Bayesian Analysis (2022) **17**, Number 1, pp. 193–221

Finding our Way in the Dark: Approximate MCMC for Approximate Bayesian Methods[*]

Evgeny Levi[†] and Radu V. Craiu[‡,§]

Abstract. With larger data at their disposal, scientists are emboldened to tackle complex questions that require sophisticated statistical models. It is not unusual for the latter to have likelihood functions that elude analytical formulations. Even under such adversity, when one can simulate from the sampling distribution, Bayesian analysis can be conducted using approximate methods such as Approximate Bayesian Computation (ABC) or Bayesian Synthetic Likelihood (BSL). A significant drawback of these methods is that the number of required simulations can be prohibitively large, thus severely limiting their scope. In this paper we design perturbed MCMC samplers that can be used within the ABC and BSL paradigms to significantly accelerate computation while maintaining control on computational efficiency. The proposed strategy relies on recycling samples from the chain's past. The algorithmic design is supported by a theoretical analysis while practical performance is examined via a series of simulation examples and data analyses.

MSC2020 subject classifications: Primary 62-08, 60K35; secondary 60J22.

Keywords: Approximate Bayesian Computation, Synthetic Likelihood, Perturbed MCMC, k-Nearest Neighbour.

1 Introduction

Since the early 1990s Bayesian statisticians have been able to operate largely due to the rapid development of Markov chain Monte Carlo (MCMC) sampling methods (see, for example Craiu and Rosenthal, 2014, for a recent review). Given observed data $\mathbf{y}_0 \in \mathcal{X}^n$ with sampling density $f(\mathbf{y}_0|\theta)$ indexed by parameter $\theta \in \mathbf{R}^q$, Bayesian inference for functions of θ relies on the characteristics of the posterior distribution

$$\pi(\theta|\mathbf{y}_0) = \frac{p(\theta)f(\mathbf{y}_0|\theta)}{\int_{R^q} p(\theta)f(\mathbf{y}_0|\theta)d\theta} \propto p(\theta)f(\mathbf{y}_0|\theta), \tag{1.1}$$

where $p(\theta)$ denotes the prior distribution. When the posterior (1.1) cannot be studied analytically, we rely on MCMC algorithms to generate samples from π. While traditional

[*] We thank the Editor, Michele Guindani, as well as Brunero Liseo, Jeffrey Rosenthal, Stanislav Volgushev, an Associate Editor and two referees for constructive suggestions that have improved the paper. Funding support for the work was provided by NSERC of Canada and the Canadian Statistical Sciences Institute (CANSSI).

[†] Department of Statistical Sciences, University of Toronto, 700 University Ave, Toronto, M5G 1Z5, Canada, evgeny.levi@utoronto.ca

[‡] Department of Statistical Sciences, University of Toronto, 700 University Ave, Toronto, M5G 1Z5, Canada, radu.craiu@utoronto.ca

[§] Thanks the organizers of the BIRS workshop "Validating and Expanding Approximate Bayesian Computation Methods" for creating a stimulating environment that has planted the seed for this work.

https://doi.org/10.1214/20-BA1250

MCMC samplers such as Metropolis-Hastings or Hamiltonian MCMC (see Brooks et al., 2011, and references therein) can sample distributions with unknown normalizing constants, they rely on the closed form of the unnormalized posterior, $p(\theta)f(\mathbf{y}_0|\theta)$.

The framework we just described has been altered in multiple ways by the advent of large data. First, larger data tend to yield likelihood functions that are much more expensive to compute, thus exposing the liability inherent in the iterative nature of MCMC samplers. In response to this challenge, new computational methods based on divide and conquer (Scott et al., 2016; Wang and Dunson, 2013; Entezari et al., 2018), subsampling (Bardenet et al., 2014; Quiroz et al., 2015), pre-computation (Boland et al., 2018), or sequential (Balakrishnan et al., 2006; Maclaurin and Adams, 2015) strategies have emerged. Second, it is understood that larger data should yield answers to more complex problems. This implies the use of increasingly complex models, in as much as the sampling distribution is no longer available in closed form.

In the absence of a tractable likelihood function, statisticians have developed approximate methods to perform Bayesian inference when, for any parameter value $\theta \in \mathbf{R}^q$, data $\mathbf{y} \sim f(\mathbf{y}|\theta)$ can be sampled from the model. Here we consider two alternative approaches that have been proposed and gained considerable momentum in recent years: the Approximate Bayesian Computation (ABC) (Marin et al., 2012; Baragatti and Pudlo, 2014; Sisson et al., 2018a; Drovandi, 2018) and the Bayesian Synthetic Likelihood (BSL) (Wood, 2010; Drovandi et al., 2018a; Price et al., 2018). Both algorithms are effective when they are combined with Markov chain Monte Carlo sampling schemes to produce samples from an approximation of π and both share the need for generating many pseudo-data sets $\mathbf{y} \sim f(\mathbf{y}|\theta)$. This comes with serious challenges when the data is large and generating a pseudo-data set is computationally expensive. In this paper we tackle the reduction of computational burden by recycling draws from the chain's history. The latter is achieved by reusing the quantities central to running the MCMC-ABC and MCMC-BSL chains, i.e. the parameter values and the corresponding pseudo-data discrepancies (for ABC) and summary statistics (for BSL). The information in past variates is incorporated using a k-Nearest-Neighbour (kNN) approach to estimate the transition kernel. The idea of using the past draws and simulations to accelerate the likelihood-free methods was addressed previously – the references to the appropriate literature are presented in Sections 2 and 4. The alternative approaches use Gaussian Processes (GP) to borrow information from past realizations of the chain that are close to the proposed next state. However, unlike kNN, the GP model does not guarantee that the estimated likelihood would approach the true unknown likelihood. Moreover, the kNN approach is easier to implement and computationally faster than the GP since the latter requires inversion of large-dimensional matrices and crucially depends on the choice of the covariance function. We demonstrate that we can control the approximating error introduced when perturbing the original kernel using some of the error analysis for perturbed Markov chains developed recently by Mitrophanov (2005), Johndrow et al. (2015b) and Johndrow and Mattingly (2017).

The paper is structured as follows. Section 2 briefly reviews the ABC method and Section 3 introduces the proposed MCMC algorithms for ABC. Section 4 reviews BSL sampling and extends the proposed methods to this class of approximations. The practical impact of these algorithms is evaluated via simulations in Section 5 and data analyses in the Supplementary Material (Levi and Craiu, 2020). The theoretical analysis showing control of perturbation errors in total variation norm is in Section 6. The paper closes with ideas for future work and conclusions.

2 Approximate Bayesian Computation

In order to illustrate the ABC sampler, let us consider the following Hidden Markov Model (HMM)

$$X_0 \sim \quad p(x_0),\ X_i|x_{i-1} \sim \quad p(X_i|x_{i-1},\theta),\ Y_i|x_i \sim \quad p(Y_i|x_i,\theta), \quad i=1,\ldots,n. \tag{2.1}$$

Unless Gaussian distributions are used to specify the transition and emission laws given in (2.1), the marginal distribution $P(y_1,\cdots,y_n|\theta)$ cannot be calculated in closed form. It is possible to treat the hidden random variables X_i as auxiliary and sample them using Particle MCMC (PMCMC) (Andrieu et al., 2010) or ensemble MCMC (Shestopaloff and Neal, 2013). However, computations become increasingly difficult as n increases. For some financial time series models such as Stochastic Volatility for log return, the α-Stable distribution may be useful to model transition and/or emission probabilities (Nolan, 2003). However, the stable distributions do not have closed form densities, thus rendering the particle and ensemble MCMC impossible to use. Other widely used examples where the likelihood functions cannot be expressed analytically include various networks models (e.g., Kolaczyk and Csárdi, 2014) and Markov random fields (Rue and Held, 2005). For such models with intractable or computationally expensive likelihood evaluations, simulation based algorithms such as ABC are frequently used for inference. In its simplest form, the ABC is an accept/reject sampler. Given a user-defined summary statistic $S(\mathbf{y}) \in \mathbf{R}^p$, the Accept/Reject ABC is described in Algorithm 1.

Algorithm 1 Accept/Reject ABC.

```
Given observed $\mathbf{y}_0$ and required number of samples $M$.
for $t = 1,\cdots,M$ do
    Match = FALSE
    while Not Match do
        $\zeta^* \sim p(\zeta)$ and $\mathbf{y} \sim f(\mathbf{y}|\zeta^*)$
        if $S(\mathbf{y}) = S(\mathbf{y}_0)$ then
            $\theta^{(t)} = \zeta^*$,
            Match = TRUE.
        end if
    end while
end for
```

We emphasize that a closed form equation for the likelihood is not needed, only the ability to generate from $f(\mathbf{y}|\theta)$ for any θ. If $S(\mathbf{y})$ is a sufficient statistic and $\Pr(S(\mathbf{y}) = s_0) > 0$ then the algorithm yields posterior samples from the true posterior $\pi(\theta|\mathbf{y}_0)$, where we defined $s_0 = S(\mathbf{y}_0)$. Alas, the level of complexity for models where ABC is needed, makes it unlikely for these two conditions to hold. In order to implement ABC under more realistic assumptions, a (small) constant ϵ is chosen and ζ^* is accepted whenever $d(S(\mathbf{y}), s_0) < \epsilon$, where $d(S(\mathbf{y}), s_0)$ is a user-defined distance function. The introduction of $\epsilon > 0$ and the use of non-sufficient statistics remove layers of exactness from the target distribution. The approximating distribution is $\pi_\epsilon(\theta|s_0)$ and we have

$$\lim_{\epsilon \downarrow 0} \pi_\epsilon(\theta|s_0) = \pi(\theta|s_0). \tag{2.2}$$

In light of (2.2) one would like to have $S(\mathbf{y}) = \mathbf{y}$, but if the sample size of $\mathbf{y}_0$ is large, then the curse of dimensionality leads to $\Pr(d(\mathbf{y}, \mathbf{y}_0) < \epsilon) \approx 0$. Consequently, obtaining even a moderate number of samples using ABC becomes an unattainable goal. In almost all cases of interest, S is not a sufficient statistic, implying that some information about θ is lost. Not surprisingly, much attention has been given to finding appropriate low-dimensional summary statistics for inference (see, for example Robert et al., 2011; Fearnhead and Prangle, 2012; Marin et al., 2014; Prangle, 2015). In this paper we assume that the summary statistic $S(\mathbf{y})$ is given.

