics. These dynamics are deterministic, unlike the stochastic dynamics of the Langevin algorithm. Another important difference in Hamiltonian Monte Carlo is that the p-dimensional state space is augmented with p additional momentum variables z to facilitate sampling. We then specify a Hamiltonian function $H(x, z) = U(x)+K(x, z)$. The Hamiltonian dynamics will preserve the distribution $\mathrm{e}^{-H(x,z)}$, and hence by letting $U(x) = -\log\{f(x)\}$, and choosing $K(x, z)$ such that $\mathrm{e}^{-K(x,z)}$ is a valid conditional density for $z \mid x$, the invariant distribution will have x-marginal distribution $f(x)$. Typically one lets $K(x, z) = \log|M| + z'M^{-1}z/2$, making the momenta multivariate Gaussian independent of x, but other choices can be considered to improve performance. Riemannian Hamiltonian Monte Carlo allows the matrix M to depend on x so that $K(x, z) = \log|M(x)| + z'M(x)^{-1}z/2$ (Girolami & Calderhead, 2011). This naturally accommodates local variation in the geometry of the target. Nishimura et al. (2017) use a Laplace momentum in developing generalizations to discrete and discontinuous targets, while Livingstone et al. (2019) study how good choices of momenta relate to the tails of the target.

Hamiltonian dynamics are defined by the differential equations

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = \frac{\partial H}{\partial z_i}, \quad \frac{\mathrm{d}z_i}{\mathrm{d}t} = -\frac{\partial H}{\partial x_i}.$$

As exact solutions are typically unavailable, discrete-time approximations are used. A popular choice is the leapfrog algorithm, which approximates the dynamics across an s time window by a series of linear steps of size ϵ. In posterior sampling, a single step proceeds by: (1) sampling the momentum $z \mid x \sim \mathrm{e}^{-K(x,z)}$; (2) beginning at (x, z), approximately simulating Hamiltonian dynamics for time s via s/ϵ iterations of the leapfrog algorithm to obtain a new state (x^*, z^*); and (3) using (x^*, z^*) as a proposal in a Hastings step. Because leapfrog dynamics are reversible and volume preserving, there is no need to compute the forward and backward conditional densities. More details can be found in the excellent review of Neal (2013).

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024

While Langevin-based algorithms are popular in the molecular simulation and machine learning literature even without Hastings correction, in the statistics literature Hamiltonian Monte Carlo has received more attention. One reason is that while both algorithms use gradients to create proposals, the underlying dynamics are fundamentally different. Langevin dynamics tend to move to higher values of $f(x)$, since $\nabla \log f(x)$ always points in the direction of increasing target density, while Hamiltonian dynamics move along level sets of $H(x, z)$. However, because the momentum is randomly sampled at the start of each iteration, and because level sets of $H(x, z)$ may span many different levels of the marginal potential $f(x) = \mathrm{e}^{-U(x)}$, the dynamics are richer and the algorithm has less of a tendency to propose very close to the mode, instead spending most of its time in a region that contains the bulk of the target's mass (Betancourt & Girolami, 2015). This results in better Monte Carlo estimates. Another reason for the popularity is the appearance of tuning-free versions of Hamiltonian Monte Carlo, such as the no-U-turn sampler (Hoffman & Gelman, 2014), and its use in the popular `Stan` package.

2.6. *Theory of Markov chains*

Although our focus is mostly practical, we would be remiss not to mention the theory of Hastings algorithms and Markov chains in general, given the close relationship between theory and practice in this field. We focus here on a brief and very incomplete summary of the literature on convergence rates. The most popular approach for studying convergence is the use of Lyapunov drift functions to control tail behaviour combined with minorization or coupling arguments on sublevel sets of the Lyapunov function. These techniques date at least to Meyn & Tweedie (1993), and were prevalent in the earlier stochastic dynamics literature (Khasminskii, 1980). Important early applications of these techniques to Markov chain Monte Carlo include Rosenthal (1995), which gave general conditions for geometric ergodicity, meaning that the Markov chain converges to the target distribution at an exponential rate in the number of steps. Mengersen & Tweedie (1996) applied these techniques to show conditions under which random walk Hastings algorithms are geometrically ergodic. One such condition is that, roughly speaking, the tails of the target must be lighter than exponential; this condition can be weakened considerably by employing transformations (Johnson & Geyer, 2012). Similar results exist for the Metropolis-adjusted Langevin algorithm (Roberts & Tweedie, 1996; Livingstone et al., 2019; Bou-Rabee & Sanz-Serna, 2017; Durmus et al., 2019). Other work has given general conditions for polynomial convergence rates (Fort & Moulines, 2003; Jarner & Tweedie, 2003; Douc et al., 2004), again using notions of a drift function. Another area of theory that has had great impact on practice is the literature on optimal scaling and acceptance rates for random walk, Langevin and Hamiltonian proposals, which we discuss in detail in § 4.2.

3. Challenging applications

3.1. *Multimodal targets*

When the target $f(x)$ is nice, for example log-concave or unimodal, the algorithms discussed thus far can work reasonably well in most small- to moderate-p problems. However, many $f(x)$ are not nice, for example due to multimodality. In Bayesian inference, multimodality of posterior distributions is common, particularly in mixture models and variable or model selection contexts. For multimodal $f(x)$, the available theoretical guarantees for the Hastings algorithm are much weaker, and any of the flavours of Hastings discussed in the previous section can fail to mix well. Even gradient-based methods face energy barriers in moving between modes. This has motivated a rich literature on algorithms designed to improve mixing, including simulated, parallel and

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024

geometric tempering, the equi-energy sampler (Kou et al., 2006), split–merge samplers, and the recent birth–death algorithm of Lu et al. (2019).

A common application in Bayesian statistics that often leads to multimodal targets is variable dimension mixture modelling, which involves uncertainty in the number of components in the mixture. In such cases one can define a joint posterior distribution over the number of mixture components k and the component-specific model parameters $x_{1:k}$. A Metropolis–Hastings sampler can be implemented by defining a proposal distribution that generates new values for $(k, x_{1:k})$. However, it is not valid to apply the usual Hastings acceptance probability when x varies in dimensionality with k. To address such problems, Green (1995) proposed an adjustment to the acceptance probability that takes into account the change in dimension. Split-merge samplers represent a particular class of reversible jump algorithms designed for efficient computation for Bayesian models involving an unknown partitioning of $\{1, \ldots, n\}$ into k groups, where k may be unknown. This occurs in variable-dimension mixtures, which include a component index $z_i \in \{1, \ldots, k\}$ for each subject i, for $i = 1, \ldots, n$. Subjects having the same values of z_i are in the same cluster. Markov chain Monte Carlo algorithms commonly rely on Gibbs steps for updating each z_i from its conditional posterior distribution, often resulting in slow mixing, particularly when n is large. Split-merge samplers instead construct proposals for directly modifying the partition by splitting or merging components (Green & Richardson, 2001). As in other applications of reversible jump Metropolis–Hastings, a major challenge is defining efficient proposals. One approach uses restricted Gibbs sampling scans to create split and merge proposals, which are then accepted or rejected using an ordinary Hastings step (Jain & Neal, 2004). The improvement in mixing can be very dramatic, especially for high-dimensional mixtures.

Tempering considers sampling from $f(x)^{(1/T)}$, where $T \geqslant 1$ is the temperature. Raising the target to a fractional power reduces the depths of troughs and the heights of peaks in f. This makes it easier for even simple random walk proposals to transition between modes, since it is no longer extremely unlikely to visit points in the troughs separating the modes. As $f(x)^{(1/T)}$ is only the target of interest when $T = 1$, in practice a series of temperatures $1 = T_1, \ldots, T_m$ is chosen. One such algorithm is simulated tempering (Marinari & Parisi, 1992; Geyer & Thompson, 1995), which runs a Markov chain on an augmented space $\mathcal{X} \times \{1, \ldots, m\}$ with target distribution

$$f^*(x,j) \propto C_j f_j(x), \tag{4}$$

where C_j is the pseudo-prior probability of the temperature parameter taking value T_j. The samples of x are not all from a Markov chain targeting $f(x)$, but one can save only those samples of x such that the temperature is equal to one. This discards all but a small proportion of the samples when m is not small, and an alternative is to use an importance-weighted average of all the samples, known as importance tempering (Gramacy et al., 2010). Parallel tempering (Geyer, 1991; Hukushima & Nemoto, 1996) instead runs m Markov chains in parallel, each evolving according to a transition kernel K_j targeting f_j, and periodically proposes to exchange states with neighbouring chains on the temperature ladder. This can be much more efficient than simulated tempering in parallel computing environments. It is worth noting that tempering does not always work; it can fail when some modes are much more concentrated than others, with the result that weights on the modes of the tempered target are very sensitive to temperature. Woodard et al. (2009a,b) give conditions for torpid and rapid mixing, respectively. Tawn et al. (2019) improves performance of tempering in cases where the weights are sensitive to temperature by stabilizing the weights on the modes while still allowing the target density to spread out as temperature increases.

