Received: 12 November 2017 | Revised: 8 April 2018 | Accepted: 9 April 2018

DOI: 10.1002/wics.1435

WILEY WIREs COMPUTATIONAL STATISTICS

OVERVIEW

Accelerating MCMC algorithms

Christian P. Robert[1,2] | Víctor Elvira[3,4] | Nick Tawn[2] | Changye Wu[1]

[1]Université Paris Dauphine, PSL Research University, Paris, France

[2]Department of Statistics, University of Warwick, Coventry, UK

[3]IMT Lille Douai, Douai, France

[4]CRIStAL, Lille, France

Correspondence
Christian P. Robert, CEREMADE, Université Paris-Dauphine, 75775 Paris cedex 16, France.
Email: christian.robert@ceremade.dauphine.fr

Funding information
EPSRC grant, Grant/Award Number: EP/K014463/1; REA grant, Grant/Award Number: PCOFUND-GA-2013-609102; Marie Curie Fellowship, Grant/Award Number: FP7/2007-2013; Fulbright program; *Agence Nationale de la Recherche* of France under PISCES project, Grant/Award Number: ANR-17-CE40-0031-01; Chinese Government (CSC); Institut Universitaire de France, Grant/Award Number: 2016–2021

Markov chain Monte Carlo algorithms are used to simulate from complex statistical distributions by way of a local exploration of these distributions. This local feature avoids heavy requests on understanding the nature of the target, but it also potentially induces a lengthy exploration of this target, with a requirement on the number of simulations that grows with the dimension of the problem and with the complexity of the data behind it. Several techniques are available toward accelerating the convergence of these Monte Carlo algorithms, either at the exploration level (as in tempering, Hamiltonian Monte Carlo and partly deterministic methods) or at the exploitation level (with Rao–Blackwellization and scalable methods).

This article is categorized under:
Statistical and Graphical Methods of Data Analysis > Markov Chain Monte Carlo (MCMC)
Algorithms and Computational Methods > Algorithms
Statistical and Graphical Methods of Data Analysis > Monte Carlo Methods

KEYWORDS

Bayesian analysis, computational statistics, convergence of algorithms, efficiency of algorithms, Hamiltonian Monte Carlo, Monte Carlo methods, Rao-Blackwellisation, simulation, tempering

1 | INTRODUCTION

Markov chain Monte Carlo (MCMC) algorithms have been used for nearly 60 years and have become a reference method for analyzing Bayesian complex models in the early 1990s (Gelfand & Smith, 1990). The strength of this method is that it guarantees convergence to the quantity (or quantities) of interest with minimal requirements on the targeted distribution (also called *target*) behind such quantities. In that sense, MCMC algorithms are robust or universal, as opposed to the most standard Monte Carlo methods (e.g., Rubinstein, 1981; Robert & Casella, 2004) that require direct simulations from the target distribution. This robustness may, however, induce a slow convergence behavior in that the exploration of the relevant space—meaning the part of the space supporting the distribution that has a significant probability mass under that distribution—may take a long while, as the simulation usually proceeds by local jumps in the vicinity of the current position. In other words, MCMC–especially in its off-the-shelf versions like Gibbs sampling and Metropolis–Hastings (MH) algorithms—is very often myopic in that it provides a good illumination of a local area, while remaining unaware of the global support of the distribution. As with most other simulation methods, there always exist ways of creating highly convergent MCMC algorithms by taking further advantage of the structure of the target distribution. Here, we mostly limit ourselves to the realistic situation where the target density is only known as the output of a computer code or to a setting similarly limited in its information content.

The approaches to the acceleration of MCMC algorithms can be divided in several categories, from those which improve our knowledge about the target distribution, to those that modify the proposal in the algorithm, including those that exploit better the outcome of the original MCMC algorithm. The following sections provide more details about these directions and the solutions proposed in the literature.

1.1 | What is MCMC and why does it need accelerating?

MCMC methods have a history (e.g., Cappé & Robert, 2000) that starts at approximately the same time as the Monte Carlo methods, in conjunction with the conception of the first computers. They have been devised to handle the simulation of complex target distributions, when complexity stems from the shape of the target density, the size of the associated data, the dimension of the object to be simulated, or from time requirements. For instance, the target density $\pi(\theta)$ may happen to be expressed in terms of multiple integrals that cannot be solved analytically,

$$\pi(\theta) = \int \omega(\theta, \xi)\mathrm{d}\xi$$

which requires the simulation of the entire vector (θ, ξ). In cases when ξ is of the same dimension as the data, as for instance in latent variable models, this significant increase in the dimension of the object to be simulated creates computational difficulties for standard Monte Carlo methods, from managing the new target $\omega(\theta, \xi)$, to devise a new and efficient simulation algorithm. A MCMC algorithm allows for an alternative resolution of this computational challenge by simulating a Markov chain that explores the space of interest (and possibly supplementary spaces of auxiliary variables) without requiring a deep preliminary knowledge on the density π, besides the ability to compute $\pi(\theta_0)$ for a given parameter value θ_0 (if up to a normalizing constant) and possibly the gradient $\nabla \log \pi(\theta_0)$. The validation of the method (e.g., Robert & Casella, 2004) is that the Markov chain is *ergodic* (e.g., Meyn & Tweedie, 1993), namely that it converges in distribution to the distribution with density π, no matter where the Markov chain is started at time $t = 0$.

The Metropolis–Hastings algorithm is a generic illustration of this principle. The basic algorithm is constructed by choosing a *proposal*, that is, a conditional density $K(\theta^{'}|\theta)$ (also known as a *Markov kernel*), the Markov chain $\{\theta_t\}_{t=1}^{\infty}$ being then derived by successive simulations of the transition.

$$\theta_{t+1} = \begin{cases} \theta' \sim K(\theta'|\theta_t) & \text{with probability } \left\{\dfrac{\pi(\theta')}{\pi(\theta_t)} \times \dfrac{K(\theta_t|\theta')}{K(\theta'|\theta_t)}\right\} \wedge 1, \\ \theta_t & \text{otherwise.} \end{cases}$$

This acceptance–rejection feature of the algorithm makes it appropriate for targeting π as its stationary distribution if the resulting Markov chain $\{\theta_t\}_{t=1}^{\infty}$ is irreducible, that is, has a positive probability of visiting any region of the support of π in a finite number of iterations. (Stationarity can easily be shown, e.g., by using the so-called *detailed balance property* that makes the chain time-reversible; see, e.g., Robert & Casella, 2004.)

Considering the initial goal of simulating samples from the target distribution π, the performances of MCMC methods like the Metropolis–Hastings algorithm above often vary quite a lot, depending primarily on the correspondance between the proposal K and the target π. For instance, if $K(\theta|\theta_t) = \pi(\theta)$, the Metropolis–Hastings algorithm reduces to i.i.d. sampling from the target, which is of course a formal option when i.i.d. sampling from π proves impossible to implement. Although there exist rare instances when the Markov chain $\{\theta_t\}_{t=1}^{\infty}$ leads to negative correlations between the successive terms of the chain, making it *more efficient* than regular i.i.d. sampling (Liu, Wong, & Kong, 1995), the most common occurrence is one of positive correlation between the simulated values (sometimes uniformly, see Liu, Wong, & Kong, 1994). This feature implies a reduced efficiency of the algorithm and hence requires a larger number of simulations to achieve the same precision as an approximation based on i.i.d. simulations (without accounting for differences in computing time). More generally, a MCMC algorithm may require a large number of iterations to escape the attraction of its starting point θ_0 and to reach stationarity, to the extent that some versions of such algorithms fail to converge in the time available (i.e., in practice if not in theory).

It thus makes sense to seek ways of accelerating (a) the convergence of a given MCMC algorithm to its stationary distribution, (b) the convergence of a given MCMC estimate to its expectation, and/or (c) the exploration of a given MCMC algorithm of the support of the target distribution. Those goals are related but still distinct. For instance, a chain initialized by simulating from the target distribution may still fail to explore the whole support in an acceptable number of iterations. While there is not an optimal and universal solution to this issue, below we will discuss approaches that are as generic as possible, as opposed to artificial ones taking advantage of the mathematical structure of a specific target distribution. Ideally, we aim at covering realistic situations when the target density is only known (up to a constant or an additional completion step) as the output of an existing computer code. Pragmatically, we also cover here solutions that require more efforts and calibration steps when they apply to a wide enough class of problems.

1.2 | Accelerating MCMC by exploiting the geometry of the target

While there is no end in trying to construct more efficient and faster MCMC algorithms, and while this (endless) goal needs to account for the cost of devising such alternatives under limited resources budgets, there exist several generic solutions such that a given target can first be explored in terms of the geometry (or topology) of the density before constructing the algorithm. Although this type of methods somehow takes us away from our original purpose which was to improve upon an existing algorithm, they still make sense within this survey in that they allow for almost automated implementations.