In the absence of information about the model parameters, the prior and posterior distributions may be misaligned, having non-overlapping regions of mass concentration. Hence, parameter values that are drawn from the prior will be rarely retained making the algorithm very inefficient. Algorithm 2 presents the ABC-MCMC algorithm of Marjoram et al. (2003) which avoids sampling from the prior and instead relies on building a chain with a Metropolis-Hastings (MH) transition kernel, with state space $\{(\theta, \mathbf{y}) \in \mathbf{R}^q \times \mathcal{X}^n\}$, proposal distribution $q(\zeta|\theta) \times f(\mathbf{y}|\zeta)$ and target distribution

$$\pi_\epsilon(\theta, \mathbf{y}|\mathbf{y}_0) \propto p(\theta) f(\mathbf{y}|\theta) \mathbf{1}_{\{\delta(\mathbf{y}_0, \mathbf{y}) < \epsilon\}}, \tag{2.3}$$

where $\delta(\mathbf{y}_0, \mathbf{y}) = d(S(\mathbf{y}), s_0)$. Note that the goal is the marginal distribution for θ which is:

$$\pi_\epsilon(\theta|\mathbf{y}_0) = \int \pi_\epsilon(\theta, \mathbf{y}|\mathbf{y}_0) d\mathbf{y} \propto \int p(\theta) f(\mathbf{y}|\theta) \mathbf{1}_{\{\delta(\mathbf{y}_0, \mathbf{y}) < \epsilon\}} d\mathbf{y} = p(\theta) \Pr(\delta(\mathbf{y}_0, \mathbf{y}) < \epsilon|\theta). \tag{2.4}$$

Algorithm 2 ABC MCMC.

```
Given $\mathbf{y}_0$, $s_0$, $\epsilon > 0$ and required number of samples $M$.
Find initial $\theta^{(0)}$ and $\mathbf{y} \sim f(\mathbf{y}|\theta^{(0)})$ such that $d(S(\mathbf{y}), s_0) < \epsilon$.
for $t = 1, \cdots, M$ do
    Generate $\zeta^* \sim q(\cdot|\theta^{(t-1)})$.
    Simulate $\mathbf{y}^* \sim f(\mathbf{y}|\zeta^*)$ and let $\delta(\mathbf{y}^*, \mathbf{y}_0) = d(S(\mathbf{y}^*), s_0)$.
    Calculate $\alpha = \min\left\{1, \mathbf{1}_{\{\delta(\mathbf{y}^*, \mathbf{y}_0) < \epsilon\}} \frac{p(\zeta^*) q(\theta^{(t-1)}|\zeta^*)}{p(\theta^{(t-1)}) q(\zeta^*|\theta^{(t-1)})}\right\}$
    Set $\theta^{(t)} = \zeta^*$ with probability $\alpha$, and $\theta^{(t)} = \theta^{(t-1)}$ otherwise.
end for
```

There are alternatives to Algorithm 2. For instance, Lee et al. (2012) approximates $P(\delta(\mathbf{y}_0, \mathbf{y}) < \epsilon|\theta)$ via one of its unbiased estimators, $J^{-1} \sum_{j=1}^{J} 1_{\{\delta(\mathbf{y}_0, \mathbf{y}_j) < \epsilon\}}$ where $J \geq 1$ and each $\mathbf{y}_j$ is simulated from $f(\mathbf{y}|\theta)$. The use of unbiased estimators for $P(\delta(\mathbf{y}_0, \mathbf{y}) <$

$\epsilon|\theta)$ when computing the MH acceptance ratio can be validated using the theory of pseudo-marginal MCMC samplers (see the seminal paper of Andrieu and Roberts, 2009). Clearly, when the probability $P(\delta < \epsilon|\theta)$ is very small, this method would require simulating a large number of δs (or, equivalently, $\mathbf{y}$'s) in order to move to a new state. Other MCMC designs suitable for ABC can be found in Bornn et al. (2014).

Sequential Monte Carlo (SMC) samplers have also been successfully used for ABC (henceforth denoted ABC-SMC) (Sisson et al., 2007; Lee, 2012; Filippi et al., 2013). ABC-SMC requires a specified decreasing sequence of tolerances, $\epsilon_0 > \cdots > \epsilon_J$. The method of Lee (2012) uses the Particle MCMC design (Andrieu et al., 2010) in which samples are updated as the target distribution evolves with ϵ. More specifically, it starts by sampling $\theta_0^{(1)}, \ldots, \theta_0^{(M)}$ from $\pi_{\epsilon_0}(\theta|\mathbf{y}_0)$ using Accept-Reject ABC. Subsequently, at time $t+1$ all samples are sequentially updated so their distribution is $\pi_{\epsilon_{t+1}}(\theta|\mathbf{y}_0)$ (see Lee, 2012, for a complete description). The advantage of this method is not only that it starts from large ϵ, but also that it generates independent draws. A comprehensive coverage of computational techniques for ABC can be found in Sisson et al. (2018b) and references therein. We also note a general lack of guidelines concerning the selection of ϵ, which is unfortunate as the performance of ABC sampling depends heavily on its value. To make a fair comparison between different methods, we revise ABC-MCMC algorithm by introducing a decreasing sequence $\epsilon_0 > \cdots > \epsilon_J$ (J is number of "steps") similar to ABC-SMC and "learning" transition kernel during the burn-in as in Algorithm 3. Since the choice of proposal distribution $q(\cdot|\theta)$ can considerably influence the performance of the ABC-MCMC sampler, we consider finite adaptation during the burn-in period of length B. In addition, during the burn-in the ϵ also varies, starting with a higher value (which makes it easier to find the initial $\theta^{(0)}$ value) and gradually decreasing in accordance to a pre-determined scheme. In our implementations we use independent MH sampling or RWM. In the former case, the proposal is Gaussian $\mathcal{N}(\cdot|\tilde{\mu}, \tilde{\Sigma})$ with

Algorithm 3 ABC MCMC modified (ABC-MCMC-M).

```
Given $\mathbf{y}_0$, $s_0$, sequence $\epsilon_0 > \cdots > \epsilon_J$, constant $c$, burn-in period $B$ and required number of samples $M$.
Define $\epsilon = \epsilon_0$ and find initial $\theta^{(0)}$ with simulated $\mathbf{y}$ such that $d(S(\mathbf{y}), s_0) < \epsilon$.
Let $\tilde{\mu} = \mathrm{E}_p(\theta)$ and $\tilde{\Sigma} = c\Sigma$ where $\Sigma = \mathrm{Var}_p(\theta)$.
Define, $b = \lfloor (B/J) \rfloor$ and define sequence $(a_1, \cdots, a_J) = (b, 2b, \cdots, Jb)$.
for $t = 1, \cdots, M$ do
    if $t = a_j$ for some $j = 1, \cdots, J$ then
        Set $\epsilon = \epsilon_j$ and set $\tilde{\mu}$ as mean of $\{\theta^{(t)}\}$ $t = 1, \cdots, (a_j - 1)$ and $\tilde{\Sigma} = c\Sigma$ where $\Sigma$ is covariance of $\{\theta^{(t)}\}$ $t = 1, \cdots, (a_j - 1)$.
    end if
    Generate $\zeta^* \sim q(\cdot|\theta^{(t-1)}, \tilde{\mu}, \tilde{\Sigma})$.
    Simulate $\mathbf{y}^* \sim f(\mathbf{y}|\zeta^*)$ and let $\delta^* = d(S(\mathbf{y}^*), s_0)$.
    Calculate $\alpha = \min\left\{1, \mathbf{1}_{\{\delta^* < \epsilon\}} \frac{p(\zeta^*) q(\theta^{(t-1)}|\zeta^*, \tilde{\mu}, \tilde{\Sigma})}{p(\theta^{(t-1)}) q(\zeta^*|\theta^{(t-1)}, \tilde{\mu}, \tilde{\Sigma})}\right\}$
    Set $\theta^{(t)} = \zeta^*$ with probability $\alpha$ and $\theta^{(t)} = \theta^{(t-1)}$ otherwise.
end for
```

$c = 3$. The RWM proposal is $\mathcal{N}(\cdot|\theta^{(t-1)}, \tilde{\Sigma})$ with $c = 2.38^2/q$ (following Roberts et al., 1997; Roberts and Rosenthal, 2001). All the algorithms discussed so far rely on numerous generations of pseudo-data. Researchers have recognized that the latter can be computationally costly to produce, so proposals for reducing the simulation cost have been made by Wilkinson (2014), Drovandi et al. (2018b), Järvenpää et al. (2018) and Sherlock et al. (2017), among others. Järvenpää et al. (2018) utilized the simulated pairs (ζ, δ) to estimate the conditional distribution of $\delta|\zeta$ using a Gaussian Process (GP) approach which allowed, for a new proposal ζ^*, a fast calculation of $P(\delta < \epsilon|\zeta^*)$. Instead of estimating the conditional distribution, Wilkinson (2014) uses a GP approach to link ζ and $\log \hat{P}(\delta < \epsilon)$, hence to approximate the log likelihood for any ζ^*. Similar ideas can also be used for Bayesian Synthetic Likelihood sampling methods. For instance, in Drovandi et al. (2018b) the GP model is used to estimate the true likelihood or its unbiased estimate. The implementation of GP may not be universally appropriate as it assumes that the distribution of $\log \hat{P}(\delta < \epsilon)$ is normal and has constant variance for all ζ which is clearly not true when the number of pseudo-data simulations (for each ζ) is small. Also the final likelihood estimates (from GP) are generally no longer unbiased and thus theoretical justifications for the proposed methods are necessary. The proposed idea in this paper is related to the work of Sherlock et al. (2017), where authors reduce the computational burden by utilizing a delayed-acceptance MCMC. The delayed-acceptance methods involve two acceptance ratios at each MCMC iteration. In the first one the likelihood is estimated using an estimator that involves a small computational cost (the authors use kNN) and, conditional on the proposal being accepted in stage one, the second acceptance ratio is calculated with the computationally expensive likelihood. The advantage of this method is that there is no need to compute the expensive likelihood if the proposal is not accepted in the first stage. One obvious limitation is that the expensive log-likelihood still must be calculated each time the chain moves.