An illustration of the massive improvements that are possible with tempering is shown in Fig. 1. We show single paths of length 5 million from four algorithms targeting $f(x) = 1/2$

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024

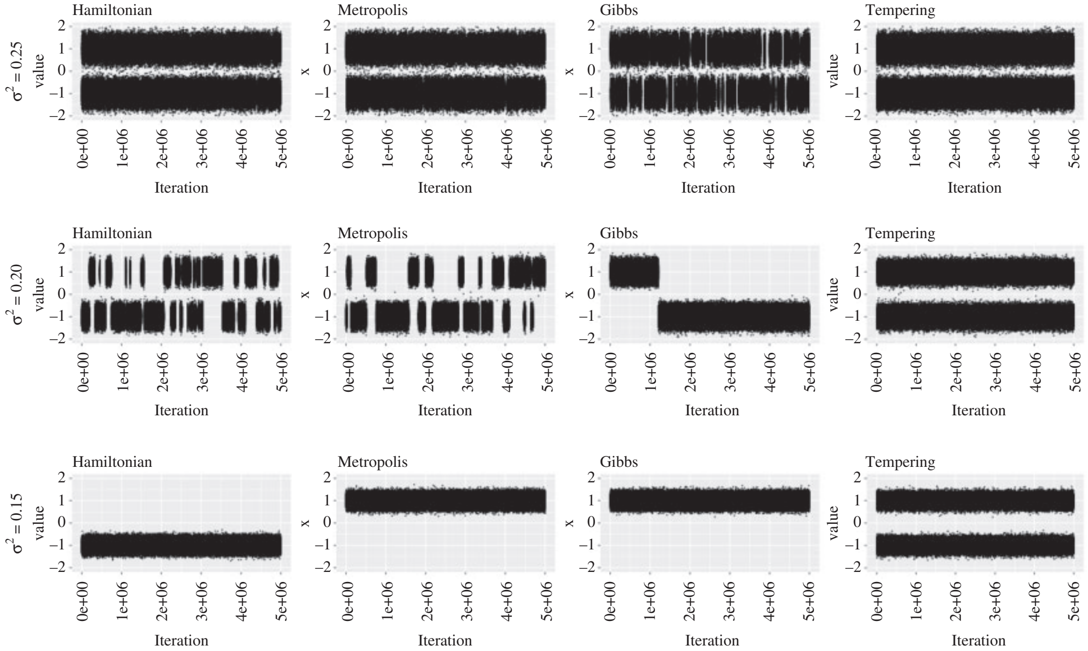

Fig. 1. Sample paths of length 5×10^6 from the four algorithms targeting the mixture of two normals as described in § 3.1. Each column of the figure corresponds to one algorithm, as indicated by the column heading. The values of σ^2 are indicated on the left of each row.

Table 1. *Effective sample sizes for the samplers shown in Fig. 1 based on 5 million iterations for a multimodal target*

	Hamiltonian	Metropolis	Gibbs	Tempering
$\sigma = 0.25$	32065	31774	31456	47413
$\sigma = 0.20$	30637	30585	31004	45206
$\sigma = 0.15$	1847220	617926	4987130	44232

$\phi\{(x-1)/\sigma\} + \frac{1}{2}\phi\{(x+1)/\sigma\}$, where $\phi(\cdot)$ is the standard normal density function. This simple example of a bimodal target is studied in Mangoubi et al. (2018), which shows that optimally tuned Hamiltonian Monte Carlo and random walk Metropolis both have mixing times that scale like $e^{\frac{1}{2}\sigma^{-2}}$ as $\sigma \rightarrow 0$. This is illustrated by the three simulations in Fig. 1 for $\sigma^2 = 0.25$, $\sigma^2 = 0.20$ and $\sigma^2 = 0.15$. In addition to Hamiltonian Monte Carlo implemented in `Stan` and random walk Metropolis, we also show paths from a data augmentation Gibbs sampler. All these samplers have reasonable performance for $\sigma = 0.25$, and are able to transition between the two modes at an acceptably fast rate. For $\sigma = 0.2$ the performance of all three is lacklustre, with at most a few dozen transitions observed in 5 million iterations, with data augmentation Gibbs being noticeably worse. For $\sigma = 0.15$ none of these algorithms experiences a single transition between modes in 5 million iterations. In contrast, an implementation of parallel tempering with a nine-component geometric temperature ladder and random walk Metropolis base sampler is able to easily transition between modes for all three values of σ.

Multimodal targets also expose the shortcomings of many approaches for assessing convergence. Table 1 shows estimates of the effective sample size for the 12 simulations described above. The effective sample size is a measure of computational efficiency that is inversely proportional to the asymptotic variance of the time-averaging estimator of a parameter. The asymptotic variance

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024

can be estimated from a finite-length path using a variety of methods, ideally after discarding a substantial number of burn-in samples. Some good references on estimation of standard errors for Markov chain Monte Carlo are Flegal et al. (2008) and Flegal & Jones (2010). Here, we have estimated the effective sample sizes using the overlapping batch means estimator with $T^{1/3}$ sample size implemented in the R package mcmcse. The effective sample sizes are roughly equal for $\sigma = 0.25$ and $\sigma = 0.2$ for Hamiltonian Monte Carlo, random walk Metropolis and Gibbs, but are much higher for all three of these algorithms when $\sigma = 0.15$. Perversely, the effective sample size for tempering appears much smaller than the other algorithms when $\sigma = 0.15$, even though in reality it is the only useful algorithm of the four in that case. This occurs because the effective sample size is essentially measuring the extent of autocorrelation, and if the path never hits the second mode, the empirical autocorrelations are quite low. Effective sample size and autocorrelation are more useful for estimating standard errors when one is reasonably sure that the algorithm has nearly converged than as ways to assess convergence. In this example, convergence diagnostics that involve running multiple chains from different starting points, or that simulate from couplings originated at distant points, would have a good chance of detecting nonconvergence. There are a considerable number of such methods, both old (Gelman & Rubin, 1992; Brooks & Gelman, 1998; Johnson, 1996, 1998) and new (Biswas et al., 2019).

3.2. *Intractable likelihoods*

In Bayesian modelling it is not uncommon for the likelihood function $L(x)$ to be intractable, meaning it is not computable even up to a normalizing constant. Such applications are referred to as doubly intractable, since one can neither compute the marginal likelihood of the data nor the conditional likelihood of the data given parameters. This leads to problems in implementing the Hastings algorithm, as the acceptance probability in (1) requires evaluating the likelihood function for the candidate and current values of x up to a normalizing constant. Intractable likelihoods arise in a variety of contexts. For example, the likelihood may involve an infinite sum or intractable integral, such as when evaluation of the likelihood requires the solution of a differential equation. Murray et al. (2006) and Møller et al. (2006) propose an exchange algorithm that generalizes the usual Hastings algorithm to allow intractable likelihoods using an auxiliary variable scheme. Rao et al. (2016) propose an alternative approach for cases in which the likelihood can be induced by rejection sampling.