1.3 | Hamiltonian Monte Carlo

From the point of view of this review, Hamiltonian (or hybrid) Monte Carlo (HMC) is an auxiliary variable technique that takes advantage of a continuous time Markov process to sample from the target π. This approach comes from physics (Duane, Kennedy, Pendleton, & Roweth, 1987) and was popularized in statistics by Neal (1999, 2011) and MacKay (2002). Given a target $\pi(\theta)$, where $\theta \in \mathbb{R}^d$, an artificial auxiliary variable $\vartheta \in \mathbb{R}^d$ is introduced along with a density $\varpi(\vartheta|\theta)$ so that the joint distribution of (θ, ϑ) enjoys $\pi(\theta)$ as its marginal. While there is complete freedom in this representation, the HMC literature often calls ϑ the *momentum* of a particle located at θ by analogy with physics. Based on the representation of the joint distribution.

$$\omega(\theta,\vartheta) = \pi(\theta)\varpi(\vartheta|\theta) \propto \exp\{-H(\theta,\vartheta)\}$$

where $H(\cdot)$ is called the *Hamiltonian*, Hamiltonian Monte Carlo (HMC) is associated with the continuous time process (θ_t, ϑ_t) generated by the so-called *Hamiltonian equations.*

$$\frac{\mathrm{d}\theta_t}{\mathrm{d}t} = \frac{\partial H}{\partial \vartheta}(\theta_t,\vartheta_t) \qquad \frac{\mathrm{d}\vartheta_t}{\mathrm{d}t} = -\frac{\partial H}{\partial \theta}(\theta_t,\vartheta_t)$$

which keep the Hamiltonian target stable over time, as.

$$\frac{\mathrm{d}H(\theta_t,\vartheta_t)}{\mathrm{d}t} = \frac{\partial H}{\partial \vartheta}(\theta_t,\vartheta_t)\frac{\mathrm{d}\vartheta_t}{\mathrm{d}t} + \frac{\partial H}{\partial \theta}(\theta_t,\vartheta_t)\frac{\mathrm{d}\theta_t}{\mathrm{d}t} = 0.$$

Obviously, the above continuous time Markov process is deterministic and only explores a given level set,

$$\{(\theta,\vartheta) : H(\theta,\vartheta) = H(\theta_0,\vartheta_0)\},$$

instead of the whole augmented state space $\mathbb{R}^{2d}$, which induces an issue with irreducibility. An acceptable solution to this problem is to refresh the momentum, $\vartheta_t \sim \varpi(\vartheta|\theta_{t-})$, at random times $\{\tau_n\}_{n=1}^{\infty}$, where θ_{t-} denotes the location of θ immediately prior to time t, and the random durations $\{\tau_n - \tau_{n-1}\}_{n=2}^{\infty}$ follow an exponential distribution. By construction, continuous-time Hamiltonian Markov chain can be regarded as a specific piecewise deterministic Markov process using Hamiltonian dynamics (Davis, 1984, 1993; Bou-Rabee et al., 2017) and our target, π, is the marginal of its associated invariant distribution.

Before moving to the practical implementation of the concept, let us point out that the free cog in the machinery is the conditional density $\varpi(\vartheta|\theta)$, which is usually chosen as a Gaussian density with either a constant covariance matrix M corresponding to the target covariance or as a local curvature depending on θ in Riemannian HMC (Girolami & Calderhead, 2011). Betancourt (2017) argues in favor of these two cases against non-Gaussian alternatives and Livingstone, Faulkner, and Roberts (2017) analyze how different choices of kinetic energy in HMC affect algorithm performances. For a fixed covariance matrix, the Hamiltonian equations become:

$$\frac{\mathrm{d}\theta_t}{\mathrm{d}t} = M^{-1}\vartheta_t \qquad \frac{\mathrm{d}\vartheta_t}{\mathrm{d}t} = \nabla\mathcal{L}(\theta_t)$$

which is the score function. The velocity (or momentum) of the process is thus driven by this score function, gradient of the log-target.

The above description remains quite conceptual in that there is no generic methodology for producing this continuous time process, since the Hamiltonian equations cannot be solved exactly in most cases. Furthermore, standard numerical solvers like Euler's method create an instable approximation that induces a bias as the process drifts away from its true trajectory. There exists, however, a discretization simulation technique that produces a Markov chain and is well-suited to the Hamiltonian equations, in that it preserves the stationary distribution (Betancourt, 2017). It is called the *symplectic integrator*, and one version in the independent case with constant covariance consists in the following (so-called *leapfrog*) steps

$$\begin{aligned}
\vartheta_{t+\epsilon/2} &= \vartheta_t + \epsilon\nabla\mathcal{L}(\theta_t)/2,\\
\theta_{t+\epsilon} &= \theta_t + \epsilon M^{-1}\vartheta_{t+\epsilon/2},\\
\vartheta_{t+\epsilon} &= \vartheta_{t+\epsilon/2} + \epsilon\nabla\mathcal{L}(\theta_{t+\epsilon})/2,
\end{aligned}$$

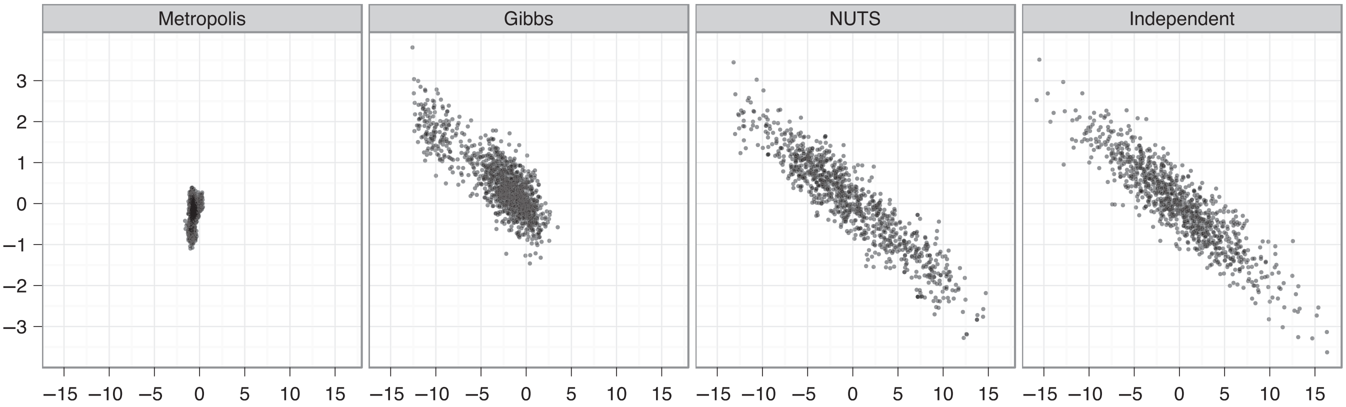

FIGURE 1 Comparisons between random-walk Metropolis-Hastings, Gibbs sampling, and NUTS algorithm of samples corresponding to a highly correlated 250-dimensional multivariate Gaussian target. Similar computation budgets are used for all methods to produce the 1,000 samples on display. Source*:* Hoffman and Gelman (2014)

where ϵ is the time-discretization step. Using a proposal on ϑ_0 drawn from the Gaussian auxiliary target and deciding on the acceptance of the value of $(\theta_{T\epsilon}, \vartheta_{T\epsilon})$ by a Metropolis–Hastings step can limit the danger of missing the target. Note that the first two leapfrog steps induce a Langevin move on θ_t:

$$\theta_{t+\epsilon} = \theta_t + \epsilon^2 M^{-1} \nabla \mathcal{L}(\theta_t)/2 + \epsilon M^{-1} \vartheta_t$$

thus connecting with the Metropolis-adjusted Langevin algorithm (MALA) discussed below (see Durmus and Moulines, 2017 for a theoretical discussion of the optimal choice of ϵ). Note that the leapfrog integrator is quite an appealing middleground between accuracy (as it is second-order accurate) and computational efficiency.

In practice, it is important to note that discretizing the Hamiltonian dynamics introduces two free parameters, the step size ϵ and the trajectory length $T\epsilon$, both to be calibrated. As an empirically successful and popular variant of HMC, the "no-U-turn sampler" (NUTS) of Hoffman and Gelman (2014) adapts the value of ϵ based on primal-dual averaging. It also eliminates the need to choose the trajectory length T via a recursive algorithm that builds a set of candidate proposals for a number of forward and backward leapfrog steps and stops automatically when the simulated path steps back.

A further acceleration step in this area is proposed by Rasmussen (2003) (see also Fielding, Nott, & Liong, 2011), namely the replacement of the exact target density $\pi(\cdot)$ by an approximation $\hat{\pi}(\cdot)$ that is much faster to compute in the many iterations of the HMC algorithm. A generic way of constructing this approximation is to rely on Gaussian processes, when interpreted as prior distributions on the target density $\pi(\cdot)$, which is only observed at some values of θ, $\pi(\theta_1)$, …, $\pi(\theta_n)$ (Rasmussen and Williams, 2005). This solution is speeding up the algorithm, possibly by orders of magnitude, but it introduces a further approximation into the Monte Carlo approach, even when the true target is used at the end of the leapfrog discretization, as in Fielding et al. (2011).