To accelerate ABC-MCMC we consider a different approach and propose to store and utilize past simulations (with appropriate weights) in order to speed up the calculation while keeping under control the resulting approximating errors. The objective is to approximate $P(\delta < \epsilon|\zeta^*)$ for any ζ^* at every MCMC iteration using past simulated (ζ, δ) proposals, making the whole procedure computationally faster. The changes proposed perturb the chain's transition kernel and we rely on the theory developed by Mitrophanov (2005) and Johndrow et al. (2015a) to assess the approximating error for the posterior. The k-Nearest-Neighbor (kNN) method is used to integrate past observations into the transition kernel. For a large enough chain history, we can control the error between the intended stationary distribution and that of the proposed accelerated MCMC as shown in Section 6.

3 Approximated ABC-MCMC (AABC-MCMC)

In this section we describe an ABC-MCMC algorithm that utilizes past simulations to significantly improve computational efficiency. As noted previously, the ABC-MCMC with threshold ϵ targets the density

$$\pi_\epsilon(\theta|\mathbf{y}_0) \propto p(\theta)P(\delta(\mathbf{y}_0, \mathbf{y}) < \epsilon|\theta), \tag{3.1}$$

where $\delta(\mathbf{y}_0, \mathbf{y}) = d(S(\mathbf{y}), s_0)$ with $\mathbf{y} \sim f(\mathbf{y}|\theta)$ and $\theta \in \Theta$. Denote $h(\theta) := P(\delta(\mathbf{y}_0, \mathbf{y}) < \epsilon|\theta)$ and note that if h were known for every θ then we could run an MH-MCMC chain with the target proportional to $p(\theta)h(\theta)$. Alas, h is usually unknown and unbiased estimates can be computationally expensive or statistically inefficient. We build an alternative approach that hinges on consistent estimates of h. The latter use the chain's past history, are much cheaper to compute, and require a new theoretical treatment.

To fix ideas, suppose that at time t we set the proposal $(\zeta_{t+1}, \mathbf{w}_{t+1}) \sim q(\zeta|\theta^{(t)})f(\mathbf{w}|\zeta)$ and suppose that at iteration N, all the proposals ζ_n, regardless whether they were accepted or rejected, along with corresponding distances $\delta_n = \delta(\mathbf{w}_n, \mathbf{y}_0)$ are available for $0 \le n \le N-1$. This past history is stored in the set $\mathcal{Z}_{N-1} = \{\zeta_n, \delta_n\}_{n=1}^{N-1}$. Given a new proposal $\zeta^* \sim q(|\theta^{(t)})$, we generate $\mathbf{w}^* \sim f(\cdot|\zeta^*)$ and compute $\delta^* = d(S(\mathbf{w}^*), s_0)$. Set $\zeta_N = \zeta^*$, $\mathbf{w}_N = \mathbf{w}^*$, $\mathcal{Z}_N = \mathcal{Z}_{N-1} \cup \{(\zeta_N, \delta_N)\}$ and estimate $h(\zeta^*)$ using

$$\hat{h}(\zeta^*) = \frac{\sum_{n=1}^{N} W_{Nn}(\zeta^*)\mathbf{1}_{\{\delta_n < \epsilon\}}}{\sum_{n=1}^{N} W_{Nn}(\zeta^*)}, \tag{3.2}$$

where $W_{Nn}(\zeta^*) = W(\|\zeta_n - \zeta^*\|)$ are weights and $W : \mathbf{R} \to [0, \infty)$ is a decreasing function. We discuss a couple of choices for the function $W(\cdot)$ below.

Remark 1. Note that the Markovian property of the chain is violated since the acceptance probability does not depend solely on the current state, but also on the past ones. We defer the theoretical considerations for dealing with adaptation in the context of perturbed Markov chains to a future communication. Below, we modify slightly the construction above while respecting the core idea.

In order to separate the samples used as proposals from those used to estimate h in (3.2), we will generate at each time t two independent samples $\zeta_{t+1} \sim q(\zeta|\theta^{(t)})$ and $(\tilde{\zeta}_{t+1}, \tilde{\mathbf{w}}_{t+1})$ from $q(\zeta|\theta^{(t)})f(\mathbf{w}|\zeta)$. Then, the history $\mathcal{Z}$ collects the $(\tilde{\zeta}, \tilde{\delta})$ samples while the proposal used to update the chain is the ζ sample. With this notation (3.2) becomes

$$\hat{h}(\zeta^*) = \frac{\sum_{n=1}^{N} W_{Nn}(\zeta^*)\mathbf{1}_{\{\tilde{\delta}_n < \epsilon\}}}{\sum_{n=1}^{N} W_{Nn}(\zeta^*)}, \tag{3.3}$$

where $\tilde{\delta}_n = \delta(\tilde{\mathbf{w}}, \mathbf{y}_0)$ and $W_{Nn}(\zeta^*) = W(\|\tilde{\zeta}_n - \zeta^*\|)$.

Remark 2. Even when δ^* is greater than ϵ, if there is a close neighbour of ζ^* whose corresponding δ is less than ϵ, then the estimated $h(\zeta^*)$ is not zero and the chain may move to a different state.

Remark 3. The consistent estimator $\hat{h}(\zeta^*) = \frac{\sum_{n=1}^{N} W_{Nn}(\zeta^*)\mathbf{1}_{\{\tilde{\delta}_n < \epsilon\}}}{\sum_{n=1}^{N} W_{Nn}(\zeta^*)}$ is expected to outperform the unbiased estimator $\tilde{h}(\zeta^*) = \frac{1}{K}\sum_{j=1}^{K} \mathbf{1}_{\{\tilde{\delta}_j < \epsilon\}}$, for both small and large K. For small K, the variability in the acceptance probabilities will be reduced, while for larger K the computational costs will be much smaller without sacrificing much in terms of precision. Since the proposed weighted estimate is no longer an unbiased estimator of $h(\theta)$, a new theoretical evaluation is needed to study the effect of perturbing the transition kernel on the statistical analysis. Central to the algorithm's utility is the

Algorithm 4 Approximated ABC MCMC (AABC-MCMC).

```
Given $\mathbf{y}_0$ with summary statistics $s_0$, sequence $\epsilon_0 > \cdots > \epsilon_J$, constant $c$, burn-in period $B$, required number of samples $M$, initial simulations $\mathcal{Z}_N = \{\tilde{\zeta}_n, \tilde{\delta}_n\}_{n=1}^N$ with $\tilde{\zeta}_n \sim p(\zeta)$, $\tilde{\mathbf{w}}_n \sim f(\cdot|\tilde{\zeta}_n)$ and $\tilde{\delta}_n = d(S(\tilde{\mathbf{w}}_n), s_0)$.
Define $\epsilon = \epsilon_0$ and set initial $\theta^{(0)}$ with simulated $\mathbf{y}$ such that $d(S(\mathbf{y}), s_0) < \epsilon$.
Let $\tilde{\mu} = \mathrm{E}_p[\theta]$, $\tilde{\Sigma} = c\Sigma$ where $\Sigma = \mathrm{Var}_p(\theta)$.
Define, $b = \lfloor (B/J) \rfloor$ and define sequence $(a_1, \cdots, a_J) = (b, 2b, \cdots, Jb)$
for $t = 1, \cdots, M$ do
    if $t = a_j$ for some $j = 1, \cdots, J$ then
        Set $\epsilon = \epsilon_j$ and set $\tilde{\mu}$ as mean of $\theta^{(t)}$ $t = 1, \cdots, (a_j - 1)$ and $\tilde{\Sigma} = c\Sigma$ where $\Sigma$ is covariance of $\theta^{(t)}$ $t = 1, \cdots, (a_j - 1)$.
    end if
    Generate $\zeta^* \sim \mathcal{N}(\cdot; \tilde{\mu}, \tilde{\Sigma})$ and $\tilde{\zeta}^* \sim \mathcal{N}(\cdot; \tilde{\mu}, \tilde{\Sigma})$.
    Simulate $\tilde{\mathbf{w}}^* \sim f(\cdot|\tilde{\zeta}^*)$ and let $\tilde{\delta}^* = d(S(\tilde{\mathbf{w}}^*), s_0)$.
    Add the dual simulated pair of parameter and discrepancy to the past set: $\mathcal{Z}_N = \mathcal{Z}_{N-1} \cup \{\tilde{\zeta}^*, \tilde{\delta}^*\}$ and set $N = N + 1$.
    Set $\hat{h}(\zeta^*) = \frac{\sum_{n=1}^N W_{Nn}(\zeta^*) \mathbf{1}_{\{\tilde{\delta}_n < \epsilon\}}}{\sum_{n=1}^N W_{Nn}(\zeta^*)}$ and $\hat{h}(\theta^{(t)}) = \frac{\sum_{n=1}^N W_{Nn}(\theta^{(t)}) \mathbf{1}_{\{\tilde{\delta}_n < \epsilon\}}}{\sum_{n=1}^N W_{Nn}(\theta^{(t)})}$.
    Calculate $\alpha = \min\left\{1, \frac{p(\zeta^*)\hat{h}(\zeta^*)\mathcal{N}(\theta^{(t)}; \tilde{\mu}, \tilde{\Sigma})}{p(\theta^{(t)})\hat{h}(\theta^{(t)})\mathcal{N}(\zeta^*; \tilde{\mu}, \tilde{\Sigma})}\right\}$
    Set $\theta^{(t+1)} = \zeta^*$ with probability $\alpha$ and $\theta^{(t+1)} = \theta^{(t)}$ otherwise..
end for
```

ability to control the total variation distance between the desired distribution of interest given in (3.1) and the modified chain's target. As will be shown in Section 6, we rely on three assumptions to ensure that the chain would approximately sample from (3.1): 1) compactness of Θ; 2) uniform ergodicity of the chain using the true h and 3) uniform convergence in probability of $\hat{h}(\theta)$ to $h(\theta)$ as $N \to \infty$.