Pseudo-marginal Metropolis Hastings provides an algorithm that generalizes the acceptance probability (1) in the Hastings algorithm to allow one to use an unbiased estimate of the likelihood function in place of the exact likelihood, and still maintain convergence guarantees to stationary distribution $f(x)$ (Andrieu & Roberts, 2009). In pseudo-marginal algorithms the likelihood estimate is usually constructed via sampling. For example, Stoehr et al. (2019) proposed a Monte Carlo based approach to approximating Hamiltonian dynamics in cases where the likelihood is intractable, extending Hamiltonian Monte Carlo to doubly intractable problems. Among the most popular examples of a pseudo-marginal Metropolis–Hastings algorithm is particle Markov chain Monte Carlo (Andrieu et al., 2010), which utilizes sequential Monte Carlo methods to construct efficient Metropolis–Hastings proposals.

The theory literature on pseudo-marginal algorithms has been active recently. To give just one example, Andrieu & Vihola (2016) give results that allow comparison of different implementations of the algorithm with respect to specific performance measures. This is of considerable interest, since pseudo-marginal algorithms have the right invariant measure, but typically have tuning parameters, such as the number of auxiliary variables sampled at each iteration, that can be difficult to choose efficiently.

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024

3.3. *Distributions with constrained support*

In many Bayesian models the prior distribution $\pi(x)$ has constrained support $\mathcal{C}$. Common examples include linear inequality constraints $Ax \leqslant c$ and restriction to a non-Euclidean manifold, such as a Stiefel or Grassmanian. These manifolds arise when incorporating orthogonality constraints. Constraints present a barrier to implementation of Markov chain Monte Carlo. Three solutions include: (i) ignore the constraint and simply reject proposals falling outside of $\mathcal{C}$; (ii) reparameterize to an unconstrained space before running the sampler; and (iii) implement Gibbs sampling with the conditional posterior distributions truncated to reflect the constraint. Unfortunately, (i) and (iii) can lead to slow mixing in many cases. Although (ii) is often more effective, it is usually not clear how to reparameterize; for recent developments for orthogonal matrices, refer to Jauch et al. (2019). There is a rich literature developing specialized algorithms for solving constrained sampling problems. This includes variants of Hamiltonian Monte Carlo for boundary constraints (Lan et al., 2014) and truncated multivariate Gaussian sampling (Pakman & Paninski, 2014). Duan et al. (2018) instead replace the exactly constrained posterior with an approximation, so that off-the-shelf Hamiltonian and other Markov transition kernels can be used directly. Patra & Dunson (2018) start by defining an unconstrained posterior, and then project posterior samples to satisfy the constraint.

4. Emerging areas

4.1. *Huge datasets*

Over the past decade there has been an explosion of interest in Markov chain algorithms that scale to large datasets. Broadly speaking, this work has two main components: subsampling algorithms that use only a subset of the data at each iteration, and divide-and-conquer algorithms that partition the data into multiple subsets on which computation is done independently and in parallel, followed by a recombining step to produce estimates of the quantities of interest.

Canonical examples of subsampling algorithms include stochastic gradient Langevin dynamics, which forgo the Hastings step and uses subsets of data to approximate $\nabla \log f(x)$ (Welling & Teh, 2011). Because each step of the algorithm requires much less computation, these approximate algorithms can outperform algorithms that use a Metropolis step with limited computational budgets. A nice study of the approximation error of such noisy unadjusted Langevin algorithms can be found in Dalalyan & Karagulyan (2017), while Chatterji et al. (2019) gives theoretical guarantees for unadjusted Langevin without requiring the usual smoothness conditions on the target. A strategy for reducing the variance of noisy gradient estimates is proposed in Dubey et al. (2016). Similarly, stochastic gradient Hamiltonian Monte Carlo (Chen et al., 2014) was initially conceived as an algorithm that skips the Hastings step in Hamiltonian Monte Carlo, and adds a friction term to ensure that the continuous-time dynamics using stochastic gradients preserves the target as its invariant measure. The resulting dynamics correspond to the underdamped Langevin diffusion popular in physics and applied mathematics (Pavliotis, 2014):

$$\begin{aligned}\dot{z} &= -\gamma z \,\mathrm{d}t - u\nabla U(x)\,\mathrm{d}t + \sqrt{2\gamma u}\dot{w}\\ \dot{x} &= z\,\mathrm{d}t,\end{aligned}$$

which is referred to as second-order Langevin in Chen et al. (2014). As exact solutions are intractable and discrete-time approximations are used, neither algorithm converges to the correct invariant measure unless step sizes are taken to zero as $t \to \infty$. Hence, both are examples of

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024

approximate algorithms. Nemeth & Fearnhead (2019) provides a recent review of Markov chain algorithms utilizing stochastic gradients.

Another approximate algorithm is the austerity approach of Korattikara et al. (2014), which uses subsets of data to approximate the Hastings acceptance probability. The idea is to replace the decision $\alpha(x, \tilde{x}) > u$ for $u \sim \text{Un}(0, 1)$ with a sequential test of the hypothesis $\alpha(x, \tilde{x}) > u$ of fixed Type I error rate. Additional data points are sampled and used in computing the Hastings ratio until the sequential test rejects the hypothesis $\alpha(x, \tilde{x}) \leqslant u$. More recent work in this direction has used control variates computed on all of the data in addition to subsamples to improve accuracy (Baker et al., 2019). The subsampling strategy can also be used to construct asymptotically exact algorithms, as in Firefly Monte Carlo (Maclaurin & Adams, 2015).

While these algorithms typically have lower costs per step than traditional Hastings algorithms, it is unknown whether subsampling generally harms mixing properties, which could potentially offset the advantage. Some have found that the size of subsamples needed to obtain good approximations tends to be quite large even in simple applications (Bardenet et al., 2017; Johndrow & Mattingly, 2018). Small approximation error may require the use of good control variates (Bouchard-Côté et al., 2018; Baker et al., 2019; Quiroz et al., 2019). This arguably shifts much of the work of algorithm design towards finding good control variates, which can be thought of as an optimization problem. Huggins et al. (2016) and Campbell & Broderick (2018, 2019) propose several optimization-based approaches to constructing control variates. These methods aim to find a weighted subsample of the data, which they refer to as a coreset, that gives a uniformly good approximation to the loglikelihood function.

4.2. *High-dimensional data*

Models with large numbers of parameters, for which the dimension p of the state space for the Markov chain is high, are another challenging class of applications and major focus of contemporary research. This is because Hastings algorithms have at least linear cost in dimension per step, and also mixing tends to degrade as dimension increases. The former cost can usually be assessed using known computational complexity results for the constituent algorithms used in propagating the Markov chain. There is also excellent literature on the scaling of mixing rates with dimension. For example, it has been shown under regularity conditions on the target that the mixing time of optimally tuned random walk Metropolis grows linearly in dimension (Gelman et al., 1997), for the Metropolis-adjusted Langevin algorithm the rate is $p^{1/3}$ (Pillai et al., 2012), and for Hamiltonian Monte Carlo it is $p^{1/4}$ (Beskos et al., 2013). Asymptotic rates only provide a rough guideline for practitioners, and for any particular problem in fixed dimension the implied ordering of the algorithms could be incorrect. Moreover, Hamiltonian Monte Carlo typically costs the most per step since it requires multiple gradient evaluations in addition to computation of the acceptance probability. However, ignoring this cost, Hamiltonian Monte Carlo is typically better than random walk and Langevin samplers in high-dimensional problems in our experience.

There is a rich recent literature focused specifically on high-dimensional linear models. Yang et al. (2016) shows that under certain conditions on the data, an add-delete-swap proposal for the classic Bayesian model-averaging problem has mixing times that grow linearly in p up to a logarithmic factor. A promising new algorithm (Zanella & Roberts, 2018) that combines importance tempering with Gibbs sampling shows remarkably good empirical performance for the same problem. Papaspiliopoulos et al. (2018) show that a blocking strategy in a challenging class of random effects models results in Gibbs sampling algorithms that mix in constant time in dimension. There has also been some success in designing efficient approximate algorithms.