Stan (named after Stanislas Ullam, see Carpenter et al., 2017) is a computer language for Bayesian inference that, among other approximate techniques, implements the NUTS algorithm to remove hand-tuning. More precisely, Stan is a probabilistic programming language in that the input is at the level of a statistical model, along with data, rather than the specifics of an MCMC algorithm. The algorithmic part is somehow automated, meaning that when models can be conveniently defined through this language, it offers an alternative to the sampler that produced the original chain. As an illustration of the acceleration brought by HMC, Figure 1, reproduced from Hoffman and Gelman (2014), shows the performance of NUTS, compared with both random-walk MH and Gibbs samplers.

1.4 | Accelerating MCMC by breaking the problem into pieces

The explosion in the collection and analysis of "big" data sets in recent years has brought new challenges to the MCMC algorithms that are used for Bayesian inference. When examining whether or not a new proposed sample is accepted at the accept–reject step, an MCMC algorithm such as the Metropolis–Hastings version needs to sweep over the whole data set, at each and every iteration, for the evaluation of the likelihood function. MCMC algorithms are then difficult to scale up, which strongly hinders their application in big data settings. In some cases, the data sets may be too large to fit on a single machine. It may also be that confidentiality measures impose different databases to stand on separate networks, with the possible added burden of encrypted data (Aslett, Esperança, & Holmes, 2015). Communication between the separate machines may prove impossible on an MCMC scale that involves thousands or hundreds of thousands of iterations.

1.5 | Scalable MCMC methods

In the recent years, efforts have been made to design *scalable* algorithms, namely, solutions that manage to handle large-scale targets by breaking the problem into manageable or scalable pieces. Roughly speaking, these methods can be classified into two categories (Bardenet, Doucet, & Holmes, 2015): divide-and-conquer approaches and subsampling approaches.

Divide-and-conquer approaches partition the whole data set, denoted $\mathcal{X}$, into batches, $\{\mathcal{X}_1,\cdots,\mathcal{X}_k\}$, and run separate MCMC algorithms on each data batch, independently, as if they were independent Bayesian inference problems.[1] These methods then combine the simulated parameter outcomes together to approximate the original posterior distribution. Depending on the treatments of the batches selected in the MCMC stages, these approaches can be further subdivided into two finer groups: subposterior methods and boosted subposterior methods. Subposterior methods are motivated by the independent product equation:

$$\pi(\theta) \propto \prod_{i=1}^{k}\left(\pi_0(\theta)^{1/k}\prod_{\ell\in\mathcal{X}_i}p(x_\ell|\theta)\right) = \prod_{i=1}^{k}\pi_i(\theta) \tag{1}$$

and they target the densities $\pi_i(\theta)$ (up to a constant) in their respective MCMC steps. They thus bypass communication costs (Scott et al., 2016), by running MCMC samplers independently on each batch, and they most often increase MCMC mixing rates (in effective samples sizes produced by second), given that the subposterior distributions $\pi_i(\theta)$ are based on smaller data sets. For instance, Scott et al. (2016) combine the samples from the subposteriors, $\pi_i(\theta)$, by a Gaussian reweighting. Neiswanger, Wang, and Xing (2013) estimate the subposteriors $\pi_i(\theta)$ by nonparametric and semi-parametric methods, and they run additional MCMC samplers on the product of these estimators toward approximating the true posterior $\pi(\theta)$. Wang and Dunson (2013) refine this product estimator with an additional Weierstrass sampler, while Wang, Guo, Heller, and Dunson (2015) estimate the posterior by partitioning the space of samples with step functions.

As an alternative to sampling from the subposteriors, boosted subposterior methods target instead the components

$$\widetilde{\pi}_i(\theta) \propto \pi_0(\theta)\left(\prod_{\ell\in\mathcal{X}_i}p(x_\ell|\theta)\right)^k \tag{2}$$

in separate MCMC runs. Since they formaly amount to repeating each batch k times toward producing pseudo data sets with the same size as the true one, the resulting boosted subposteriors, $\widetilde{\pi}_1(\theta),\cdots,\widetilde{\pi}_k(\theta)$, have the same scale in variance of each component of the parameters, θ, as the true posterior, and can thus be treated as a group of estimators of the true posterior. In the subsequent combining stage, these subposteriors are merged together to construct a better approximation of the target distribution. For instance, Minsker, Srivastava, Lin, and Dunson (2014) approximate the posterior with the geometric median of the boosted subposteriors, embedding them into associated reproducing kernel Hilbert spaces, while Srivastava, Cevher, Dinh, and Dunson (2015) achieve this goal using the barycenters of $\widetilde{\pi}_1,\cdots,\widetilde{\pi}_k$, these barycenters being computed with respect to a Wasserstein distance.

In a perspective different from the above parallel scheme of divide-and-conquer approaches, subsampling approaches aim at reducing the number of individual datapoint likelihood evaluations operated at each iteration toward accelerating MCMC algorithms. From a general perspective, these approaches can be further classified into two finer classes: exact subsampling methods and approximate subsampling methods, depending on their resulting outputs. Exact subsampling approaches typically require subsets of data of random size at each iteration. One solution to this effect is taking advantage of pseudo-marginal MCMC via constructing unbiased estimators of the target density evaluated on subsets of the data (Andrieu & Roberts, 2009). Quiroz, Villani, and Kohn (2016) follow this direction by combining the powerful debiasing technique of Rhee and Glynn (2015) and the correlated pseudo-marginal MCMC approach of Deligiannidis, Doucet, and Pitt (2015). Another direction is to use piecewise deterministic Markov processes (PDMP) (Davis, 1984, 1993), which enjoy the target distribution as the marginal of their invariant distribution. This PDMP version requires unbiased estimators of the gradients of the log-likelihood function, instead of the likelihood itself. By using a tight enough bound on the event rate function of the associated Poisson processes, PDMP can produce super-efficient scalable MCMC algorithms. The bouncy particle sampler (Bouchard-Côté, Vollmer, & Doucet, 2017) and the zig-zag sampler (Bierkens, Fearnhead, & Roberts, 2016) are two competing PDMP algorithms, while Bierkens et al. (2017) unify and extend these two methods. Besides, one should note that PDMP produces a non-reversible Markov chain, which means that the algorithm should be more efficient in terms of mixing rate and asymptotic variance, when compared with reversible MCMC algorithms, such as MH, HMC, and MALA, as observed in some theoretical and experimental works (Bierkens, 2016; Chen & Hwang, 2013; Hwang, Hwang-Ma, & Sheu, 1993; Sun, Gomez, & Schmidhuber, 2010).

Approximate subsampling approaches aim at constructing an approximation of the target distribution. Beside the aforementioned attempts of Rasmussen (2003) and Fielding et al. (2011), one direction is to approximate the acceptance probability

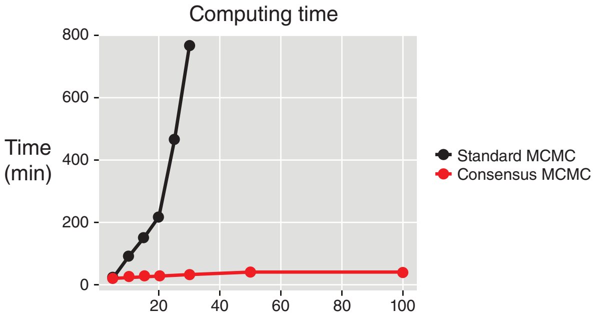

FIGURE 2 Elapsed time when drawing 10,000 MCMC samples with different amounts of data under the single machine and consensus Monte Carlo algorithms for a hierarchical Poisson regression. The horizontal axis represents the amounts of data. The single machine algorithm stops after 30 because of the explosion in computation budget. Source*:* Scott et al. (2016)

with high accuracy by using subsets of the data (Bardenet et al., 2015; Bardenet, Doucet, & Holmes, 2014). Another solution is based on a direct modification of exact methods. The seminal work of Welling and Teh (2011), stochastic gradient Langevin dynamics (SGLD), is to exploit the Langevin diffusion

$$d\theta_t = \frac{1}{2}\mathbf{\Lambda}\nabla\log\pi(\theta_t)dt + \mathbf{\Lambda}^{1/2}d\mathbf{B}_t, \quad \theta_0\in\mathbb{R}^d, t\in[0,\infty) \tag{3}$$

where $\mathbf{\Lambda}$ is a user-specified matrix, π is the target distribution, and $\mathbf{B}_t$ is a d-dimensional Brownian process. By virtue of the Euler–Maruyama discretization and using unbiased estimators of the gradient of the log-target density, SGLD and its variants (Chen, Fox, & Guestrin, 2014; Ding et al., 2014) often produce fast and accurate results in practice when compared with MCMC algorithms using MH steps.

Figure 2 shows the time requirements of a consensus Monte Carlo algorithm (Scott et al., 2016) compared with a Metropolis–Hastings algorithm using the whole data set, while Figure 3 displays the saving in likelihood evaluations in confidence sampler of Bardenet et al. (2015).