The k-Nearest-Neighbor (kNN) regression approach (Fix and Hodges, 1951; Biau and Devroye, 2015) has a property of uniform consistency (Cheng, 1984). Define $K = g(N)$ (in our numerical experiments we have used $g(\cdot) = \sqrt{(\cdot)}$). Without loss of generality we relabel the elements of $\mathcal{Z}_N = \{\tilde{\zeta}_n, \tilde{\delta}_n\}_{n=1}^N$ according to distance $\|\tilde{\zeta}_n - \zeta^*\|$ so that $(\tilde{\zeta}_1, \tilde{\delta}_1)$ and $(\tilde{\zeta}_N, \tilde{\delta}_N)$ corresponds to the smallest and largest among all distances $\{\|\tilde{\zeta}_j - \zeta^*\| : 1 \leq j \leq N\}$, respectively. The kNN method sets $W_{Nn}(\zeta^*)$ to zero for all $n > K$. For $n \leq K$, we focus on the following two weighting schemes:

(U) The *uniform* kNN with $W_{Nn}(\zeta^*) = 1$ for all $n \leq K$;

(L) The *linear* kNN with $W_{Nn}(\zeta^*) = W(\|\tilde{\zeta}_n - \zeta^*\|) = 1 - \|\tilde{\zeta}_n - \zeta^*\| / \|\tilde{\zeta}_K - \zeta^*\|$ for $n \leq K$ so that the weight decreases from 1 to 0 as n increases from 1 to K.

The kNN's theoretical properties that are used to validate our sampler rely on the independence between the pairs $\{\tilde{\zeta}_n, \tilde{\delta}_n\}_{n \geq 1}$. Therefore, throughout the paper, we use an independent proposal in the MH sampler, i.e. $q(\cdot|\theta^{(t)}) = q(\cdot)$ and q is Gaussian. The entire procedure is outlined in Algorithm 4.

Therefore, at the end of a simulation of size M the MCMC samples are $\{\theta^{(1)},\ldots,\theta^{(M)}\}$ and the history used for updating the chain is $\{(\tilde{\zeta}_1,\tilde{\delta}_1),\ldots,(\tilde{\zeta}_M,\tilde{\delta}_M)\}$. Note that for any $N>0$, the elements in $\mathcal{Z}_N$ are independent of the chain's history up to time N. Therefore, the transition kernel of the chain depends only on the current state so it is Markovian and non-adaptive. Note also that $\hat{h}(\theta^{(t)})$ is required in order to determine the acceptance probability at step $t+1$. In this case the h-value may be updated if $\|\theta^{(t)}-\tilde{\zeta}^*\|$ is small enough. We could use the pseudo marginal MCMC method (Andrieu and Roberts, 2009; Andrieu and Vihola, 2015) to update only the numerator at each iteration, but this can result in slow mixing (see Drovandi et al., 2018b) since one large, "lucky" likelihood estimate can hinder the chain's prospect of moving on to a new state. However, updating both log-likelihoods at each iteration can significantly improve the mixing (see, for instance, Beaumont, 2003), but may change significantly the target distribution. We adopt an approach in which both estimates are updated. The computational cost of updating the denominator is very small in this case and the procedure is theoretically sound (see Section 6). The algorithm proposed here implements "naive" kNN which can be computationally burdensome as the chain progresses and the volume of historical data increases. Generally it requires $O(1)$ operations to add a new value to the history set and $O(M)$ operations to compute the distances between the proposal and all the past draws. Clearly this is inefficient for very large M. To reduce the number of operations, one can implement a more efficient KD-tree approach (Bentley, 1975; Friedman et al., 1977), in which past samples are stored in a multidimensional binary array. Along with the work of Sherlock et al. (2017) who proposed an adaptive variation of KD-tree, the required number of operations is $O(q\log(M))$ in order to add a new point or to search for the nearest neighbours. This can considerably speed up the proposed algorithms, but we do not pursue it in this paper.

In the next section we extend the approximate MCMC construction to Bayesian Synthetic Likelihood. In Section 5 of the Supplementary Material (Levi and Craiu, 2020) we use simulations to show that the proposed procedure generally improves the mixing of a chain.

4 BSL and Approximated BSL (ABSL)

An alternative way to bypass the intractability of the sampling distribution is proposed by Wood (2010). His indirect inference approach assumes that the conditional distribution for a user-defined statistic $S(\mathbf{y})$ given θ is Gaussian with mean μ_θ and covariance matrix Σ_θ. The *Synthetic Likelihood* (SL) procedure assigns to each θ the likelihood $SL(\theta)=\mathcal{N}(s_0;\mu_\theta,\Sigma_\theta)$, where as before $s_0=S(\mathbf{y}_0)$ and $\mathcal{N}(x;\mu,\Sigma)$ denotes the density of a normal with mean μ and covariance Σ. SL can be used for maximum likelihood estimation as in Wood (2010) or within the Bayesian paradigm as proposed by Drovandi et al. (2018a) and Price et al. (2018). The latter work proposes to sample the approximate posterior generated by the Bayesian Synthetic Likelihood (BSL) approach, $\pi(\theta|s_0)\propto p(\theta)\mathcal{N}(s_0;\mu_\theta,\Sigma_\theta)$, using a MH sampler. Direct calculation of the acceptance probability is not possible because the conditional mean and covariance are unknown for any θ. However, both can be estimated based on m statistics $(s_1,\cdots,s_m)$ sampled from their conditional distribution given θ. More precisely, after simulating

$\mathbf{y}_i \sim f(\mathbf{y}|\theta)$ and setting $s_i = S(\mathbf{y}_i)$, $i = 1, \cdots, m$, one can estimate

$$\hat{\mu}_\theta = \frac{\sum_{i=1}^m s_i}{m} \text{ and } \hat{\Sigma}_\theta = \frac{\sum_{i=1}^m (s_i - \hat{\mu}_\theta)(s_i - \hat{\mu}_\theta)^T}{m-1}, \tag{4.1}$$

so that the synthetic likelihood is $SL(\theta|\mathbf{y}_0) = \mathcal{N}(s_0; \hat{\mu}_\theta, \hat{\Sigma}_\theta)$. The pseudo-code in Algorithm 5 shows the steps involved in the BSL-MCMC sampler. Since each MH step requires calculating the ratios of two SLs calculated at different parameter values, one can anticipate the heavy computational load involved in running the chain for thousands of iterations, especially if sampling data $\mathbf{y}$ is expensive. Note that even though these estimates for the conditional mean and covariance are unbiased, the estimated value of the Gaussian likelihood is biased and therefore pseudo marginal MCMC theory is not applicable. Price et al. (2018) presented an unbiased Gaussian likelihood estimator and have empirically showed that using biased and unbiased estimates generally perform similarly. They have also noted that this procedure is robust with respect to m, and showed that using any $m \in \{50, \ldots, 200\}$ produce similar results. The normality assumption for summary statistics is certainly a strong one and may not hold in practice. An et al. (2018) replaced the joint Gaussian assumption with a Gaussian copula with non-parametric marginal estimates (NONPAR-BSL). The estimation is based, as in the BSL framework, on m pseudo-data samples simulated for each θ.

Algorithm 5 Bayesian Synthetic Likelihood (BSL-MCMC).

```
Given $s_0$, number of simulations $m$ and required number of samples $M$.
Get initial $\theta^{(0)}$, estimate $\hat{\mu}_{\theta^{(0)}}, \hat{\Sigma}_{\theta^{(0)}}$ by simulating $m$ statistics given $\theta^{(0)}$.
Define $h(\theta^{(0)}) = \mathcal{N}(s_0; \hat{\mu}_{\theta^{(0)}}, \hat{\Sigma}_{\theta^{(0)}})$.
for $t = 1, \cdots, M$ do
    Generate $\zeta^* \sim q(\cdot|\theta^{(t-1)})$.
    Estimate $\hat{\mu}_{\zeta^*}, \hat{\Sigma}_{\zeta^*}$ by simulating $m$ pseudo-data points $\{\mathbf{y}^{(j)} : 1 \le j \le m\}$ and corresponding statistics $\{S(\mathbf{y}^{(j)}) : 1 \le j \le m\}$ given $\zeta^*$.
    Calculate $h(\zeta^*) = \mathcal{N}(s_0; \hat{\mu}_{\zeta^*}, \hat{\Sigma}_{\zeta^*})$.
    Calculate $\alpha = \min\left\{1, \frac{p(\zeta^*)h(\zeta^*)q(\theta^{(t-1)}|\zeta^*)}{p(\theta^{(t-1)})h(\theta^{(t-1)})q(\zeta^*|\theta^{(t-1)})}\right\}$
    Set $\theta^{(t)} = \zeta^*$ with probability $\alpha$, and $\theta^{(t)} = \theta^{(t-1)}$ otherwise.
end for
```

Clearly, BSL is computationally costly and requires many pseudo-data simulations to obtain Monte Carlo samples of even moderate sizes. As in the case of ABC, attempts were made to reduce its computational cost. In addition to ideas proposed in Wilkinson (2014) and Drovandi et al. (2018b) which can also be used for BSL, Meeds and Welling (2014) proposed to fit a GP model to pairs (s, ζ) and to use it to approximate μ_ζ and Σ_ζ instead of simulating new pseudo-data sets. Since the SL approach assumes normality of each summary statistic (conditional on ζ) the GP approach is justified. However, the estimated likelihood based on GP is no longer unbiased and, therefore, there is no guarantee that the perturbed chain converges to the true posterior distribution. It is further assumed that the covariance matrix Σ_θ is diagonal. To alleviate this strong restriction, Everitt (2017) proposed to use a bootstrap approach for the estimation of the conditional covariance and a local linear regression model for conditional mean of the summary statistics. He also uses an SMC sampler along with the SL formulation to arrive at the final approximate posterior distribution. In this paper, instead of relying on GP or local linear regression, we implement kNN to approximate the conditional mean and covariance for any proposal ζ^*. Under weak assumptions the estimate is proved to be uniformly weakly consistent. In particular, we propose to store and utilize past simulations of (ζ, s) to approximate $\mu_{\zeta^*}, \Sigma_{\zeta^*}$ for any $\zeta^* \in \Theta$, greatly reducing the computational burden. As in the previous section, we separate the samples used to update the chain from the samples used to enrich the history of the chain. The approach can be extended for NONPAR-BSL but we will not pursue this development here. The K-Nearest-Neighbor (kNN) method is used as a non-parametric estimation tool for different quantities described above. As will be shown in Section 6 with the proposed method we can control the error between the intended stationary distribution and that of the proposed accelerated MCMC.