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024

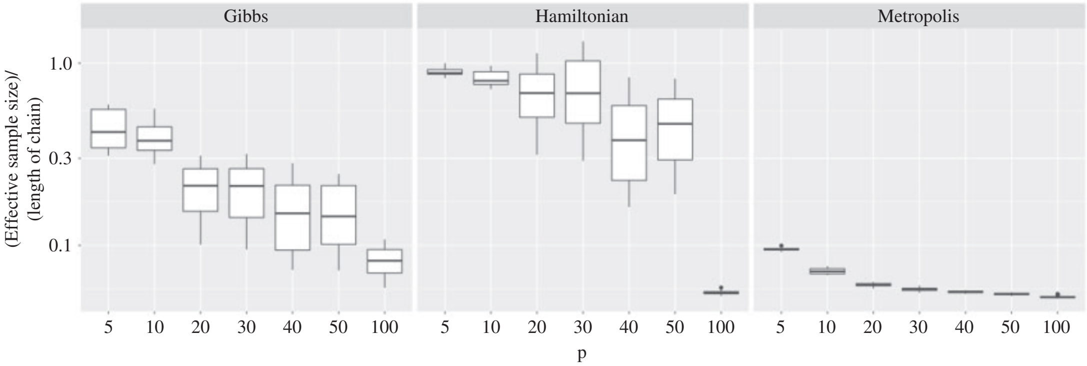

Fig. 2. Effective sample size divided by path length for alternative algorithms targeting posterior for the logistic regression model with a Gaussian prior in increasing dimension. The boxplot represents variation across the different parameters in the effective sample size divided by path length.

For example, Narisetty et al. (2019) proposes an approximate Gibbs sampler for the spike-and-slab prior in regression models, and Johndrow et al. (2018) suggests an approximate blocked Metropolis-within-Gibbs algorithm for the horseshoe shrinkage prior. Both algorithms are based on thresholding, and some theoretical results are given on accuracy.

Figure 2 shows the effective sample size divided by the number of retained samples for Hamiltonian Monte Carlo, Gibbs sampling with Polya-Gamma data augmentation (Polson et al., 2013) and Gaussian random walk Metropolis for a sequence of target distributions of increasing dimension. Random-walk Metropolis was tuned to give acceptance rates of about 0.23, and for Hamiltonian Monte Carlo we implement the no-U-turn sampler in `Stan`. The data are sampled from a logistic regression model with $n = 1000$ observations, and the regression coefficients and design matrix are sampled from independent standard normals. We use the prior $\beta \sim N(0, 25I_p)$. A burn-in of 1000 samples is discarded, and 9000 samples are used to compute effective sample sizes via the `mcmcse` package for `R` (R Development Core Team, 2020). As expected, all of the algorithms exhibit some degradation in performance as dimension increases. Hamiltonian Monte Carlo performs the best at every dimension except 100, while random walk Metropolis performs the worst. However, this fails to account for the much longer computation time required for Hamiltonian Monte Carlo.

The explanation for the relatively dismal performance of Hamiltonian Monte Carlo at dimension 100 in this example is interesting. In this case, `Stan` has warned us that many divergent transitions have occurred during the sampling phase, despite increasing the `adapt_delta` tuning parameter of the no-U-turn sampler to 0.99 from its default value of 0.95. Increasing this parameter is recommended when a large number of divergent transitions occurs. Higher values correspond to smaller step sizes in the leapfrog integrator, and therefore higher fidelity in simulation of Hamiltonian dynamics. We believe the difficulties of the no-U-turn sampler in this example originate from the fact that probabilities numerically 0 or 1 occur during maximum likelihood estimation for this model, and many of the maximum likelihood estimates are enormous. One typically expects that the regularization from the $N(0, 25I_p)$ prior will mitigate this problem for Bayesian computation, however in this case the geometry of the target appears to be too challenging for the no-U-turn sampler to perform well. Since this is not an uncommon problem in applied logistic regression, it may merit further investigation. Evidently, data augmentation Gibbs and random walk Metropolis also perform poorly when $p = 100$, but the drop in performance relative to $p = 50$ is not as dramatic.

To make the comparison transparent, we have thus far focused on the mixing properties of the different algorithms in increasing dimension without considering computation time. In Table 2

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024

Table 2. *Time in seconds to collect a path of length* 10 000 *for each of the three algorithms for different dimensions*

	Gibbs	Hamiltonian	Metropolis
5	4.48	114.39	0.56
10	4.16	135.12	0.54
20	5.27	210.72	0.67
30	6.54	245.89	0.70
40	8.81	374.98	0.83
50	11.45	874.55	0.89
100	34.03	3244.27	1.52

we show the time in seconds that it took to gather a path of length 10 000 using each of the three algorithms on a vintage 2019 MacBook Pro 2.4 GHz Intel core i9 CPU. With this additional information the comparison favours Gibbs and random walk Metropolis, due to the much greater per step cost of Hamiltonian Monte Carlo. We concede that other implementations of Hamiltonian Monte Carlo may be faster than the no-U-turn sampler implemented in `Stan`, and that we have made no effort to write different versions of our `Stan` model code to improve its speed. That said, this example provides a nice illustration of the trade-off between usability and optimal computation speed. Clearly, `Stan` is much more accessible to users than handwritten code. Although the `BayesLogit` package we use to implement Gibbs is also very user friendly, it is designed to estimate models with logistic links and Gaussian priors, whereas `Stan` is extremely flexible. On the other hand, a high computational price is paid for that flexibility. This trade-off is probably inevitable in trying to construct general-purpose probabilistic programming languages, and thus there is an ongoing need for skilled practitioners to achieve state-of-the-art performance for any particular problem.

4.3. *Parallel algorithms*

Strategies for parallelizing Markov chain algorithms include: (i) parallelizing computations within each step; (ii) asynchronous algorithms; (iii) divide-and-conquer algorithms; (iv) unbiased estimators via coupling; and (v) modularization.

Parallelization of computations, such as matrix multiplication and likelihood computations for mixture models, is now fairly routine for multicore architectures, and becoming more routine for graphical processing units. Suchard et al. (2010) advocated the use of graphical processing units in mixture modelling, and Lee et al. (2010) showed orders of magnitude improvements for sequential and population Monte Carlo applications. Despite the initial enthusiasm, graphical processing units have not come to occupy the central place in Markov chain computation that they do in machine learning, perhaps due to heavier communication costs.

A generic strategy for parallelization is to split the data needed to run the algorithm between a large number of computational workers. Unfortunately, in the case of Markov chain algorithms this typically requires large amounts of data, such as updated values of state variables, to be passed between all of the workers at every iteration, severely limiting the potential speedup. Asynchronous algorithms (Terenin et al., 2019a,b) address part of this limitation by not requiring workers to wait for updated states to arrive before continuing computation. The empirical results are promising for certain applications, but as yet there has been limited theoretical analysis of such algorithms.

An alternative is the divide-and-conquer approach, which usually runs posterior sampling algorithms independently for different subsets of the data and then recombines the samples. Because the algorithms run on the subsets are usually generic, the research activity in this area has

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024

focused on finding good recombination procedures. Scott et al. (2016) propose simply averaging samples from the posteriors of each subset of the data, which can be justified if each subset posterior is approximately Gaussian. An alternative is to use the data subsets to obtain noisy approximations to the full data posterior, in a manner related to stochastic gradient algorithms for optimization. One raises the subset likelihood to a power to match the information content in the full data. The full data posterior can then be approximated as an appropriate geometric centre of the subset-based approximations. Minsker et al. (2017) use a median, while Srivastava et al. (2018) instead employ a Wasserstein barycenter. Li et al. (2017) propose a simple implementation for calculating one-dimensional posterior summaries: sampling from each subset posterior in parallel, calculating quantiles for the posterior summaries of interest, and then averaging. This approach can be shown to provide an accurate approximation asymptotically.

An emerging literature aims to parallelize Markov chain computation in the classical way of running multiple chains, but improving the utility of this approach by using these multiple paths to construct unbiased estimators, eliminating the need to choose a burn-in period. Jacob et al. (2019) and Agapiou et al. (2018) show how to construct such estimators by simulating from couplings of the Markov chain. Both methods build upon the contributions of the 2013 PhD. thesis of C.-H. Rhee from Stanford University. This work has been extended to other algorithms, such as Hamiltonian Monte Carlo (Heng & Jacob, 2019) and pseudo-marginal algorithms (Middleton et al., 2019). This line of inquiry shows promise for solving the burn-in problem, though interesting and challenging issues remain. Perhaps chief among them is that the method requires the construction of couplings that achieve something close to the mixing time of the algorithm. While it is easy to design and sample from maximal one-step couplings for Hastings algorithms or maximal blockwise couplings for block Hastings algorithms, such couplings are not unique and may be very suboptimal over multiple steps, especially in high dimensions. Nonetheless, this is an exciting direction that we expect will see growing attention among practitioners.