1.6 | Parallelization and distributed schemes

Modern computational architectures are built with several computing units that allow for parallel processing, either fully independent or with certain communication. Although the Markovian nature of MCMC is inherently sequential and somewhat alien to the notion of parallelizing, several partial solutions have been proposed in the literature for exploiting these parallel architectures. The simplest approach consists in running several MCMC chains in parallel, blind to all others, until the allotted computing time is exhausted. Finally, the resulting estimators of all chains are averaged. However, this naive implementation may suffer from the fact that some of those chains have not reached their stationary regime by the end of the computation time, which then induces a bias in the resulting estimate. Ensuring that stationarity has been achieved is a difficult (if at all possible) task, although several approaches can be found in the literature (Guihenneuc-Jouyaux & Robert, 1998; Jacob, O'Leary, & Atchadé, 2017; Mykland, Tierney, & Yu, 1995). At the opposite extreme, complex targets may be represented as products that involve many terms that must be evaluated, each of which can be attributed to a different thread before being multiplied all together. This strategy requires communication among processors at each MCMC step. A middle-ground version (Jacob, Robert, & Smith, 2011) consists in running several Markov chains in parallel with periodic choices of the reference chain, all simulations being recycled through a Rao–Blackwell scheme. (See also Calderhead, 2014 for a similar scheme.) The family of interacting *orthogonal* MCMC methods (O-MCMC) is proposed in Martino, Elvira, Luengo, Corander, and Louzada (2016) with the aim of fostering better exploration of the state space, specially in high-dimensional and multimodal targets. Multiple MCMC chains are run in parallel exploring the space with random-walk proposals. The parallel

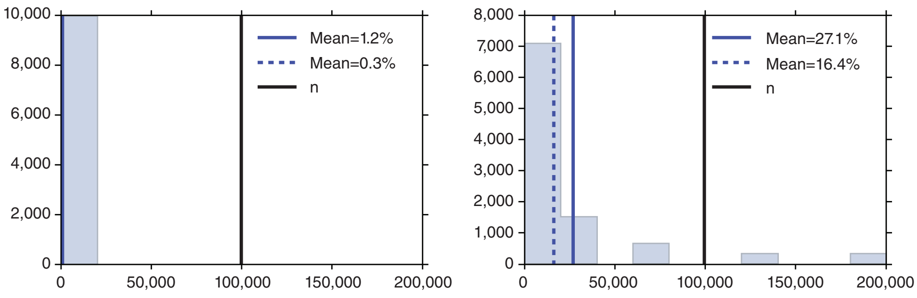

FIGURE 3 Percentage of numbers of data points used in each iteration of the confidence sampler with a single 2nd-order Taylor approximation at θ_{MAP}. The plots describe 10,000 iterations of the confidence sampler for the posterior distribution of the mean and variance of a unidimensional normal distribution with a flat prior: (left) 10,000 observations are generated from $\mathcal{N}(0,1)$, (right) 10,000 observations are generated from $\mathcal{LN}(0,1)$. Source: Bardenet et al. (2015)

chains periodically share information, also through joint MCMC steps, thus allowing an efficient combination of global (coordinated) exploration and local approximation. O-MCMC methods also allow for a parallel implementation of the Multiple Try Metropolis. In Calderhead (2014), a generalization of the Metropolis-Hastings algorithm allows for a straightforward parallelization. Each proposed point can be evaluated in a different processor at every MCMC iteration. Finally, note that the section on scalable MCMC also contains parallelizable approaches, such as the prefetching method of Angelino, Kohler, Waterland, Seltzer, and Adams (2014) (see also Banterle, Grazian, Lee, & Robert, 2015 for a related approach, primarily based on an approximation of the target). A most recent endeavor called asynchronous MCMC (Terenin, Simpson, & Draper, 2015) aims at higher gains in parallelization by reducing the amount of exchange between the parallel threads, but the notion still remains confidential at this stage.

1.7 | Accelerating MCMC by improving the proposal

In the same spirit as the previous section, this section is stretching the purpose of this paper by considering possible modifications of the MCMC algorithm itself, rather than merely exploiting the output of a given MCMC algorithm. For instance, devising an HMC algorithm is an answer to this question even though the "improvement" is not garanteed. Nonetheless, our argument here is that, once provided with this output, it is possible to derive new proposals in a semi-autonomous manner.

1.8 | Simulated tempering

The target distribution, $\pi(\theta)$ on d-dimensional state space Θ, can exhibit multimodality with the probability mass being located in different regions in the state space. The majority of MCMC algorithms use a localized proposal mechanism which is tuned toward local approximate optimality see, for example, Roberts, Gelman, and Gilks (1997) and Roberts and Rosenthal (2001). By construction, these localized proposals result in the Markov chain becoming "trapped" in a subset of the state space meaning that in finite run-time the chain can entirely fail to explore other modes in the state space, leading to biased samples. Strategies to accelerate MCMC often use local gradient information and this draws the chain back toward the center of the mode, which is the opposite of what is required in a multimodal setting.

There is an array of methodology available to overcome issues of multimodality in MCMC, the majority of which use state space augmentation. Auxiliary distributions that allow a Markov chain to explore the entirety of the state space are targeted and their mixing information is then passed on to aid mixing in the true target. While the subposteriors of the previous section can be seen as special cases of the following, the most successful and convenient implementation of these methods is to use *power-tempered target distributions.* The target distribution at inverse temperature level, β, for $\beta \in (0, 1]$ is defined as

$$\pi_\beta(\theta) = \mathfrak{K}(\beta)[\pi(\theta)]^\beta \text{ where } \mathfrak{K}(\beta) = \left[\int [\pi(\theta)]^\beta d\theta\right]^{-1}.$$

Therefore, $\pi_1(\theta) = \pi(\theta)$. Temperatures $\beta < 1$ flatten out the target distribution allowing the chain to explore the entire state space provided the β value is sufficiently small. The simulated tempering (ST) and parallel tempering (PT) algorithms (Geyer, 1991; Marinari & Parisi, 1992) typically use the power-tempered targets to overcome the issue of multimodality. The ST approach runs a single Markov chain on the augmented state space $\{B, \Theta\}$, where $B = \{\beta_0, \beta_1, \ldots, \beta_n\}$ is a discrete collection of n inverse temperature levels with $1 = \beta_0 > \beta_1 > \ldots > \beta_n > 0$. The algorithm uses a Metropolis-within-Gibbs strategy by cycling between updates in the Θ and B components of the space. For instance, a proposed temperature swap move $\beta_i \to \beta_j$ is accepted with probability

$$\min\left\{1, \frac{\pi_{\beta_j}(\theta)}{\pi_{\beta_i}(\theta)}\right\}$$

in order to preserve detailed balance. Note that this acceptance ratio depends on the normalization constants $\mathfrak{K}(\beta)$ which are typically unknown, although they can sometimes be estimated, as in, for example, Wang and Landau (2001) and Atchadé and Liu (2004). In case estimation of the marginal normalization constants is impractical then the PT algorithm is employed. This approach simultaneously runs a Markov chain at each of the $n + 1$ temperature levels targeting the joint distribution given by $\prod_{i=0}^n [\pi(\theta_i)]^{\beta i}$. Swap moves between chains at adjacent temperature levels are accepted according to a ratio that no longer depends on the marginal normalization constants. Indeed, this power tempering approach has been successfully employed in a number of settings and is widely used for example, Neal (1996), Earl and Deem (2005), Xie, Zhou, and Jiang (2010), Mohamed, Calderhead, Filippone, Christie, and Girolami, (2012) and Carter and White (2013).

In both approaches, there is a "Goldilocks" principle to setting up the inverse temperature schedule. Spacings between temperature levels that are "too large" result in swap moves that are rarely accepted, hence delaying the transfer of hot state mixing information to the cold states. On the other hand, spacings that are too small require a large number of intermediate

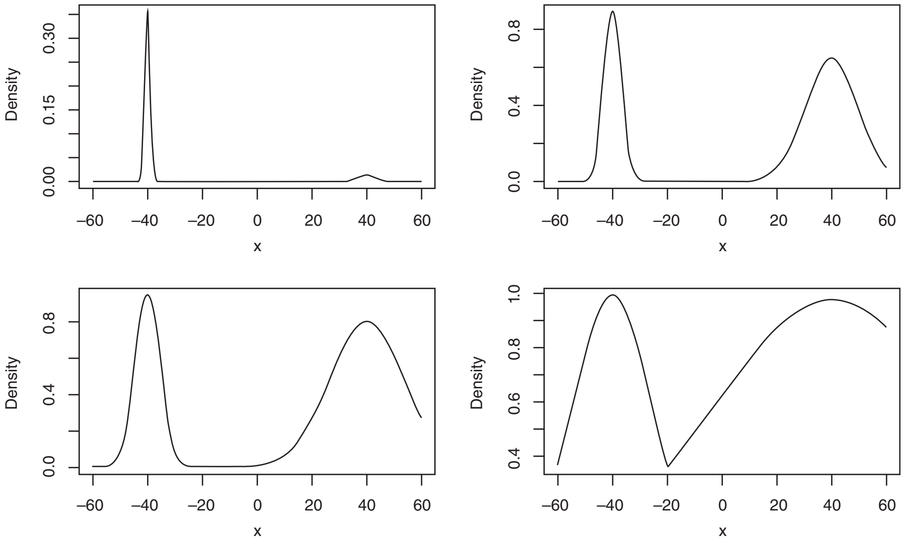

FIGURE 4 Unnormalized tempered target densities of a bimodal Gaussian mixture using inverse temperature levels $\beta = \{1, .1, .05, .005\}$, respectively. At the hot state (bottom right) it is evident that the mode centred on 40 begins to dominate the weight as β increases to ∞ even though at the cold state it was only attributable for a fraction (.2) of the total mass

temperature levels, again resulting in slow mixing through the temperature space. This problem becomes even more difficult as the dimensionality of Θ increases.