Approximated Bayesian Synthetic Likelihood (ABSL)

If we set $s_0 = S(\mathbf{y}_0)$ and assume the conditional normality for s_0, the objective is to sample from

$$\pi(\theta|s_0) \propto p(\theta)\mathcal{N}(s_0; \mu_\theta, \Sigma_\theta). \tag{4.2}$$

During the MCMC run, the proposal ζ^* is generated from $q(\cdot)$ and the history $\mathcal{Z}_N$ is enriched using $\tilde{\zeta}^* \sim q(\cdot)$, $\{\tilde{\mathbf{y}}^{*(j)}\}_{j=1}^m \overset{iid}{\sim} f(\mathbf{y}|\tilde{\zeta}^*)$ and $\{\tilde{s}^{*(j)} = S(\tilde{\mathbf{y}}^{*(j)})\}_{j=1}^m$. Then for any ζ, the conditional mean and covariance of interest is estimated using past samples as weighted averages:

$$\begin{aligned} \hat{\mu}_\zeta &= \frac{\sum_{n=1}^N [W_{Nn}(\zeta) \sum_{j=1}^m \tilde{s}_n^{(j)}]}{m \sum_{n=1}^N W_{Nn}(\zeta)}, \\ \hat{\Sigma}_\zeta &= \frac{\sum_{i=1}^N [W_{Nn}(\zeta) \sum_{j=1}^m (\tilde{s}_n^{(j)} - \hat{\mu}_\zeta)(\tilde{s}_n^{(j)} - \hat{\mu}_\zeta)^T]}{m \sum_{i=1}^N W_{Nn}(\zeta)}. \end{aligned} \tag{4.3}$$

The weights are functions of distance between proposed value and parameters' values from the past, i.e. $W_{Nn}(\zeta) = W(\|\zeta - \tilde{\zeta}_n\|)$, where $\|\cdot\|$ is the Euclidean norm. To get appropriate convergence properties we use the kNN approach to calculate weights W_{Nn}, where only the $K = \sqrt{N}$ closest values to ζ are used in the calculation of conditional means and covariances. As in the previous section, uniform (U) and linear (L) weights are used. We expect that the use of the chain's cumulated history can significantly speed up the whole procedure since it relieves the pressure to simulate many data sets $\mathbf{y}$ at every step. The use of the independent Metropolis kernel ensures that $\mathcal{Z}_N$ contains independent draws which is required for theoretical validation in Section 6. We will also prove that under mild assumptions and if Θ is compact, the proposed algorithm exhibits good error control properties. In order to get a rough idea about the proposal, we propose to perform finite adaptation using J updates of the transition kernel during the burn-in period. Section A in the Supplementary Material (Levi and Craiu, 2020) details the proposed Approximated BSL (ABSL) method. For the simulations reported in the next section, we have used $c = 1.5$ and $J = 15$ to be consistent with ABC-related procedures.

5 Numerical Experiments

We analyze the following statistical models:

(MA2) Simple Moving Average model of lag 2 (Supplementary Material (Levi and Craiu, 2020));

(R) Ricker's model;

(SVG) Stochastic volatility with Gaussian emission noise (Supplementary Material (Levi and Craiu, 2020));

(SVS) Stochastic volatility with α-Stable errors.

For all these models, the simulation of pseudo data for any parameter is simple and computationally fast, but the use of standard estimation methods can be quite challenging, especially for (R), (SVG) and (SVS). Before running the proposed algorithms, we define the discrepancy function as $\delta = d(S(\mathbf{y}), s_0) = (S(\mathbf{y}) - s_0)^T A(S(\mathbf{y}) - s_0)$ where matrix A along with the sequence of thresholds $\epsilon_0 < \epsilon_1 < \ldots < \epsilon_{15}$ are estimated from a pilot run (details are provided in the Supplementary Material (Levi and Craiu, 2020)). Moreover we introduce a large number $L = 10^{10}$ to restrict prior distributions for several models in order to guarantee the compactness assumption of the parameter space. In all the examples below this assumption is satisfied. We compare the following algorithms:

(SMC) Standard Sequential Monte Carlo for ABC;

(ABC-RW) The modified ABC-MCMC algorithm which updates ϵ and the random walk Metropolis transition kernel during burn-in;

(ABC-IS) The modified ABC-MCMC algorithm which updates ϵ and the Independent Metropolis transition kernel during burn-in;

(BSL-RW) Modified BSL where it adapts the random walk Metropolis transition kernel during burn-in;

(BSL-IS) Modified BSL where it adapts the independent Metropolis transition kernel during burn-in;

(AABC-U) Approximated ABC-MCMC with independent proposals and uniform (U) weights;

(AABC-L) Approximated ABC-MCMC with independent proposals and linear (L) weights;

(ABSL-U) Approximated BSL-MCMC with independent proposals and uniform (U) weights;

(AABC-L) Approximated BSL-MCMC with independent proposals and linear (L) weights;

(Exact) Likelihood is computable and posterior samples are generated using an MCMC algorithm that is example-specific.

For SMC, 500 particles were used, total number of iterations for ABC-RW, ABC-IS, AABC-U, AABC-L, ABSL-U and ABSL-L is 50000 with 10000 for burn-in. Since BSL-RW and BSL-IS are much more computationally expensive, the total number of iterations were fixed at 10000 with 2000 burn-in and 50 pseudo-data simulations for every proposed parameter value (i.e. $m = 50$). The Exact chain was run for 5000 iterations and 2000 for the burn-in. It must be pointed out that all approximate samplers are based on the same summary statistics, same discrepancy function and the same ϵ sequence, so that they all start with the same initial conditions.

For more reliable results we compare these sampling algorithms under data set replications. In this study we set the number of replicates $R = 100$, so that for each model 100 data sets were generated and each one was analyzed with the described above sampling methods. Various statistics and measures of efficiency were calculated for every model and data set, letting $\theta_{rs}^{(t)}$ represent posterior samples from replicate $r = 1, \cdots, R$, iteration $t = 1, \cdots, M$ and parameter component $s = 1, \cdots, q$ and similarly $\tilde{\theta}_{rs}^{(t)}$ posterior from an exact chain (all draws are after the burn-in period). We let θ_s^{true} denote the true parameter that generated the data. Moreover let $D_{rs}(x)$, $\tilde{D}_{rs}(x)$ be estimated density function at replicate $r = 1, \cdots, R$ and components $s = 1, \cdots, q$ for approximate and exact chains respectively. Then the following quantities are defined:

$$\begin{aligned}
&\text{Diff in mean (DIM)} = Mean_{r,s}(|Mean_t(\theta_{rs}^{(t)}) - Mean_t(\tilde{\theta}_{rs}^{(t)})|),\\
&\text{Diff in covariance (DIC)} = Mean_{r,s}(|Cov_t(\theta_{rs}^{(t)}) - Cov_t(\tilde{\theta}_{rs}^{(t)})|),\\
&\text{Total Variation (TV)} = Mean_{r,s}\left(0.5\int |D_{rs}(x) - \tilde{D}_{rs}(x)|dx\right),\\
&\text{Bias}^2 = Mean_s\left(\left(Mean_{tr}(\theta_{rs}^{(t)}) - \theta_s^{true}\right)^2\right),\\
&\text{VAR} = Mean_s(Var_r(Mean_t(\theta_{rs}^{(t)}))),\\
&\text{MSE} = \text{Bias}^2 + \text{VAR},
\end{aligned}$$

where $Mean_t(a_{st})$ is defined as the average of $\{a_{st}\}$ over index t and in similar manner $Var_t(a_{st})$ and $Cov_t(a_{st})$ represent variance and covariance respectively. The first three measures are useful in determining how close are the posterior draws from different samplers to the draws generated by the exact chain (when it is available). On the other hand, the last three are standard quantities that measure how close are the posterior means to the true parameters that generated the data. To study the efficiency of the proposed algorithms we need to take into account the CPU time needed to run a chain as well as the auto-correlation properties. Define the auto-correlation time (ACT) for every parameter's component and replicate of samples $\theta_{rs}^{(t)}$ as:

$$\text{ACT}_{rs} = 1 + 2\sum_{a=1}^{\infty} \rho_a(\theta_{rs}^{(t)}), \tag{5.1}$$

where ρ_a is the auto-correlation coefficient at lag a. In practice we sum all the lags up to the first negative correlation. Letting $M - B$ to be the number of chain iterations (after burn-in) and CPU_r correspond to the total CPU time to run the whole chain during replicate r, we use Effective Sample Size (ESS) and Effective Sample Size per CPU (ESS/cpu) as:

$$\begin{aligned}
&\mathrm{ESS} = Mean_{rs}((M-B)/\mathrm{ACT}_{rs}),\\
&\mathrm{ESS/cpu} = Mean_{rs}((M-B)/\mathrm{ACT}_{rs}/CPU_r).
\end{aligned} \tag{5.2}$$