Modularization applies a very different strategy for parallelization, considering different components of a Bayesian model separately. This can have dual advantages of improving robustness to model misspecification and improving computational efficiency (Liu et al., 2009; Jacob et al., 2017). Chen & Dunson (2017) used modularization in a genomics application involving a gene expression response and 38 million single nucleotide polymorphism predictors. They first defined a mixture model to flexibly characterize the marginal density of the response, running Markov chain Monte Carlo for this mixture model without considering the predictors. They then used the output from this chain in defining Bayesian tests for marginal association between each predictor and the response, inducing dependence in these screening tests. Peruzzi & Dunson (2018) instead considered modularization in the context of multiscale modelling, in which fully Bayesian modelling can encounter identifiability and computational efficiency problems. Modular Bayes methodology is still in its infancy, and there is a need for better theoretical justification for approaches that treat posterior distributions for different model components separately without imposing restrictions of a coherent joint probability model.

4.4. *Nonreversible Markov chains*

The past few years have seen an explosion of interest in nonreversible Markov chain algorithms, motivated by the fact that expanding the set of Markov chain algorithms under consideration beyond reversible ones, typical of all the common Markov chain Monte Carlo algorithms described above, conveys the potential for substantial gains in computational efficiency. One class of nonreversible algorithms of particular interest in recent years are piecewise deterministic Markov processes. In the statistical literature, the best studied of these are the zig-zag sampler (Bierkens et al., 2019a) and the bouncy particle sampler (Bouchard-Côté et al., 2018), which

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024

were originally characterized in the physics literature (Turitsyn et al., 2011; Peters & de With, 2012). In these processes particles move deterministically, though not necessarily linearly, and change direction at random time intervals. Unlike the Hastings algorithm there are no rejections, but the particle may move only for a short time between changing directions when it resides in low-probability regions of the target.

A major advantage of these algorithms is that it is possible to construct exact samplers without using all of the data at each iteration (Bierkens et al., 2019a). In particular, the stationary distribution can be maintained even when using arbitrarily small subsamples. Sen et al. (2019) recently developed efficient subsampling algorithms for piecewise deterministic Markov processes, improving on uniform subsampling. Recent work shows that the bouncy particle and zig-zag samplers achieve exponential convergence rates for a wider variety of targets than most alternatives (Bierkens et al., 2019b; Deligiannidis et al., 2019).

We expect continued interest in nonreversible algorithms, and hope to see more progress in using these algorithms for challenging applied problems in the near future. These algorithms are in the very early stages of their development, and several practical hurdles must still be overcome before they can be used routinely. One of the major issues is that implementations require construction of computational upper bounds; useful bounds are currently nontrivial to obtain outside of certain restrictive model classes, such as for logistic regression. It also remains to be seen whether piecewise deterministic Markov processes can be defined to have clearly improved practical performance relative to state-of-the-art reversible Markov chain Monte Carlo competitors in high-dimensional and challenging applications. So far most implementations of piecewise deterministic Markov processes have focused on low-dimensional cases.

5. Open areas and ongoing challenges

5.1. *Moving from an art to a science*

The design of effective sampling algorithms has largely been more of an art than a science. Effective Bayesian practitioners build up a bag of tricks over the years, which they employ to tweak and refine algorithms when initial off-the-shelf approaches fail to work adequately in a given application. Such failures remain common, presenting a barrier to routine implementation of Bayesian methods in many application areas and to entry into the field. Although probabilistic programming languages, such as `Stan`, have taken large strides in automating Bayesian posterior sampling, there are still many cases in which these automated approaches simply do not work well. These scenarios often need to be studied on a case-by-case basis; without clear theoretical understanding of why standard algorithms fail, statisticians are effectively operating in the dark based on intuition and experience. There continue to be regular discoveries of problematic special cases. An example is Johndrow et al. (2019), which showed that data augmentation algorithms fail badly for imbalanced binary data. It is our hope that in the coming years major steps will be taken to automate this algorithm design process, fundamentally improving understanding of when and why current algorithms fail, how to automatically design better algorithms to address these failures, and how to develop more robust algorithms.

5.2. *Automating scalability*

One of the particularly challenging cases is high-dimensional data and parameter models. As mentioned in § 3.5, the state of the art is currently focused on developing specialized algorithms for very specific models and priors, for example the algorithms of Johndrow et al. (2018) and Hahn et al. (2019) for high-dimensional linear regression with horseshoe shrinkage priors. There is a lack of general-purpose algorithms for scaling up computation in broad classes of models,

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024

and even relatively few metastrategies that are available. Hence, it is widely thought that posterior sampling algorithms are simply not scalable, despite rather convincing examples where sampling can be much faster than optimization, such as in mixture models (Ma et al., 2019b). This leads to use of ad hoc variational approximations, which lack any characterization of approximation error, and indeed can be extremely inaccurate in many cases. Hamiltonian Monte Carlo, and related algorithms, have some ability to scale up to moderately high dimensions in certain classes of models, but do not naturally lend themselves to cases where there is special structure in the target, such as sparsity or more generally lower-dimensional structure. Such structure is very common in high-dimensional statistical models, and seems to lend itself better to blocked Gibbs sampling or Hastings algorithms with proposals that can more easily jump between configurations than gradient-based methods. However, scaling to truly high-dimensional cases would seem to necessarily require some use of divide-and-conquer and parallelization. In the large sample size case a number of such algorithms have been developed, as noted in § 3.4. The high-dimensional case is fundamentally different, and we predict a major area of future development will be focused on developing truly parallelizable versions of the Hastings algorithm, beyond the current use of graphical processing units and related tricks as in Jordan et al. (2019), Terenin et al. (2019a) and Vono et al. (2019). Any such parallelizable Hastings algorithm would need to carefully consider scalability in terms of not only cost per iteration, but also mixing times.

5.3. *Learning from optimization*

Many of the above problems relate to the need to develop general use algorithms that are well understood, scalable and can be used in most settings. The optimization literature is much closer to this ideal than the posterior sampling literature, due in part to the larger research community focused on optimization and to the focus on convex problems. Unfortunately, standard optimization algorithms are not aimed at approximating expectations, and hence are of limited use in Bayesian inference. Such approaches instead produce point estimates of parameters, typically without uncertainty quantification. One interesting emerging area is to view posterior sampling as an optimization problem on the space of probability measures, as in Wibisono (2018) and Ma et al. (2019a). One can then potentially exploit powerful tools developed in optimization to better design efficient algorithms. Thus far, this literature has focused on unadjusted overdamped and underdamped Langevin algorithms.

5.4. *Generalized Bayes*

One promising direction in addressing some of the above challenges is to consider generalized Bayes procedures, which involve some modification of the usual Bayesian paradigm such as replacing the likelihood function with a pseudo, modular or composite likelihood (Chernozhukov & Hong, 2003; Dunson & Taylor, 2005; Jiang & Tanner, 2008; Yang & He, 2012). Such generalized Bayes procedures can be designed to automatically improve scalability and robustness to model misspecification (Bissiri et al., 2016; Miller & Dunson, 2018) and data contamination, which are critical problems in large data applications. Considerable work remains to be done in this area to maintain the appealing interpretation of the Bayesian framework, while optimally balancing other considerations ranging from robustness to computational complexity.

Acknowledgement

This work was partially supported by the United States National Science Foundation.

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024

References

Agapiou, S., Roberts, G. O. & Vollmer, S. J. (2018). Unbiased Monte Carlo: Posterior estimation for intractable/infinite-dimensional models. *Bernoulli* **24**, 1726–86.

Albert, J. H. & Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *J. Am. Statist. Assoc.* **88**, 669–79.

Amit, Y. & Grenander, U. (1991). Comparing sweep strategies for stochastic relaxation. *J. Mult. Anal.* **37**, 197–222.

Andrieu, C., Doucet, A. & Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *J. R. Statist. Soc.* B **72**, 269–342.

Andrieu, C. & Roberts, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *Ann. Statist.* **37**, 697–725.