Much of the historical literature suggested that a geometric spacing was optimal that is, there exists $c \in (0, 1)$ such that $\beta_{i+1} = c\beta_i$ for $i = 0, 1, \ldots, n$. However, in the case of the ST version, Atchadé, Roberts, and Rosenthal (2011) considered the problem as an optimal scaling problem by maximizing the (asymptotic in dimension) expected squared jumping distance in the B space for temperature swap moves. Under restrictive assumptions, they showed that the spacings between consecutive inverse temperature levels should scale with dimension as $O(d^{-1/2})$ to prevent degeneracy of the swap move acceptance rate. For a practitioner the result gave guidance on optimal setup since it suggested a corresponding optimal swap move acceptance rate of 0.234 between consecutive inverse temperature levels, in accordance with Gelman, Gilks, and Roberts (1996). Finally, contrary to the historically recommended geometric schedule, the authors suggested that temperature schedule setup should be constructed consecutively so as to induce an approximate 0.234 swap acceptance rate between consecutive levels; which is achieved adaptively in Miasojedow, Moulines, and Vihola (2013). The use of expected squared jumping distance as the measure of mixing speed was justified in Roberts and Rosenthal (2014) where, under the same conditions as in Atchadé et al. (2011), it was shown that the temperature component of the ST chain has an associated diffusion process.

The target of an 0.234 acceptance rate gives good guidance to setting up the ST/PT algorithms in certain settings, but there is a major warning for practitioners following this rule for optimal setup. The assumptions made in Atchadé et al. (2011) and Roberts and Rosenthal (2014) ignore the restrictions of mixing within a temperature level, instead assuming that this can be done infinitely fast relative to the mixing within the temperature space. Woodard, Schmidler, and Huber (2009a, 2009b)and Bhatnagar and Randall (2016) undertake a comprehensive analysis of the spectral gap of the ST/PT chains and their conclusion is rather damning of the ST/PT approaches that use power-tempered targets. Essentially, in situations where the modes have different structures, the time required to reach a given level of convergence for the ST/PT algorithms can grow exponentially in dimension. A major reason for this is that power-based tempering does not preserve the relative weights/mass between regions at the different temperature levels, see Figure 4. This issue can scale exponentially in dimension. From a practical perspective, in these finite run high-dimensional nonidentical modal structure settings the swap acceptance rates can be very misleading, meaning that they have limited use as a diagnostic for intermodal mixing quality.

1.9 | Adaptive MCMC

Improving and calibrating an MCMC algorithm toward a better correspondance with the intended target is a natural step in making the algorithm more efficient, provided enough information is available about this target distribution. For instance, when an MCMC sample associated with this target is available, even when it has not fully explored the range of the target, it contains some amount of information, which can then be exploited to construct new MCMC algorithms. Some of the solutions available in the literature (e.g., Liang, Liu, & Carroll, 2007) proceed by repeating blocks of MCMC iterations and updating the proposal K after each block, aiming at a particular optimality goal like a specific acceptance rate like 0.234 for Metropolis–Hastings steps (Gelman et al., 1996). Most versions of this method update the scale structure of a random walk proposal, based on previous realizations (Robert & Casella, 2009) or on an entire sample (Douc, Guillin, Marin, & Robert, 2007), which turns the method into iterated importance sampling with Markovian dependence. (It can also be seen as a static version of particle filtering, Doucet, Godsill, & Andrieu, 2000; Andrieu & Doucet, 2002; Storvik, 2002.)

Other adaptive resolutions bypass this preliminary and somewhat ad hoc construction and aim instead at a permanent updating within the algorithm, motivated by the idea that a continuous adaptation keeps improving the correspondance with the target. In order to preserve the validation of the method (Gelman et al., 1996; Haario, Saksman, & Tamminen, 1999; Roberts & Rosenthal, 2007; Saksman & Vihola, 2010), namely that the chain produced by the algorithm converges to the intended target, specific convergence results need be established, as the ergodic theorem behind standard MCMC algorithms does not apply. Without due caution (see Figure 5), an adaptive MCMC algorithm may fail to converge due to over-fitting. A drawback of adaptivity is that the update of the proposal distribution relies *too much* on the earlier simulations and thus reinforces the exclusion of parts of the space that have not yet been explored.

For the validation of adaptive MCMC methods, stricter constraints must thus be imposed on the algorithm. One well-described solution (Roberts & Rosenthal, 2009) is called *diminishing adaptation.* Informally, it consists in imposing a distance between two consecutive proposal kernels to uniformly decrease to zero. In practice, this means stabilizing the changes in the proposal by ridge-like factors as in the early proposal by Haario et al. (1999). A drawback of this resolution is that the decrease itself must be calibrated and may well fail to bring a significant improvement over the original proposal.

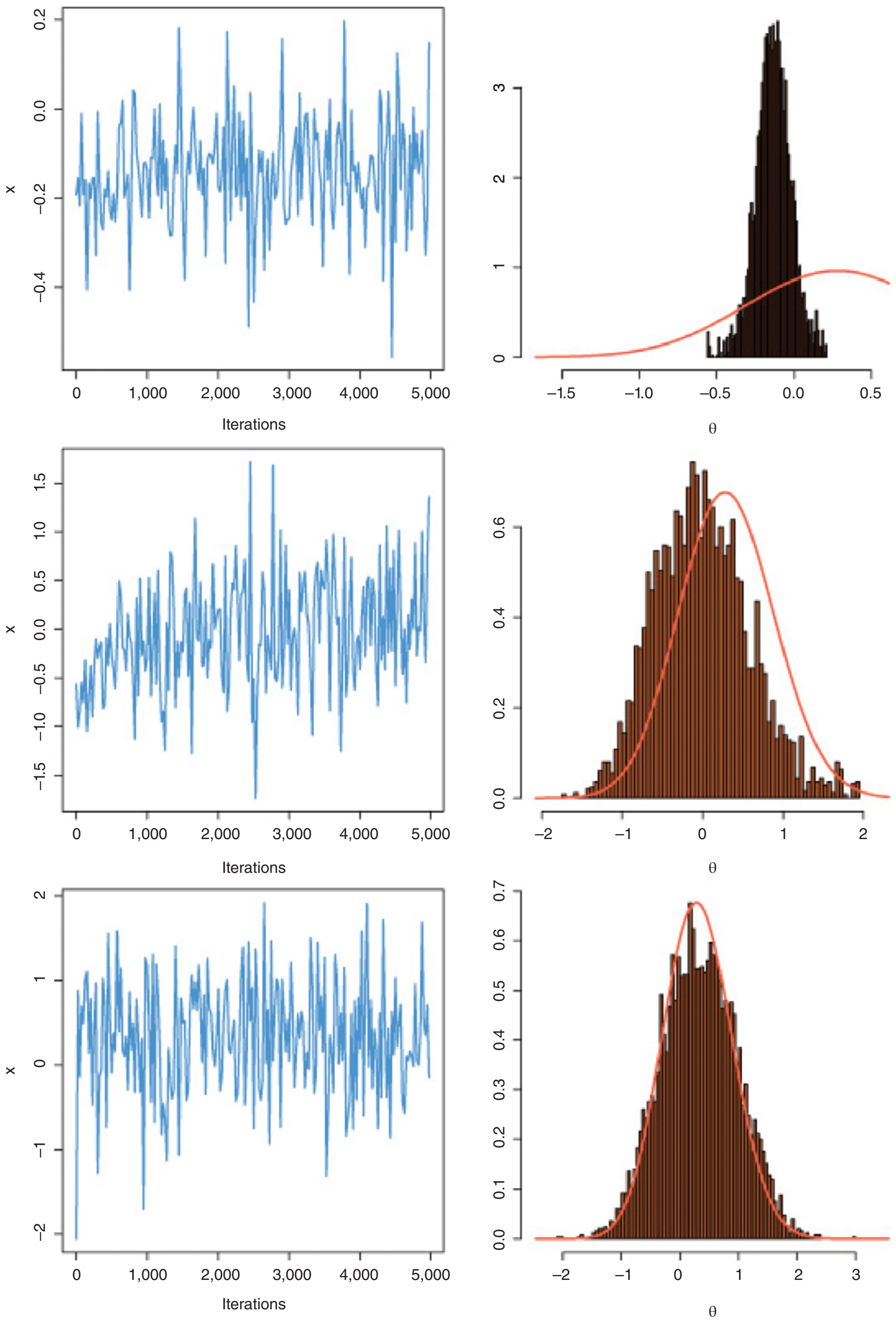

FIGURE 5 Markov chains produced by an adaptive algorithm where the proposal distribution is a Gaussian distribution with mean and variance computed from the past simulations of the chain. The three rows correspond to different initial distributions. The fit of the histogram of the resulting MCMC sample is poor, even for the most spread-out initial distribution (bottom). Source: Robert and Casella (2004)

1.10 | Multiple try MCMC

A completely different approach to improve the original proposal used in an MCMC algorithm is to consider a collection of proposals, built on different rationales and experiments. The *multiple try MCMC algorithm* (Liu, Liang, & Wong, 2000; Bédard, Douc, & Moulines, 2012; Martino, 2018) follows this perspective. As the name suggests, the starting point of a multiple try MCMC algorithm is to simultaneously propose N potential moves $\theta_t^1, \ldots, \theta_t^N$ of the Markov chain, instead of a single value. The proposed values θ_t^i may be independently generated according to N different proposal densities $K_i(\cdot|\theta_t)$ that are conditional on the current value of the Markov chain, θ_t. One of the θ_t^i's is selected based on the importance sampling weights $w_t^i \propto \pi\left(\theta_t^i\right)/K_i(\cdot|\theta_t)$. The selected value is then accepted by a further Metropolis–Hastings step which involves a ratio of normalization constants for the importance stage, one corresponding to the selection made previously and another one created for this purpose. Indeed, besides the added cost of computing the sum of the importance weights and generating the different variates, this method faces the non-negligible drawback of requiring $N - 1$ supplementary simulations that are only used for achieving detailed balance and computing a backward summation of importance weights. This constraint may vanish when considering a collection of independent Metropolis-Hastings proposals, $q(\theta)$, but this setting is rarely realistic as it requires some amount of prior knowledge or experimentation to build a relevant distribution.