Note that these indicators are averaged over the parameter components and replicates. ESS intuitively can be thought as the approximate number of "independent" samples out of $M-B$, the higher is ESS the more efficient is the sampling algorithm, when ESS is combined with CPU (ESS/cpu) it provides a powerful indicator for MCMC's efficiency. Generally, a sampler with highest ESS/cpu is preferred as it produces larger number of "independent" draws per unit time. In the case of the SMC sampler, the formulas above are generally not applicable and therefore ACT and ESS are not calculated for this sampler. Instead, the efficiency of the samplers is assessed by examining the variance of the Monte Carlo estimators of various characteristics of the posterior distribution. Here, we fix one data set and generate $R = 100$ replicates of posterior samples from each algorithm. Denote A_{rs} represent an estimate of a population characteristic (quantile, expectation, etc.) and CPU_r the total CPU time to run the sampler for replicate $r = 1, \cdots, R$ and parameter component $s = 1, \cdots, q$, we define the following measure of efficiency:

$$\mathrm{V}_{\mathrm{A}} \times \mathrm{cpu} = Mean_s(Var_r(\mathrm{A}_{rs})) \times Mean_r(CPU_r). \tag{5.3}$$

Generally, a sampler with lower $\mathrm{V}_{\mathrm{A}} \times$cpu is preferable. We consider three characteristics A: $F_\theta^{-1}(0.05)$ (A=QL, lower quantile), $F_\theta^{-1}(0.95)$ (A=QU, upper quantile) and $E(\theta)$ (A=E, expectation). Please note that this method can be applied for the SMC sampler but it is less reliable than ESS/cpu since it is only based on one posterior distribution. For fair comparisons we generate 1000 posterior samples from each sampler after burn-in period, so that $M-B = 1000$ for all the samplers except for SMC, and 1000 particles for SMC.

Additional simulations are included in the Supplementary Material (Levi and Craiu, 2020). Those are used to assess the performance of the proposed samplers with the random walk Metropolis (RWM) kernel instead of the independent one. These samplers are not justifiable by the theory developed in Section 6 but still show good performance in terms of the proximity to the true target distribution and the efficiency. When the models considered have a large-dimensional parameter space, the construction of an independent proposal may be difficult, and RWM offers a viable alternative. Future work will focus on developing theoretical foundations for the RWM implementation.

5.1 Ricker's Model

Ricker's model is frequently analyzed using Synthetic Likelihood procedures (Wood, 2010; Price et al., 2018). It is a particular instance of a hidden Markov model:

$$\begin{aligned}
&x_{-49} = 1; \quad z_i \overset{iid}{\sim} \mathcal{N}(0, \exp(\theta_2)^2); \quad i = \{-48, \cdots, n\},\\
&x_i = \exp(\exp(\theta_1))x_{i-1}\exp(-x_{i-1} + z_i); \quad i = \{-48, \cdots, n\},\\
&y_i = Pois(\exp(\theta_3)x_i); \quad i = \{-48, \cdots, n\},
\end{aligned} \tag{5.4}$$

where $Pois(\lambda)$ is Poisson distribution with mean parameter λ and $n = 100$. Only $\mathbf{y} = (y_1, \cdots, y_n)$ sequence is observed, because the first 50 values are ignored. Note that all parameters $\theta = (\theta_1, \theta_2, \theta_3)$ are unrestricted, the prior is given as (each prior parameter is independent):

$$\theta_1 \sim \mathcal{N}(0, 1, -L, L),\ \theta_2 \sim Unif[-2.3, 0], \theta_3 \sim \mathcal{N}(0, 4, -L, L). \tag{5.5}$$

Here $N(a, b, c, d)$ is defined as a truncated normal distribution with mean a, variance b, lower and upper bounds c and d, respectively. We restrict the range of θ_2 as all algorithms become unstable for θ_2 outside this interval. Note that the marginal distribution of $\mathbf{y}$ is not available in closed form, but the transition distribution of hidden variables $X_i|x_{i-1}$ and the emission probabilities $Y_i|x_i$ are known and hence we can run the Particle MCMC (PMCMC) (Andrieu et al., 2010) or Ensemble MCMC (Shestopaloff and Neal, 2013) to sample from the posterior distribution $\pi(\theta|\mathbf{y}_0)$. Here we are utilizing the Particle MCMC with 100 particles. As suggested in Wood (2010) we set $\theta_0 = (\log(3.8), 0.9, 2.3)$ and define the summary statistics $S(\mathbf{y})$ as the 14-dimensional vector whose components are:

(C1) $\#\{i : y_i = 0\}$,

(C2) Average of $\mathbf{y}$, $\bar{y}$,

(C3:C7) Sample auto-correlations at lags 1 through 5,

(C8:C11) Coefficients $\beta_0, \beta_1, \beta_2, \beta_3$ of cubic regression
$(y_i - y_{i-1}) = \beta_0 + \beta_1 y_i + \beta_2 y_i^2 + \beta_3 y_i^3 + \epsilon_i$, $i = 2, \ldots, n$,

(C12:C14) Coefficients $\beta_0, \beta_1, \beta_2$ of quadratic regression $y_i^{0.3} = \beta_0 + \beta_1 y_{i-1}^{0.3} + \beta_2 y_{i-1}^{0.6} + \epsilon_i$, $i = 2, \ldots, n$.

Figure 1 shows trace-plots, histograms and ACF function for AABC-U, ABSL-U and ABC-RW samplers for θ_1 (red lines correspond to the true parameter). Additional plots for all the parameters are included in the Supplementary Material (Levi and Craiu, 2020).

We show here ABC-RW instead of ABC-IS because the latter exhibits a poorer performance. The main observation is that the mixing of AABC-U is much better than that of ABC-RW with smaller auto-correlation values. ABSL-U exhibits a similar performance (plot is included in the Supplementary Material (Levi and Craiu, 2020)). To see how close are the draws from simulation-based algorithms to the draws from the Exact chain, in the Supplementary Material (Levi and Craiu, 2020)) we include a plot of the estimated approximate posterior marginal densities. A more general study, where results are averaged over 100 independent replicates, is shown in Table 1. Clearly, the proposed methods clearly outperform in terms of overall efficiency ESS/cpu, $\mathrm{V_{QL}} * \mathrm{cpu}$, $\mathrm{V_{QU}} * \mathrm{cpu}$ and $\mathrm{V_E} * \mathrm{cpu}$. For instance, AABC-U is about 20 times more efficient than standard ABC-RW in terms of ESS/cpu and shows improvement over SMC sampler when variance of quantiles, multiplied by CPU time, is considered. ABSL-U is 6 times more efficient than BSL-RW in terms of ESS/cpu and shows considerable improvement in efficiency when variance of quantiles and mean, multiplied by CPU time, is

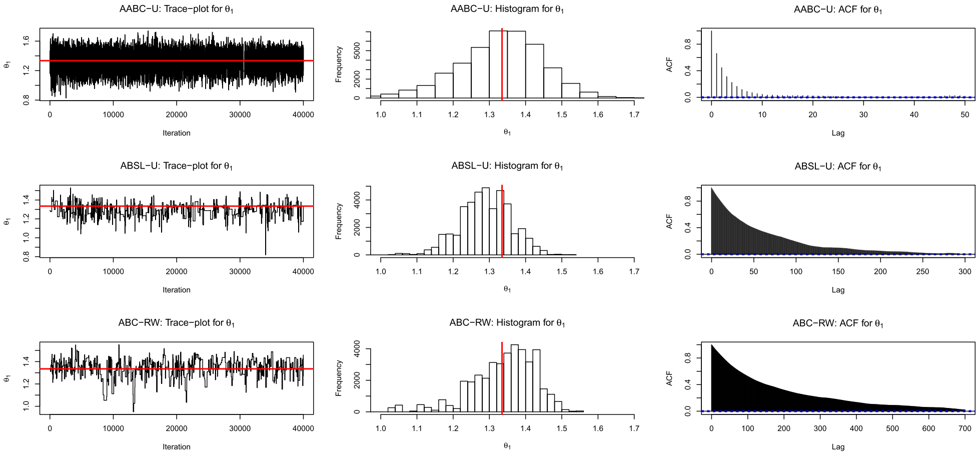

Figure 1: Ricker's model. Each row corresponds to AABC-U (top row), ABSL-U (middle row) and ABC-RW (bottom row) and shows in order from left to right: Trace-plot, Histogram and Auto-correlation function for θ_1 samples. Red lines represent true parameter value.

considered. At the same time DIM, DIC, TV and MSE are generally smaller for the approximate methods. However, for this model, samplers with linear weights show minor loss in efficiency compared to uniform weights.

5.2 Stochastic Volatility with α-Stable Errors

When analyzing stationary time series, it is frequently observed that the periods of high and low volatility alternate. Such phenomenon is called *volatility clustering*, see for example (Lux and Marchesi, 2000). One way to model such a behaviour is through a Stochastic Volatility (SV) model, where variances of the observed time series depend on hidden states that themselves form a stationary time series. The standard SV model assumes that the conditional distribution of the observed variables is Gaussian (see Supplementary Material (Levi and Craiu, 2020) for more details). Frequently, in the financial time series, a large sudden drop occurs, thus raising serious doubts about the latter assumption. Often, it is suggested to use heavy tailed distributions (instead of Gaussian) to model financial data. We consider a family of distributions named α-Stable, denoted $Stab(\alpha, \beta)$, with two parameters $\alpha \in (0, 2]$ (stability parameter) and $\beta \in [-1, 1]$ (skew parameter). Two special cases are $\alpha = 1$ and $\alpha = 2$ which correspond to Cauchy and Gaussian distribution, respectively. Note that for $\alpha < 2$ the distribution has undefined variance. We define the following SV model with α-Stable errors with parameter $\theta = (\theta_1, \theta_2, \theta_3, \theta_4)^T \in \mathbf{R}^4$:

$$\begin{aligned}
&x_1 \sim \mathcal{N}(0, 1/(1-\theta_1^2)); \quad v_i \overset{iid}{\sim} \mathcal{N}(0,1); \quad w_i \overset{iid}{\sim} Stab(\theta_4, -1); \quad i = \{1, \cdots, n\}, \\
&x_i = \theta_1 x_{i-1} + v_i; \quad i = \{2, \cdots, n\}, \\
&y_i = \sqrt{\exp(\theta_2 + \exp(\theta_3) x_i)} w_i; \quad i = \{1, \cdots, n\}.
\end{aligned} \tag{5.6}$$