Andrieu, C. & Vihola, M. (2016). Establishing some order amongst exact approximations of MCMCs. *Ann. Appl. Prob.* **26**, 2661–96.

Atchadé, Y. F. & Rosenthal, J. S. (2005). On adaptive Markov chain Monte Carlo algorithms. *Bernoulli* **11**, 815–28.

Attias, H. (1999). Inferring parameters and structure of latent variable models by variational Bayes. In *Proc. 15th Conf. on Uncertainty in Artificial Intelligence*. Burlington, MA: Morgan Kaufmann Publishers Inc.

Baker, J., Fearnhead, P., Fox, E. B. & Nemeth, C. (2019). Control variates for stochastic gradient MCMC. *Statist. Comp.* **29**, 599–615.

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

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *J. R. Statist. Soc.* B **36**, 192–225.

Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J.-M. & Stuart, A. (2013). Optimal tuning of the hybrid Monte Carlo algorithm. *Bernoulli* **19**, 1501–34.

Betancourt, M. & Girolami, M. (2015). Hamiltonian Monte Carlo for hierarchical models. In *Current Trends in Bayesian Methodology with Applications*, S. K. Upadhyay, U. Singh, D. K. Dey and A. Loganathan, eds. pp. 79–102, Boca Raton, FL: CRC Press.

Bierkens, J., Fearnhead, P. & Roberts, G. O. (2019a). The zig-zag process and super-efficient sampling for Bayesian analysis of big data. *Ann. Statist.* **47**, 1288–320.

Bierkens, J., Roberts, G. O. & Zitt, P.-A. (2019b). Ergodicity of the zigzag process. *Ann. Appl. Prob.* **29**, 2266–301.

Bissiri, P., Holmes, C. & Walker, S. (2016). A general framework for updating belief distributions. *J. R. Statist. Soc.* B **78**, 1103–30.

Biswas, N., Jacob, P. E. & Vanetti, P. (2019). Estimating convergence of Markov chains with L-lag couplings. *arXiv:*1905.09971v3.

Blei, D. M., Kucukelbir, A. & McAuliffe, J. D. (2018). Variational inference: A review for statisticians. *arXiv:*1601.00670v9.

Bou-Rabee, N. & Sanz-Serna, J. M. (2017). Randomized Hamiltonian Monte Carlo. *Ann. Appl. Prob.* **27**, 2159–94.

Bouchard-Côté, A., Vollmer, S. J. & Doucet, A. (2018). The bouncy particle sampler: A nonreversible rejection-free Markov chain Monte Carlo method. *J. Am. Statist. Assoc.* **113**, 855–67.

Brooks, S., Gelman, A., Jones, G. & Meng, X.-L. (2011). *Handbook of Markov chain Monte Carlo*. Boca Raton, FL: CRC Press.

Brooks, S. P. & Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. *J. Comp. Graph. Statist.* **7**, 434–55.

Campbell, T. & Broderick, T. (2018). Bayesian coreset construction via greedy iterative geodesic ascent. *arXiv:*1802.01737v2.

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

Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P. & Riddell, A. (2017). Stan: A probabilistic programming language. *J. Statist. Software* **76**, 1–32.

Casella, G. & George, E. (1992). Explaining the Gibbs sampler. *Am. Statistician* **46**, 167–74.

Chatterji, N. S., Diakonikolas, J., Jordan, M. I. & Bartlett, P. L. (2019). Langevin Monte Carlo without smoothness. *arXiv:*1905.13285.

Chen, T., Fox, E. & Guestrin, C. (2014). Stochastic gradient Hamiltonian Monte Carlo. In *Proc. 31st Int. Conf. on Machine Learning*, vol. 32, pp. 1683–91.

Chen, Y. & Dunson, D. B. (2017). Modular Bayes screening for high-dimensional predictors. *arXiv:*1703.09906.

Chernozhukov, V. & Hong, H. (2003). An MCMC approach to classical estimation. *J. Economet.* **115**, 293–346.

Chib, S. & Greenberg, E. (1995). Understanding the Metropolis–Hastings algorithm. *Am. Statistician* **49**, 327–35.

Dalalyan, A. S. & Karagulyan, A. (2017). User-friendly guarantees for the Langevin Monte Carlo with inaccurate gradient. *Stoch. Proces. Appli.* **129**, 5278–311.

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024

de Valpine, P., Turek, D., Paciorek, C. J., Anderson-Bergman, C., Lang, D. T. & Bodik, R. (2017). Programming with models: Writing statistical algorithms for general model structures with Nimble. *J. Comp. Graph. Statist.* **26**, 403–13.

Del Moral, P., Doucet, A. & Jasra, A. (2006). Sequential Monte Carlo samplers. *J. R. Statist. Soc.* B **68**, 411–36.

Deligiannidis, G., Bouchard-Côté, A. & Doucet, A. (2019). Exponential ergodicity of the bouncy particle sampler. *Ann. Statist.* **47**, 1268–87.

Devroye, L. (1986). Nonuniform random variate generation. *Hand. Oper. Res. Manag. Sci.* **13**, 83–121.

Diaconis, P. & Saloff-Coste, L. (1998). What do we know about the Metropolis algorithm? *J. Comp. Syst. Sci.* **57**, 20–36.

Dongarra, J. & Sullivan, F. (2000). Guest editors' introduction: The top 10 algorithms. *Comp. Sci. Eng.* **2**, 22.

Douc, R., Fort, G., Moulines, E. & Soulier, P. (2004). Practical drift conditions for subgeometric rates of convergence. *Ann. Appl. Prob.* **14**, 1353–77.

Duan, L. L., Young, A. L., Nishimura, A. & Dunson, D. B. (2018). Bayesian constraint relaxation. *Biometrika*, to appear.

Duane, S., Kennedy, A. D., Pendleton, B. & Roweth, D. (1987). Hybrid Monte Carlo. *Phys. Lett.* **195**, 216–22.

Dubey, K. A., Reddi, S. J., Williamson, S. A., Poczos, B., Smola, A. J. & Xing, E. P. (2016). Variance reduction in stochastic gradient Langevin dynamics. *Adv. Neur. Info. Proces. Syst.* **29**, 1154–62.

Dunson, D. & Taylor, J. A. (2005). Approximate Bayesian inference for quantiles. *J. Nonparam. Statist.* **17**, 385–400.

Durmus, A., Moulines, E. & Saksman, E. (2019). On the convergence of Hamiltonian Monte Carlo. *arXiv:*1705.00166v2.

Flegal, J. M., Haran, M. & Jones, G. L. (2008). Markov chain Monte Carlo: Can we trust the third significant figure? *Statist. Sci.* **23**, 250–60.

Flegal, J. M. & Jones, G. L. (2010). Batch means and spectral variance estimators in Markov chain Monte Carlo. *Ann. Statist.* **38**, 1034–70.

Fort, G. & Moulines, E. (2003). Polynomial ergodicity of Markov transition kernels. *Stoch. Proces. Appl.* **103**, 57–99.

Frühwirth-Schnatter, S. & Frühwirth, R. (2010). Data augmentation and MCMC for binary and multinomial logit models. In *Statistical Modelling and Regression Structures*, T. Kneib and G. Tutz, eds. pp. 111–32. New York: Springer.

Gamerman, D. & Lopes, H. F. (2006). *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. London: Chapman and Hall/CRC.

Gelfand, A. & Smith, A. (1990). Sampling-based approaches to calculating marginal densities. *J. Am. Statist. Assoc.* **85**, 398–409.

Gelman, A., Gilks, W. & Roberts, G. O. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *Ann. Appl. Prob.* **7**, 110–20.

Gelman, A. & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statist. Sci.* **7**, 457–72.

Geman, S. & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *Trans. Patt. Anal. Mach. Intel.* **6**, 721–41.

Geyer, C. (1991). Markov chain Monte Carlo maximum likelihood, computing science and statistics. In *Proc. 23rd Symp. Interface*.

Geyer, C. J. & Thompson, E. A. (1995). Annealing Markov chain Monte Carlo with applications to ancestral inference. *J. Am. Statist. Assoc.* **90**, 909–20.

Gilks, W. & Wild, P. (1992). Adaptive rejection sampling for Gibbs sampling. *Appl. Statist.* **41**, 337–48.