An alternative found in the literature is *ensemble Monte Carlo* (Iba, 2000; Cappé, Douc, Guillin, Marin, & Robert, 2008; Neal, 2011; Martino, 2018), illustrated in Figure 6 which produces a whole sample at each iteration, with target the product of the initial targets, in closer proximity with particle methods (Cappé, Guillin, Marin, & Robert, 2004; Mengersen & Robert, 2003).

Yet another implementation of this principle is called *delayed rejection* (Tierney & Mira, 1998; Mira, 2001; Mira & Sargent, 2003), where proposals are instead considered sequentially, once the previous proposed value has been rejected, to speed up MCMC by considering several possibilities, if sequentially. A computational difficulty with this approach is that the associated acceptance probabilities get increasingly complex as the number of delays grows, which may annihilate its appeal relative to simultaneous multiple tries. A further difficulty is to devise the sequence of proposals in a diverse enough manner.

1.11 | Accelerating MCMC by reducing the variance

Since the main goal of MCMC is to produce approximations for quantities of interest of the form,

$$\mathfrak{I}_h = \int_\Theta h(\theta)\pi(\theta)\mathrm{d}\theta,$$

an alternative (and cumulative) way of accelerating these algorithms is to improve the quality of the approximation derived from an MCMC output. That is, given an MCMC sequence $\theta_1, \ldots, \theta_T$, converging to $\pi(\cdot)$, one can go beyond resorting to the basic Monte Carlo approximation

$$\hat{\mathfrak{I}}_h^T = \frac{1}{T}\sum_{t=1}^{T} h(\theta_t) \tag{4}$$

toward reducing the variance (if not the speed of convergence) of $\hat{\mathfrak{I}}_h^T$ to $\mathfrak{I}_h$.

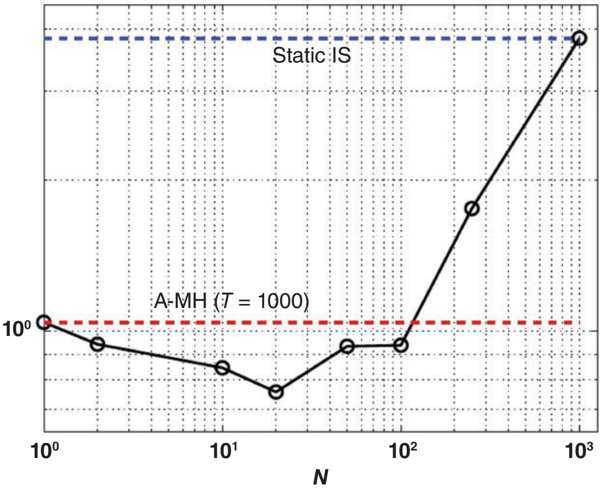

FIGURE 6 A comparison of an ensemble MCMC approach with a regular adaptive MCMC algorithm (lower line) and a static importance sampling approach, in terms of mean square error (MSE), for a fixed total number of likelihood evaluations, where N denotes the size of the ensemble. Source: Martino (2018)

A common remark when considering Monte Carlo approximations of $\mathfrak{I}_h$ is that the representation of the integral as an expectation is not unique (e.g., Robert & Casella, 2004). This leads to the technique of importance sampling where alternative distributions are used in replacement of $\pi(\theta)$, possibly in an adaptive manner (Douc et al., 2007), or sequentially as in particle filters (Andrieu, Doucet, & Holenstein, 2011; Del Moral, Doucet, & Jasra, 2006). Within the framework of this essay, the outcome of a given MCMC sampler can also be exploited in several ways that lead to an improvement of the approximation of $\mathfrak{I}_h$.

1.12 | Rao–Blackwellization and other averaging techniques

The name "Rao–Blackwellisation" was coined by Gelfand and Smith (1990) in their foundational Gibbs sampling paper and it has since then become a standard way of reducing the variance of integral approximations. While it essentially proceeds from the basic probability identity.

$$\mathbb{E}^{\pi}[h(\theta)] = \mathbb{E}^{\pi_1}[\mathbb{E}^{\pi_2}\{h(\theta)|\xi\}],$$

when π can be expressed as the following marginal density

$$\pi(\theta) = \int_{\Xi} \pi_1(\xi)\pi_2(\theta|\xi)\mathrm{d}\xi,$$

and while sufficiency does not have a clear equivalence for Monte Carlo approximation, the name stems from the Rao–Blackwell theorem (Lehmann & Casella, 1998) that improves upon a given estimator by conditioning upon a sufficient statistics. In a Monte Carlo setting, this means that Equation (4) can be improved by a partly integrated version

$$\tilde{\mathfrak{I}}_h^T = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}^{\pi_2}[h(\theta)|\xi^t] \tag{5}$$

assuming that a second and connected sequence of simulations (ξ_t) is available and that the conditional expectation is easily constructed. For instance, Gibbs sampling (Gelfand & Smith, 1990) is often open to this Rao–Blackwell decomposition as it relies on successive simulations from several conditional distributions, possibly including auxiliary variates and nuisance parameters. In particular, a generic form of Gibbs sampling called the slice sampler (Robert & Casella, 2004) produces one or several uniform variates at each iteration.

However, a more universal type of Rao–Blackwellization is available (Casella & Robert, 1996) for all MCMC methods involving rejection, first and foremost, Metropolis–Hastings algorithms. Indeed, first, the distribution of the rejected variables can be derived or approximated, which leads to an importance correction of the original estimator. Furthermore, the accept–reject step depends on a uniform variate, but this uniform variate can be integrated out. Namely, given a sample produced by a Metropolis–Hastings algorithm $\theta^{(1)}, \ldots, \theta^{(T)}$, one can exploit both underlying samples, the proposed values $\vartheta_1, \ldots, \vartheta_T$, and the uniform $u_1, \ldots, u_T$, so that the ergodic mean can be rewritten as

$$\hat{\mathfrak{I}}_h^T = \frac{1}{T}\sum_{t=1}^{T} h\left(\theta^{(t)}\right) = \frac{1}{T}\sum_{t=1}^{T} h(\vartheta_t)\sum_{i=t}^{T}\mathbb{I}_{\theta^{(i)}=\vartheta_t}.$$

The conditional expectation

$$\begin{aligned}\tilde{\mathfrak{I}}_h^T &= \frac{1}{T}\sum_{t=1}^{T} h(\vartheta_t)\mathbb{E}\left[\sum_{i=t}^{T}\mathbb{I}_{\theta^{(i)}=\vartheta_t}|\vartheta_1,\ldots,\vartheta_T\right]\\ &= \frac{1}{T}\sum_{t=1}^{T} h(\vartheta_t)\left\{\sum_{i=t}^{T}\mathbb{P}\left(\theta^{(i)}=\vartheta_t|\vartheta_1,\ldots,\vartheta_T\right)\right\}\end{aligned}$$

then enjoys a smaller variance. See also Tjelmeland (2004) and Douc and Robert (2010) for connected improvements based on multiple tries. An even more rudimentary (and cheaper) version can be considered by integrating out the decision step at each Metropolis–Hastings iteration: if θ_t is the current value of the Markov chain and ϑ_t the proposed value, to be accepted (as θ_{t+1}) with probability α_t, the version

$$\frac{1}{T}\sum_{t=1}^{T}\{\alpha_t h(\vartheta_t) + (1-\alpha_t)h(\theta_t)\}$$

should most often[2] bring an improvement over the basic estimate (Liu et al., 1995; Robert & Casella, 2004).

2 | CONCLUSIONS

Accelerating MCMC algorithms may sound like a new Achille versus tortoise paradox in that there are aways methods to speed up a given algorithm. The stopping rule of this infinite regress is, however, that the added pain in achieving this acceleration may overcome the added gain at some point. While we have only and mostly superficially covered some of the possible directions in this survey, we thus encourage most warmly readers to keep an awareness for the potential brought by a wide array of almost cost-free accelerating solutions as well as to keep trying devising more fine-tuned improvements in every new MCMC implementation. For instance, for at least one of us, Rao–Blackwellization is always considered at this stage. Keeping at least one such bag of tricks at one's disposal is thus strongly advised.