	Diff with Exact			Diff with True Par		Efficiency			
Sampler	DIM	DIC	TV	$\sqrt{\text{Bias}^2}$	$\sqrt{\text{MSE}}$	ESS/cpu	$\text{V}_{\text{QL}} * \text{cpu}$	$\text{V}_{\text{QU}} * \text{cpu}$	$\text{V}_{\text{E}} * \text{cpu}$
SMC	0.152	0.018	0.378	0.086	0.219	–	0.655	0.180	0.065
ABC-RW	0.135	0.020	0.389	0.059	0.189	0.199	1.283	0.760	0.907
ABC-IS	0.139	0.022	0.485	0.063	0.205	0.099	0.800	1.875	0.881
AABC-U	0.147	0.028	0.402	0.076	0.204	4.390	0.320	0.101	0.063
AABC-L	0.141	0.026	0.392	0.070	0.201	5.193	0.505	0.065	0.245
BSL-RW	0.129	0.008	0.382	0.038	0.209	0.030	3.280	1.272	0.900
BSL-IS	0.122	0.008	0.455	0.022	0.198	0.007	31.815	4.890	6.892
ABSL-U	0.103	0.005	0.377	0.023	0.171	0.180	1.042	0.444	0.242
ABSL-L	0.106	0.005	0.382	0.012	0.173	0.135	2.280	0.664	0.267

Table 1: Simulation Results (Ricker's model): Average Difference in Mean, Difference in Covariance, Total Variation, square roots of Bias and MSE, Effective Sample Size per CPU time, Variances of Lower/Upper Quantiles and Mean times CPU time for every sampling algorithm.

This model is very similar to the simple SV with the only difference that the emission errors follow a α-Stable distribution with unknown stable parameter and fixed skew of -1. We generally prefer a negative skew emission probability to model large negative financial returns. As in the previous simulation example θ_2 and θ_3 are unrestricted. The prior distributions for this model are, independently:

$$\theta_1 \sim Unif[0,1],\ \theta_2 \sim \mathcal{N}(0,1,-L,L),\ \theta_3 \sim \mathcal{N}(0,1,-L,L),\ \theta_4 \sim Unif[1.5,2]. \tag{5.7}$$

We set the true parameters to $\theta_1 = 0.95, \theta_2 = -2, \theta_3 = -1, \theta_4 = 1.8$ and length of the time series $n = 500$. The major challenge with this model is that there are no closed-form densities for α-Stable distributions. Hence, most MCMC samplers, including PMCMC and ensemble MCMC, cannot be used to sample from the posterior. However, sampling data from this family of distributions is feasible which makes it particularly amenable for simulation based methods such as ABC and BSL. For summary statistics we use a 7-dimensional vector whose components are:

(C1) $\#\{i : y_i^2 > \text{quantile}(\mathbf{y}_0^2, 0.99)\}$,

(C2) Average of $\mathbf{y}^2$,

(C3) Standard deviation of $\mathbf{y}^2$,

(C4) Sum of the first 5 auto-correlations of $\mathbf{y}^2$,

(C5) Sum of the first 5 auto-correlations of $\{\mathbf{1}_{\{y_i^2 < \text{quantile}(\mathbf{y}^2, 0.1)\}}\}_{i=1}^n$,

(C6) Sum of the first 5 auto-correlations of $\{\mathbf{1}_{\{y_i^2 < \text{quantile}(\mathbf{y}^2, 0.5)\}}\}_{i=1}^n$,

(C7) Sum of the first 5 auto-correlations of $\{\mathbf{1}_{\{y_i^2 < \text{quantile}(\mathbf{y}^2, 0.9)\}}\}_{i=1}^n$.

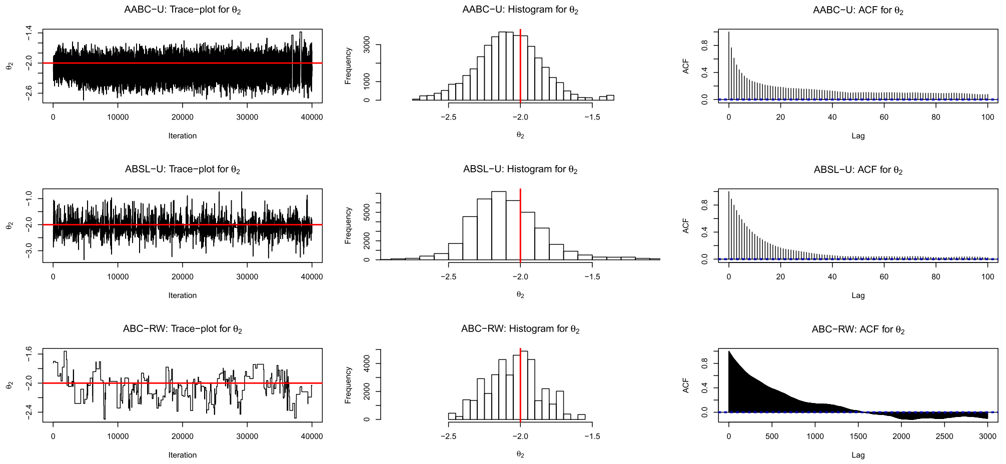

Figure 2: SV α-Stable model. Each row corresponds to AABC-U (top row), ABSL-U (middle row) and ABC-RW (bottom row) and shows in order from left to right: Trace-plot, Histogram and Auto-correlation function for θ_2 samples. Red lines represent true parameter value.

Here quantile($\mathbf{y}, \tau$) is defined as τ-quantile of the sequence $\mathbf{y}$. As was shown in Schmitt et al. (2015) and Dette et al. (2015) the auto-correlation of indicators (under different quantiles) can be very useful in characterizing a time series and that is why we have added (C5), (C6) and (C7) to the summary statistic. We focus here on $\mathbf{y}^2$ and its auto-correlations since the model parameters only affect the variability of $\mathbf{y}$ (auto-correlation of $\mathbf{y}$ is zero for any lag).

Figure 2 illustrates the performance of AABC-U, ABSL-U and ABC-RW algorithms, by plotting the trace plot, the histogram and the auto-correlation function for θ_2. Additional plots for all the parameters are included in the Supplementary Material (Levi and Craiu, 2020). As in the previous example, the mixing of AABC-U is much better than of ABC-RW. Since the exact sampling is not feasible in this example we compare samplers to SMC.

For more general conclusions we show average results in Table 2 over 100 data replicates. To calculate DIM, DIC and TV, samplers are compared to SMC since the exact draws cannot be obtained. Again, efficiency measures for AABC-U, AABC-L, ABSL-U and ABSL-L show significant improvement over the benchmark algorithms. All proposed methods outperform the benchmark samplers in terms of all efficiency measures except for $\text{V}_{\text{QU}} * \text{cpu}$, where SMC outperforms AABC-U. For this example looking at DIM, DIC and TV may be misleading since approximate samplers are compared to another approximate sampler. Much more informative is the MSE, which is very similar across ABC-based and BSL-based algorithms.

	Diff with Exact			Diff with True Par		Efficiency			
Sampler	DIM	DIC	TV	$\sqrt{\text{Bias}^2}$	$\sqrt{\text{MSE}}$	ESS/cpu	$\text{V}_{\text{QL}} * \text{cpu}$	$\text{V}_{\text{QU}} * \text{cpu}$	$\text{V}_{\text{E}} * \text{cpu}$
SMC	0.000	0.000	0.000	0.221	0.299	–	1.754	0.375	0.142
ABC-RW	0.078	0.013	0.205	0.248	0.317	0.069	1.387	0.249	0.565
ABC-IS	0.082	0.015	0.306	0.232	0.320	0.071	0.799	0.871	0.472
AABC-U	0.069	0.012	0.170	0.250	0.310	1.617	0.252	0.511	0.085
AABC-L	0.069	0.013	0.161	0.246	0.305	1.546	0.111	0.093	0.080
BSL-RW	0.044	0.012	0.122	0.225	0.289	0.037	10.847	3.657	3.111
BSL-IS	0.045	0.010	0.125	0.226	0.287	0.084	12.356	5.826	1.246
ABSL-U	0.063	0.013	0.228	0.225	0.289	0.735	0.706	0.495	0.196
ABSL-L	0.061	0.014	0.230	0.236	0.299	0.671	1.372	0.470	0.238

Table 2: Simulation Results (SV α-Stable model): Average Difference in Mean, Difference in Covariance, Total Variation, square roots of Bias and MSE, Effective Sample Size per CPU time, Variances of Lower/Upper Quantiles and Mean times CPU time for every sampling algorithm. In DIM, DIC and TV, samplers are compared to SMC.

6 Theoretical Justifications

In this section we show that the novel approximated ABC MCMC and BSL samplers with independent proposals exhibit proper ergodic properties in the long run. Specifically, we will show that as the number of MCMC iterations increases, the marginal distribution of $\{\theta^{(t)}\}$ converges to the appropriate posterior distribution in total variation and the sample averages converge to the true expectations. In the next two sections we extend the work of Johndrow et al. (2015b) on the perturbed MCMC and then in Section 6.3 discuss necessary conditions for the ergodicity of AABC and ABSL. Note that in Sections 6.1 and 6.2, ϵ corresponds to the discrepancy between the exact and perturbed kernels and not to the threshold in ABC-based methods.

6.1 Notation

We start by reviewing the notation. Let $p(\theta), q(\theta)$ denote the prior and the proposal distributions for $\theta \in \Theta$, respectively. Then, given a proposal ζ^*, the acceptance probability is:

$$\begin{aligned} a(\theta,\zeta^*) &= \min\{1, \alpha(\theta,\zeta^*)\}, \\ \alpha(\theta,\zeta^*) &= \frac{p(\zeta^*)q(\theta)h(\zeta^*)}{p(\theta)q(\zeta^*)h(\theta)}. \end{aligned} \tag{6.1}$$

This independent MH acceptance rate defines an exact transition kernel which we call $P(\cdot,\cdot)$. Suppose $h(\theta)$ is not known, and instead it is estimated using $\hat{h}(\theta|\mathcal{Z}_N)$ where $\mathcal{Z}_N$ are some auxiliary variables.