Gilks, W. R., Best, N. G. & Tan, K. (1995). Adaptive rejection Metropolis sampling within Gibbs sampling. *Appl. Statist.* **44**, 455–72.

Girolami, M. & Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *J. R. Statist. Soc.* B **73**, 123–214.

Gramacy, R., Samworth, R. & King, R. (2010). Importance tempering. *Statist. Comp.* **20**, 1–7.

Green, P. (1995). Reversible-jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* **82**, 711–32.

Green, P. J. & Richardson, S. (2001). Modelling heterogeneity with and without the Dirichlet process. *Scand. J. Statist.* **28**, 355–75.

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

Hahn, P. R., He, J. & Lopes, H. F. (2019). Efficient sampling for Gaussian linear regression with arbitrary priors. *J. Comp. Graph. Statist.* **28**, 142–54.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**, 97–109.

Heng, J. & Jacob, P. E. (2019). Unbiased Hamiltonian Monte Carlo with couplings. *Biometrika* **106**, 287–302.

Hitchcock, D. B. (2003). A history of the Metropolis–Hastings algorithm. *Am. Statistician* **57**, 254–7.

Hoffman, M. D. & Gelman, A. (2014). The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.* **15**, 1593–623.

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024

Holmes, C. C. & Held, L. (2006). Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian Anal.* **1**, 145–68.

Huggins, J., Campbell, T. & Broderick, T. (2016). Coresets for scalable Bayesian logistic regression. In *Proc. Advances in Neural Information Processing Systems*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon and R. Garnett. eds.

Hukushima, K. & Nemoto, K. (1996). Exchange Monte Carlo method and application to spin glass simulations. *J. Phys. Soc. Japan* **65**, 1604–8.

Jacob, P. E., Murray, L. M., Holmes, C. C. & Robert, C. P. (2017). Better together? Statistical learning in models made of modules. *arXiv:*1708.08719.

Jacob, P. E., O'Leary, J. & Atchadé, Y. F. (2019). Unbiased Markov chain Monte Carlo with couplings. *arXiv:*1708.03625v5.

Jain, S. & Neal, R. M. (2004). A split–merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *J. Comp. Graph. Statist.* **13**, 158–82.

Jarner, S. F. & Tweedie, R. L. (2003). Necessary conditions for geometric and polynomial ergodicity of random-walk-type. *Bernoulli* **9**, 559–78.

Jauch, M., Hoff, P. D. & Dunson, D. B. (2019). Monte Carlo simulation on the Stiefel manifold via polar expansion. *arXiv:*1906.07684.

Jiang, W. & Tanner, M. (2008). Gibbs posterior for variable selection in high-dimensional classification and data mining. *Ann. Statist.* **36**, 2207–31.

Johndrow, J. E. & Mattingly, J. C. (2018). Error bounds for approximations of Markov chains used in Bayesian sampling. *arXiv:*1711.05382v2.

Johndrow, J. E., Orenstein, P. & Bhattacharya, A. (2018). Bayes shrinkage at GWAS scale: Convergence and approximation theory of a scalable MCMC algorithm for the horseshoe prior. *arXiv:*1705.00841v3.

Johndrow, J. E., Smith, A., Pillai, N. & Dunson, D. B. (2019). MCMC for imbalanced categorical data. *J. Am. Statist. Assoc.* **114**, 1394–403.

Johnson, L. T. & Geyer, C. J. (2012). Variable transformation to obtain geometric ergodicity in the random-walk Metropolis algorithm. *Ann. Statist.* **40**, 3050–76.

Johnson, V. E. (1996). Studying convergence of Markov chain Monte Carlo algorithms using coupled sample paths. *J. Am. Statist. Assoc.* **91**, 154–66.

Johnson, V. E. (1998). A coupling-regeneration scheme for diagnosing convergence in Markov chain Monte Carlo algorithms. *J. Am. Statist. Assoc.* **93**, 238–48.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S. & Saul, L. K. (1999). An introduction to variational methods for graphical models. *Mach. Learn.* **37**, 183–233.

Jordan, M. I., Lee, J. D. & Yang, Y. (2019). Communication-efficient distributed statistical inference. *J. Am. Statist. Assoc.* **114**, 668–81.

Khasminskii, R. (1980). *Stochastic Stability of Differential Equations*. New York: Springer.

Korattikara, A., Chen, Y. & Welling, M. (2014). Austerity in MCMC land: Cutting the Metropolis–Hastings budget. In *Proc. Int. Conf. on Machine Learning*.

Kou, S., Zhou, Q. & Wong, W. H. (2006). Equi-energy sampler with applications in statistical inference and statistical mechanics. *Ann. Statist.* **34**, 1581–619.

Lan, S., Zhou, B. & Shahbaba, B. (2014). Spherical Hamiltonian Monte Carlo for constrained target distributions. In *Proc. JMLR Workshop and Conf.*, vol. 32.

Lee, A., Yau, C., Giles, M. B., Doucet, A. & Holmes, C. C. (2010). On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods. *J. Comp. Graph. Statist.* **19**, 769–89.

Li, C., Srivastava, S. & Dunson, D. B. (2017). Simple, scalable and accurate posterior interval estimation. *Biometrika* **104**, 665–80.

Liu, F., Bayarri, M. & Berger, J. (2009). Modularization in Bayesian analysis, with emphasis on analysis of computer models. *Bayesian Anal.* **4**, 119–50.

Liu, J. S. (2008). *Monte Carlo Strategies in Scientific Computing*. New York: Springer Science & Business Media.

Livingstone, S., Betancourt, M., Byrne, S. & Girolami, M. (2016). On the geometric ergodicity of Hamiltonian Monte Carlo. *Bernouilli*, **25**, 3109–38.

Livingstone, S., Faulkner, M. F. & Roberts, G. O. (2019). Kinetic energy choice in Hamiltonian/hybrid Monte Carlo. *Biometrika* **106**, 303–19.

Lu, Y., Lu, J. & Nolen, J. (2019). Accelerating Langevin sampling with birth–death. *arXiv:*1905.09863.

Ma, Y.-A., Chatterji, N., Cheng, X., Flammarion, N., Bartlett, P. & Jordan, M. I. (2019a). Is there an analog of Nesterov acceleration for MCMC? *arXiv:*1902.00996v2.

Ma, Y.-A., Chen, Y., Jin, C., Flammarion, N. & Jordan, M. I. (2019b). Sampling can be faster than optimization. *arXiv:*1811.08413v2.

Maclaurin, D. & Adams, R. P. (2015). Firefly Monte Carlo: Exact MCMC with subsets of data. In *Proc. 24th Int. Joint Conf. on Artificial Intelligence*.

Mangoubi, O., Pillai, N. S. & Smith, A. (2018). Does Hamiltonian Monte Carlo mix faster than a random walk on multimodal densities? *arXiv:*1808.03230v2.

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024

Marinari, E. & Parisi, G. (1992). Simulated tempering: A new Monte Carlo scheme. *Europhys. Lett.* **19**, 451.

Mengersen, K. L. & Tweedie, R. L. (1996). Rates of convergence of the Hastings and Metropolis algorithms. *Ann. Statist.* **24**, 101–21.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. & Teller, E. (1953). Equations of state calculations by fast computing machines. *J. Chem. Phys.* **21**, 1087–92.

Meyn, S. P. & Tweedie, R. L. (1993). *Markov Chains and Stochastic Stability*. New York: Springer.

Middleton, L., Deligiannidis, G., Doucet, A. & Jacob, P. E. (2019). Unbiased Markov chain Monte Carlo for intractable target distributions. *arXiv:*1807.08691v2.

Miller, J. & Dunson, D. (2018). Robust Bayesian inference via coarsening. *J. Am. Statist. Assoc.* **114**, 1113–25.

Minsker, S., Srivastava, S., Lin, L. & Dunson, D. B. (2017). Robust and scalable Bayes via a median of subset posterior measures. *J. Mach. Learn. Res.* **18**, 4488–527.

Møller, J., Pettitt, A. N., Reeves, R. & Berthelsen, K. K. (2006). An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants. *Biometrika* **93**, 451–8.