ACKNOWLEDGMENTS

C.P.R. is grateful to Gareth Roberts, Mike Betancourt, and Julien Stoehr for helpful discussions. He is currently supported by an Institut Universitaire de France 2016–2021 senior grant. N. T. was supported by the EPSRC grant EP/K014463/1 grant. C.W. is currently a Ph.D. candidate at Université Paris-Dauphine and supported by a grant of the Chinese Government (CSC). V.E. acknowledges support from the *Agence Nationale de la Recherche* of France under PISCES project (ANR-17-CE40-0031-01), the Fulbright program, and from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement number PCOFUND-GA-2013-609102, through the PRESTIGE program. The authors are quite grateful to a reviewer for his or her detailed coverage of an earlier version of the paper, which contributed to significant improvements in the presentation and coverage of the topic. All remaining errors and omissions are solely the responsability of the authors.

CONFLICT OF INTEREST

The authors have declared no conflicts of interest for this article.

NOTES

[1] In order to keep the notations consistent, we still denote the target density by π, with the prior density denoted as π_0 and the sampling distribution of one observation x as $p(x|\theta)$. The dependence on the sample $\mathcal{X}$ is not reported unless necessary.

[2] The improvement is not universal, due to the correlation between the terms of the sum induced by the Markovian nature of the sequence $\{\theta_t\}_{t=1}^{T}$.

RELATED WIREs ARTICLES

Bayesian computation: a summary of the current state, and samples backwards and forwards

REFERENCES

Andrieu, C., & Doucet, A. (2002). Particle filtering for partially observed Gaussian state space models. *Journal of Royal Statistical Society Series B*, *64*, 827–836.

Andrieu, C., Doucet, A., & Holenstein, R. (2011). Particle Markov chain Monte Carlo (with discussion). *Journal of Royal Statistical Society Series B*, *72*(2), 269–342.

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

Angelino, E., Kohler, E., Waterland, A., Seltzer, M., & Adams, R. (2014). Accelerating MCMC via parallel predictive prefetching. *arXiv preprint arXiv:1403.7265.*

Aslett, L., Esperança, P., & Holmes, C. (2015). A review of homomorphic encryption and software tools for encrypted statistical machine learning. *arXiv preprint arXiv:1508.06574*.

Atchadé, Y. F., & Liu, J. S. (2004). The Wang-Landau algorithm for Monte Carlo computation in general state spaces. *Statistica Sinica*, *20*, 209–233.

Atchadé, Y. F., Roberts, G., & Rosenthal, J. (2011). Towards optimal scaling of Metropolis-coupled Markov chain Monte Carlo. *Statistics and Computing*, *21*, 555–568.

Banterle, M., Grazian, C., Lee, A., & Robert, C. P. (2015). Accelerating Metropolis–Hastings algorithms by delayed acceptance. *arXiv preprint arXiv:1503.00996.*

Bardenet, R., Doucet, A., & Holmes, C. (2014). Towards scaling up Markov chain Monte Carlo: An adaptive subsampling approach. Paper presented at International Conference on Machine Learning (ICML), 405–413.

Bardenet, R., Doucet, A., & Holmes, C. (2015). On Markov chain Monte Carlo methods for tall data. *arXiv preprint arXiv:1505.02827.*

Bédard, M., Douc, R., & Moulines, E. (2012). Scaling analysis of multiple-try MCMC methods. *Stochastic Processes and their Applications*, *122*, 758–786.

Betancourt, M. (2017). A conceptual introduction to Hamiltonian Monte Carlo. *ArXiv e-prints: 1701.02434.*

Bhatnagar, N., & Randall, D. (2016). Simulated tempering and swapping on mean-field models. *Journal of Statistical Physics*, *164*, 495–530.

Bierkens, J. (2016). Non-reversible Metropolis-Hastings. *Statistics and Computing*, *26*, 1213–1228.

Bierkens, J., Bouchard-Côté, A., Doucet, A., Duncan, A. B., Fearnhead, P., Roberts, G., & Vollmer, S. J. (2017). Piecewise deterministic Markov processes for scalable Monte Carlo on restricted domains. *arXiv preprint arXiv:1701.04244.*

Bierkens, J., Fearnhead, P., & Roberts, G. (2016). The zig-zag process and super-efficient sampling for Bayesian analysis of big data. *arXiv preprint arXiv:1607.03188.*

Bou-Rabee, N., Sanz-Serna, J. M., et al. (2017). Randomized Hamiltonian Monte Carlo. *The Annals of Applied Probability*, *27*, 2159–2194.

Bouchard-Côté, A., Vollmer, S. J., & Doucet, A. (2017). The bouncy particle sampler: A non-reversible rejection-free Markov chain Monte Carlo method. *Journal of the American Statistical Association*, To appear.

Calderhead, B. (2014). A general construction for parallelizing Metropolis–Hastings algorithms. *Proceedings of the National Academy of Sciences*, *111*, 17408–17413.

Cappé, O., Douc, R., Guillin, A., Marin, J.-M., & Robert, C. (2008). Adaptive importance sampling in general mixture classes. *Statistics and Computing*, *18*, 447–459.

Cappé, O., Guillin, A., Marin, J.-M., & Robert, C. (2004). Population Monte Carlo. *Journal of Computational and Graphical Statistics*, *13*, 907–929.

Cappé, O., & Robert, C. (2000). Ten years and still running! *Journal of American Statistical Association*, *95*, 1282–1286.

Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., … Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software*, *76*.

Carter, J., & White, D. (2013). History matching on the Imperial College fault model using parallel tempering. *Computational Geosciences*, *17*, 43–65.

Casella, G., & Robert, C. (1996). Rao-Blackwellization of sampling schemes. *Biometrika*, *83*, 81–94.

Chen, T., Fox, E., & Guestrin, C. (2014). Stochastic gradient Hamiltonian Monte Carlo. In Proceedings of the International Conference on Machine Learning, ICML'2014 (pp. 1683–1691).

Chen, T., & Hwang, C. (2013). Accelerating reversible Markov chains. *Statistics & Probability Letters*, *83*, 1956–1962.

Davis, M. H. (1984). Piecewise-deterministic Markov processes: A general class of non-diffusion stochastic models. *Journal of the Royal Statistical Society: Series B: Methodological*, 353–388.

Davis, M. H. (1993). *Markov models & optimization* (Vol. 49). CRC Press.

Del Moral, P., Doucet, A., & Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of Royal Statistical Society Series B*, *68*, 411–436.

Deligiannidis, G., Doucet, A., & Pitt, M. K. (2015). The correlated pseudo-marginal method. *arXiv preprint arXiv:1511.04992.*

Ding, N., Fang, Y., Babbush, R., Chen, C., Skeel, R. D., & Neven, H. (2014). Bayesian sampling using stochastic gradient thermostats. *Proceedings of the 27th International Conference on Neural Information Processing Systems* - Volume 2, NIPS 2015 (pp. 3203–3211).

Douc, R., Guillin, A., Marin, J.-M., & Robert, C. (2007). Convergence of adaptive mixtures of importance sampling schemes. *Annals of Statistics*, *35*(1), 420–448.

Douc, R., & Robert, C. (2010). A vanilla variance importance sampling via population Monte Carlo. *Annals of Statistics*, *39*(1), 261–277.

Doucet, A., Godsill, S., & Andrieu, C. (2000). On sequential Monte-Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, *10*, 197–208.

Duane, S., Kennedy, A. D., Pendleton, B. J., & Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, *195*, 216–222.

Durmus, A., & Moulines, E. (2017). Nonasymptotic convergence analysis for the unadjusted Langevin algorithm. *Annals of Applied Probability*, *27*, 1551–1587.

Earl, D. J., & Deem, M. W. (2005). Parallel tempering: Theory, applications, and new perspectives. *Physical Chemistry Chemical Physics*, *7*, 3910–3916.

Fielding, M., Nott, D. J., & Liong, S.-Y. (2011). Efficient MCMC schemes for computationally expensive posterior distributions. *Technometrics*, *53*, 16–28.

Gelfand, A., & Smith, A. (1990). Sampling based approaches to calculating marginal densities. *Journal of the American Statistical Association*, *85*, 398–409.

Gelman, A., Gilks, W., & Roberts, G. (1996). Efficient Metropolis jumping rules. In J. Berger, J. Bernardo, A. Dawid, D. Lindley, & A. Smith (Eds.), *Bayesian statistics 5* (pp. 599–608). Oxford, England: Oxford University Press.

Geyer, C. J. (1991). Markov chain Monte Carlo maximum likelihood. *Computing Science and Statistics*, *23*, 156–163.

Girolami, M., & Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society, Series B: Statistical Methodology*, *73*, 123–214.

Guihenneuc-Jouyaux, C., & Robert, C. P. (1998). Discretization of continuous Markov chains and Markov chain Monte Carlo convergence assessment. *Journal of the American Statistical Association*, *93*, 1055–1067.

Haario, H., Saksman, E., & Tamminen, J. (1999). Adaptive proposal distribution for random walk Metropolis algorithm. *Computational Statistics*, *14*(3), 375–395.