Murray, I., Adams, R. P. & MacKay, D. J. (2010). Elliptical slice sampling. *J. Mach. Learn. Res.* **9**, 541–8.

Murray, I., Ghahramani, Z. & MacKay, D. (2006). MCMC for doubly-intractable distributions. In *Proc. 22nd Conf. on Uncertainty in Artificial Intelligence*.

Narisetty, N. N., Shen, J. & He, X. (2019). Skinny Gibbs: A consistent and scalable Gibbs sampler for model selection. *J. Am. Statist. Assoc.* **114**, 1205–17.

Neal, R. (2013). MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo*, S. Brooks, A. Gelman, G. Jones and X.-L. Meng, eds. pp. 113–62.

Neal, R. M. (2003). Slice sampling. *Ann. Statist.* **31**, 705–67.

Nemeth, C. & Fearnhead, P. (2019). Stochastic gradient Markov chain Monte Carlo. *arXiv:*1907.06986.

Nishimura, A., Dunson, D. & Lu, J. (2017). Discontinuous Hamiltonian Monte Carlo for discrete parameters and discontinuous likelihoods. *Biometrika*, to appear.

Pakman, A. & Paninski, L. (2014). Exact Hamiltonian Monte Carlo for truncated multivariate Gaussians. *J. Comp. Graph. Statist.* **23**, 518–42.

Papaspiliopoulos, O., Roberts, G. O. & Zanella, G. (2018). Scalable inference for crossed random effects models. *Biometrika*, to appear.

Patra, S. & Dunson, D. B. (2018). Constrained Bayesian inference through posterior projections. *arXiv:*1812.05741.

Pavliotis, G. A. (2014). *Stochastic Processes and Applications: Diffusion Processes, the Fokker–Planck and Langevin Equations*. New York: Springer.

Peruzzi, M. & Dunson, D. B. (2018). Bayesian modular and multiscale regression. *arXiv:*1809.05935.

Peters, E. A. & de With, G. (2012). Rejection-free Monte Carlo sampling for general potentials. *Phys. Rev. E* **85**, 026703.

Pillai, N. S., Stuart, A. M. & Thiéry, A. H. (2012). Optimal scaling and diffusion limits for the Langevin algorithm in high dimensions. *Ann. Appl. Prob.* **22**, 2320–56.

Polson, N. G., Scott, J. G. & Windle, J. (2013). Bayesian inference for logistic models using Pólya–Gamma latent variables. *J. Am. Statist. Assoc.* **108**, 1339–49.

Quiroz, M., Kohn, R., Villani, M. & Tran, M.-N. (2019). Speeding up MCMC by efficient data subsampling. *J. Am. Statist. Assoc.* **114**, 831–43.

R Development Core Team (2020). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0, http://www.R-project.org.

Rao, V., Lin, L. & Dunson, D. B. (2016). Data augmentation for models based on rejection sampling. *Biometrika* **103**, 319–35.

Robert, C. & Casella, G. (2009). *Introducing Monte Carlo Methods*. New York: Springer.

Robert, C. & Casella, G. (2010). A history of Markov chain Monte Carlo - subjective recollections from incomplete data. In *Handbook of Markov Chain Monte Carlo*, S. Brooks, A. Gelman, G. Jones and X.-L. Meng, eds. pp. 49–66.

Robert, C. & Casella, G. (2013). *Monte Carlo Statistical Methods*. New York: Springer.

Roberts, G. O. & Rosenthal, J. S. (1999). Convergence of slice sampler Markov chains. *J. R. Statist. Soc.* B **61**, 643–60.

Roberts, G. O. & Rosenthal, J. S. (2007). Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms. *J. Appl. Prob.* **44**, 458–75.

Roberts, G. O. & Sahu, S. K. (1997). Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler. *J. R. Statist. Soc.* B **59**, 291–317.

Roberts, G. O. & Tweedie, R. L. (1996). Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli* **2**, 341–63.

Rosenthal, J. S. (1995). Minorization conditions and convergence rates for Markov chain Monte Carlo. *J. Am. Statist. Assoc.* **90**, 558–66.

Salvatier, J., Wiecki, T. V. & Fonnesbeck, C. (2016). Probabilistic programming in Python using PyMC3. *PeerJ Comp. Sci.* **2**, e55.

Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I. & McCulloch, R. E. (2016). Bayes and big data: The consensus Monte Carlo algorithm. *Int. J. Manag. Sci. Eng. Manag.* **11**, 78–88.

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024

Sen, D., Sachs, M., Lu, J. & Dunson, D. (2019). Efficient posterior sampling for high-dimensional imbalanced logistic regression. *arXiv:*1905.11232v2.

Srivastava, S., Li, C. & Dunson, D. B. (2018). Scalable Bayes via barycenter in Wasserstein space. *J. Mach. Learn. Res.* **19**, 312–46.

Stoehr, J., Benson, A. & Friel, N. (2019). Noisy Hamiltonian Monte Carlo for doubly intractable distributions. *J. Comp. Graph. Statist.* **28**, 220–32.

Suchard, M. A., Wang, Q., Chan, C., Frelinger, J., Cron, A. & West, M. (2010). Understanding GPU programming for statistical computation: Studies in massively parallel massive mixtures. *J. Comp. Graph. Statist.* **19**, 419–38.

Swendsen, R. H. & Wang, J.-S. (1987). Nonuniversal critical dynamics in Monte Carlo simulations. *Phys. Rev. Lett.* **58**, 86.

Tanner, M. & Wong, W. (1987). The calculation of posterior distributions by data augmentation. *J. Am. Statist. Assoc.* **82**, 528–50.

Tawn, N. G., Roberts, G. O. & Rosenthal, J. S. (2019). Weight-preserving simulated tempering. *Statist. Comp.*, doi 10.1007/s11222-019-09863-3.

Terenin, A., Dong, S. & Draper, D. (2019a). GPU-accelerated Gibbs sampling: A case study of the horseshoe probit model. *Statist. Comp.* **29**, 301–10.

Terenin, A., Simpson, D. & Draper, D. (2019b). Asynchronous Gibbs sampling. *arXiv:*1509.08999v6.

Tierney, L. (1991). Exploring posterior distributions using Markov chains. In *Computing Science and Statistics: Proc. 23rd Symp. on the Interface*, E. Keramidas. ed.

Tierney, L. (1994). Markov chains for exploring posterior distributions (with discussion). *Ann. Statist.* **22**, 1701–86.

Turitsyn, K. S., Chertkov, M. & Vucelja, M. (2011). Irreversible Monte Carlo algorithms for efficient sampling. *Physica D* **240**, 410–14.

Vono, M., Paulin, D. & Doucet, A. (2019). Efficient MCMC sampling with dimension-free convergence rate using ADMM-type splitting. *arXiv:*1905.11937v2.

Wainwright, M. J. & Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Foundat. Trends Mach. Learn.* **1**, 1–305.

Welling, M. & Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proc. 28th Int. Conf. on Machine Learning (ICML-11)*.

Wibisono, A. (2018). Sampling as optimization in the space of measures: The Langevin dynamics as a composite optimization problem. In *Proc. Conf. on Learning Theory*.

Woodard, D., Schmidler, S. C. & Huber, M. (2009a). Sufficient conditions for torpid mixing of parallel and simulated tempering. *Electron. J. Prob.* **14**, 780–804.

Woodard, D. B., Schmidler, S. C. & Huber, M. (2009b). Conditions for rapid mixing of parallel and simulated tempering on multimodal distributions. *Ann. Appl. Prob.* **19**, 617–40.

Yang, Y. & He, X. (2012). Bayesian empirical likelihood for quantile estimation. *Ann. Statist.* **40**, 1102–31.

Yang, Y., Wainwright, M. J. & Jordan, M. I. (2016). On the computational complexity of high-dimensional Bayesian variable selection. *Ann. Statist.* **44**, 2497–532.

Zanella, G. & Roberts, G. (2018). Scalable importance tempering and Bayesian variable selection. *J. R. Statist. Soc.* B **81**, 489–517.

[*Received on* 26 *June* 2019. *Editorial decision on* 23 *September* 2019]

Downloaded from https://academic.oup.com/biomet/article/107/1/1/5686745 by NTNU Library user on 19 January 2024