Hoffman, M. D., & Gelman, A. (2014). The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning and Research*, *15*, 1593–1623.

Hwang, C.-R., Hwang-Ma, S.-Y., & Sheu, S.-J. (1993). Accelerating gaussian diffusions. *The Annals of Applied Probability*, *3*, 897–913.

Iba, Y. (2000). Population-based Monte Carlo algorithms. *Transactions of the Japanese Society for Artificial Intelligence*, *16*, 279–286.

Jacob, P. E., O'Leary, J., & Atchadé, Y. F. (2017). Unbiased Markov chain Monte Carlo with couplings. *ArXiv e-prints*. 1708.03625.

Jacob, P., Robert, C. P., & Smith, M. H. (2011). Using parallel computation to improve independent Metropolis–Hastings based estimation. *Journal of Computational and Graphical Statistics*, *20*, 616–635.

Lehmann, E., & Casella, G. (1998). *Theory of point estimation* (revised ed.). New York, NY: Springer-Verlag.

Liang, F., Liu, C., & Carroll, R. (2007). Stochastic approximation in Monte Carlo computation. *Journal of the American Statistical Association*, *102*, 305–320.

Liu, J., Wong, W., & Kong, A. (1994). Covariance structure of the Gibbs sampler with application to the comparison of estimators and augmentation schemes. *Biometrika*, *81*, 27–40.

Liu, J., Wong, W., & Kong, A. (1995). Covariance structure and convergence rates of the Gibbs sampler with various scans. *Journal of Royal Statistical Society Series B*, *57*, 157–169.

Liu, J. S., Liang, F., & Wong, W. H. (2000). The multiple-try method and local optimization in Metropolis sampling. *Journal of the American Statistical Association*, *95*, 121–134.

Livingstone, S., Faulkner, M. F., & Roberts, G. O. (2017). Kinetic energy choice in Hamiltonian/hybrid Monte Carlo. *arXiv preprint arXiv:1706.02649*.

MacKay, D. J. C. (2002). *Information theory, inference & learning algorithms*. Cambridge, England: Cambridge University Press.

Marinari, E., & Parisi, G. (1992). Simulated tempering: A new Monte Carlo scheme. *EPL (Europhysics Letters)*, *19*, 451–458.

Martino, L., Elvira, V., Luengo, D., Corander, J., & Louzada, F. (2016). Orthogonal parallel MCMC methods for sampling and optimization. *Digital Signal Processing*, *58*, 64–84.

Martino, L. (2018). A Review of Multiple Try MCMC algorithms for Signal Processing. *ArXiv e-prints.* 1801.09065.

Mengersen, K., & Robert, C. (2003). Iid sampling with self-avoiding particle filters: The pinball sampler. In J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, & M. West (Eds.), *Bayesian statistics* (Vol. 7). Oxford, England: Oxford University Press.

Meyn, S., & Tweedie, R. (1993). *Markov chains and stochastic stability*. New York, NY: Springer-Verlag.

Miasojedow, B., Moulines, E., & Vihola, M. (2013). An adaptive parallel tempering algorithm. *Journal of Computational and Graphical Statistics*, *22*, 649–664.

Minsker, S., Srivastava, S., Lin, L., & Dunson, D. B. (2014). Scalable and robust Bayesian inference via the median posterior. In *Proceedings of the 31st International Conference on International Conference on Machine Learning* - Volume 32 (pp. 1656–1664). ICML'14, JMLR.org.

Mira, A. (2001). On Metropolis-Hastings algorithms with delayed rejection. *Metron*, *59*(3–4), 231–241.

Mira, A., & Sargent, D. J. (2003). A new strategy for speeding Markov chain Monte Carlo algorithms. *Statistical Methods and Applications*, *12*, 49–60.

Mohamed, L., Calderhead, B., Filippone, M., Christie, M., & Girolami, M. (2012). Population MCMC methods for history matching and uncertainty quantification. *Computational Geosciences*, *16*, 423–436.

Mykland, P., Tierney, L., & Yu, B. (1995). Regeneration in Markov chain samplers. *Journal of the American Statistical Association*, *90*, 233–241.

Neal, R. M. (1996). Sampling from multimodal distributions using tempered transitions. *Statistics and Computing*, *6*, 353–366.

Neal, R. (1999). *Bayesian learning for neural networks* (Vol. 118). New York, NY: Springer Verlag Lecture notes.

Neal, R. (2011). MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. L. Jones, & X.-L. Meng (Eds.), *Handbook of Markov Chain Monte Carlo* (pp. 113–162). New York, NY: CRC Press.

Neiswanger, W., Wang, C., & Xing, E. (2013). Asymptotically exact, embarrassingly parallel MCMC. *arXiv preprint arXiv:1311.4780.*

Quiroz, M., Villani, M., & Kohn, R. (2016). Exact subsampling MCMC. *arXiv preprint arXiv:1603.08232.*

Rasmussen, C. E. (2003). Gaussian processes to speed up hybrid Monte Carlo for expensive Bayesian integrals. In J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, & M. West (Eds.), *Bayesian Statistics* (Vol. 7, pp. 651–659). Oxford, England: Oxford University Press.

Rasmussen, C. E., & Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. Cambridge, MA: The MIT Press.

Rhee, C.-H., & Glynn, P. W. (2015). Unbiased estimation with square root convergence for sde models. *Operations Research*, *63*, 1026–1043.

Robert, C., & Casella, G. (2004). *Monte Carlo statistical methods* (2nd ed.). New York, NY: Springer-Verlag.

Robert, C., & Casella, G. (2009). *Introducing Monte Carlo methods with R*. New York: Springer-Verlag.

Roberts, G., Gelman, A., & Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, *7*, 110–120.

Roberts, G., & Rosenthal, J. (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, *16*, 351–367.

Roberts, G., & Rosenthal, J. (2007). Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms. *Journal of Applied Probability*, *44*(2), 458–475.

Roberts, G., & Rosenthal, J. (2009). Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics*, *18*, 349–367.

Roberts, G., & Rosenthal, J. (2014). Minimising MCMC variance via diffusion limits, with an application to simulated tempering. *The Annals of Applied Probability*, *24*, 131–149.

Rubinstein, R. Y. (1981). *Simulation and the Monte Carlo method*. New York, NY: John Wiley.

Saksman, E., & Vihola, M. (2010). On the ergodicity of the adaptive Metropolis algorithm on unbounded domains. *The Annals of Applied Probability*, *20*(6), 2178–2203.

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. *International Journal of Management Science and Engineering Management*, *11*, 78–88.

Srivastava, S., Cevher, V., Dinh, Q., & Dunson, D. (2015). WASP: Scalable Bayes via barycenters of subset posteriors. In G. Lebanon and S. V. N. Vishwanathan (Eds.), *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, (pp. 912–920) vol. 38 of Proceedings of Machine Learning Research. PMLR, San Diego, California, USA.

Storvik, G. (2002). Particle filters for state space models with the presence of static parameters. *IEEE Transactions on Signal Processing*, *50*, 281–289.

Sun, Y., Gomez, F., & Schmidhuber, J. (2010). Improving the asymptotic performance of Markov chain Monte Carlo by inserting vortices. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems* - Volume 2 (pp. 2235–2243). NIPS'10, Curran Associates Inc., USA.

Terenin, A., Simpson, D., & Draper, D. (2015). Asynchronous Gibbs Sampling. *ArXiv e-prints*. 1509.08999.

Tierney, L., & Mira, A. (1998). Some adaptive Monte Carlo methods for Bayesian inference. *Statistics in Medicine*, *18*, 2507–2515.

Tjelmeland, H. (2004). *Using all Metropolis-Hastings proposals to estimate mean values*. (Technical Report 4). Norwegian University of Science and Technology, Trondheim, Norway.

Wang, F., & Landau, D. (2001). Determining the density of states for classical statistical models: A random walk algorithm to produce a flat histogram. *Physical Review E*, *64*, 056101.

Wang, X. & Dunson, D. (2013). Parallelizing MCMC via weierstrass sampler. *arXiv preprint arXiv:1312.4605.*

Wang, X., Guo, F., Heller, K., & Dunson, D. (2015). Parallelizing MCMC with random partition trees. *Advances in Neural Information Processing Systems*, 451–459.

Welling, M. & Teh, Y. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*. ICML'11 (pp. 681–688), USA: Omnipress.

Woodard, D. B., Schmidler, S. C., & Huber, M. (2009a). Conditions for rapid mixing of parallel and simulated tempering on multimodal distributions. *The Annals of Applied Probability*, *19*, 617–640.

Woodard, D. B., Schmidler, S. C., & Huber, M. (2009b). Sufficient conditions for torpid mixing of parallel and simulated tempering. *Electronic Journal of Probability*, *14*, 780–804.

Xie, Y., Zhou, J., & Jiang, S. (2010). Parallel tempering Monte Carlo simulations of lysozyme orientation on charged surfaces. *The Journal of Chemical Physics*, *132*. 02B602.

How to cite this article: Robert CP, Elvira V, Tawn N, Wu C. Accelerating MCMC algorithms. *WIREs Comput Stat*. 2018;10:e1435. https://doi.org/10.1002/wics.1435