The acceptance probability for this perturbed algorithm (more on perturbed MCMC can be found in Roberts et al., 1998; Pillai and Smith, 2014; Johndrow and Mattingly, 2017) is:

$$\hat{a}(\theta,\zeta^*;\mathcal{Z}_N) = \min\{1, \hat{\alpha}(\theta,\zeta^*;\mathcal{Z}_N)\}, \text{ where } \hat{\alpha}(\theta,\zeta^*;\mathcal{Z}_N) = \frac{p(\zeta^*)q(\theta)\hat{h}(\zeta^*;\mathcal{Z}_N)}{p(\theta)q(\zeta^*)\hat{h}(\theta;\mathcal{Z}_N)}. \tag{6.2}$$

If the approximate kernel transition is $\hat{P}_N(\cdot,\cdot) = E_{\mathcal{Z}_N}\left[\hat{P}_N(\cdot,\cdot;\mathcal{Z}_N)\right]$, then the initial goal is to show that as $N \to \infty$ the distance between this transition and the exact one converges to zero, where the distance is defined as:

$$\|\hat{P}_N(\cdot,\cdot) - P(\cdot,\cdot)\| = \sup_{\theta} \|\hat{P}_N(\theta,\cdot) - P(\theta,\cdot)\|_{TV}. \tag{6.3}$$

Here $\|\cdot\|_{TV}$ is the "total variation" distance between two measures. First we prove the intuitive result that under strong consistency assumption of $\hat{h}(\theta;\mathcal{Z}_N)$, the perturbed kernel converges to the exact one.

Theorem 6.1. *Suppose Θ is compact, $\sup_\theta \|\hat{h}(\theta;\mathcal{Z}_N) - h(\theta)\| \to 0$ with probability 1 and $h(\theta) > 0$ for all $\theta \in \Theta$. Then for any $\epsilon > 0$ there exists C such that for all $N > C$, $\|\hat{P}_N - P\| < \epsilon$.*

Next let $\mathcal{P}_\epsilon = \{\hat{P}_N : \|\hat{P}_N - P\| < \epsilon\}$ be a collection of the perturbed kernels each ϵ distance from the exact kernel. The main objective is to show that if the chain utilizes a new kernel $\hat{P}_N \in \mathcal{P}_\epsilon$ at every iteration, it still results in the ergodic chain with appropriate convergence results. To achieve that we refer to the work of Johndrow et al. (2015b) on the convergence properties of the perturbed kernels, where the authors assume the same perturbed kernel (ϵ distance to the exact one) in each MCMC iteration. In the next sub-section we will extend this result by allowing to have a different transition kernel from $\mathcal{P}_\epsilon$ at each iteration.

6.2 Extension of Perturbed MCMC Theory

To obtain useful convergence results we need to add the Doeblin Condition assumption about the exact kernel P:

Definition 6.1 (Doeblin Condition)**.** *Given a kernel P, there exists a constant $0 < \alpha < 1$ such that*

$$\sup_{(\theta,\zeta^*)\in\Theta\times\Theta} \|P(\theta,\cdot) - P(\zeta^*,\cdot)\|_{TV} < 1-\alpha.$$

We also choose ϵ (for $\mathcal{P}_\epsilon$) so that $\alpha^* = \alpha + 2\epsilon < 1$ and $\epsilon < \alpha/2$ which by Remark 2.1 in Johndrow et al. (2015b) guarantees that every member of $\mathcal{P}_\epsilon$ satisfies Doeblin Condition with constant α^* and has a unique invariant measure. Thus we define the following 3 assumptions:

(**A1**) Exact transition kernel P satisfies the Doeblin Condition,

(**A2**) For any $\hat{P}_N \in \mathcal{P}_\epsilon$, $\|\hat{P}_N - P\| < \epsilon$,

(**A3**) $\epsilon < \min\{\alpha/2, (1-\alpha)/2\}$, and set $\alpha^* = \alpha + 2\epsilon$.

In Section 6.3 we show conditions for AABC and ABSL methods that guarantee the satisfaction of these 3 assumptions. Next, we let μ to be the invariant measure of the exact kernel P, and denote the perturbed Markov Chain as $\theta^{(0)}, \theta^{(1)}, \ldots, \theta^{(t)}$ with the marginal distribution of the initial state as $\theta^{(0)} \sim \nu = \mu_0$. Also define the marginal distributions of $\theta^{(t)}$ by μ_t, $t = 1, 2, \ldots$, which are equal to

$$\mu_t = \nu \hat{P}_0 \hat{P}_1 \cdots \hat{P}_t,$$

where $\hat{P}_t \in \mathcal{P}_\epsilon$, $t = 1, 2, \ldots$, (note a different perturbed transition kernel is used at each iteration) and $\hat{P}_0$ is defined as an identity transition (for convenience).

First we need to examine the total variation distance between μ and the average measure $\sum_{t=0}^{M-1} \mu_t / M$

$$\left\| \mu - \frac{\sum_{t=0}^{M-1} \nu \hat{P}_0 \cdots \hat{P}_t}{M} \right\|_{TV}, \quad \text{where} \quad \hat{P}_0 = I. \tag{6.4}$$

We prove the following important convergence result:

Theorem 6.2. *Suppose that **(A1)**, **(A2)** and **(A3)** are satisfied. Let ν be any probability measure on $(\Theta, \mathcal{F}_0)$, then*

$$\left\| \mu - \frac{\sum_{t=0}^{M-1} \nu \hat{P}_0 \cdots \hat{P}_t}{M} \right\|_{TV} \leq \frac{(1-(1-\alpha)^M)\|\mu - \nu\|_{TV}}{M\alpha} - \frac{\epsilon(1-(1-\alpha)^M)}{M\alpha^2} + \frac{\epsilon}{\alpha}, \tag{6.5}$$

which implies that this difference can be arbitrary small for sufficiently large M and small enough ϵ.

Next we focus on the following mean squared error (MSE), $E\left[\left(\mu f - \frac{\sum_{t=0}^{M-1} f(\theta^{(t)})}{M}\right)^2\right]$, where f is a bounded function and $\mu f = E_\mu[f(\theta)]$. The main objective here is to find its upper bound when the perturbed MCMC is used and how it depends on the sample size M. We will rely on the following lemma.

Lemma 6.1. *Suppose: **(A2)** and **(A3)** are satisfied; $\theta^{(0)} \sim \nu$, where ν is a probability distribution; $\mu_t = \nu \hat{P}_1 \cdots \hat{P}_t$ is the marginal distribution of $\theta^{(t)}$, $t = 1, 2, \ldots$. Let $f(\theta)$ and $g(\theta)$ be bounded functions with $|f| = \sup_\theta f(\theta)$ and $|g| = \sup_\theta g(\theta)$. Then*

$$cov(f(\theta^{(k)}), f(\theta^{(j)})) \leq 8|f||g|(1-\alpha^*)^{|k-j|}.$$

The next important convergence results follows (similar to Theorem 2.5 of Johndrow et al., 2015b):

Theorem 6.3 (Approximation of MSE)**.** *Suppose that **(A1)**, **(A2)** and **(A3)** are satisfied. Let μ denote the invariant measure of P, $f(\theta)$ be a bounded function and $\theta^{(0)} \sim \nu$, where ν is a probability distribution. Then*

$$\begin{aligned} &E\left[\left(\mu f - \frac{1}{M}\sum_{t=0}^{M-1} f(\theta^{(t)})\right)^2\right] \\ &\leq 4|f|^2 \left(\frac{(1-(1-\alpha)^M)}{M\alpha} - \frac{\epsilon(1-(1-\alpha)^M)}{M\alpha^2} + \frac{\epsilon}{\alpha}\right)^2 \\ &+ 8|f|^2 \left(\frac{1}{M} + \frac{2}{(\alpha^*)^2}\left(\frac{(1-\alpha^*)^{M+1} - (1-\alpha^*)}{M^2} + \frac{(1-\alpha^*) - (1-\alpha^*)^2}{M}\right)\right). \end{aligned} \tag{6.6}$$

In other words this expectation can be made arbitrary small for sufficiently large M and small enough ϵ.

To summarize, the theoretical results show that if the estimator $\hat{h}(\theta;\mathcal{Z}_N)$ of the true likelihood converges uniformly in probability to $h(\theta)$ as N (the number of auxiliary variables) increases, then for any $\epsilon > 0$ there exists a large C so that for $N > C$ all perturbed kernels $\hat{P}_N$ are within ϵ-distance of the exact transition kernel (Theorem 6.1). Moreover if the MCMC is run with the perturbed kernels $\hat{P}_N$ at each iteration then the distribution of the average of the chain's states has almost the same distribution as the target (Theorem 6.2) and the average of any bounded function of the states converges to the true expectation (Theorem 6.3). The error only depends on the number of iterations M and the chosen ϵ. In the next section we provide sufficient conditions for AABC and ABSL methods to fit this framework thus proving their validity.

6.3 Ergodicity of AABC and ABSL

For illustration we focus on AABC but the same process applies to ABSL. Recall, for AABC algorithm we define a function $h(\theta)$ as $P(\delta < \epsilon|\theta)$ where $\delta = \delta(\mathbf{y}, \mathbf{y}_0)$ and $\mathbf{y} \sim f(\mathbf{y}|\theta)$. Unfortunately $h(\theta)$ is unknown and therefore estimated, in particular let $\mathcal{Z}_N = \{\tilde{\zeta}_n, \mathbf{1}_{\{\tilde{\delta}_n<\epsilon\}}\}_{n=1}^N$ represent N independent samples from $q(\zeta)P(\mathbf{1}_{\{\delta<\epsilon\}}|\zeta)$. Actually $\mathcal{Z}_N$ contains past generated samples that were saved before Nth iteration. Given θ (current state) and ζ^* (proposed state) we apply kNN to approximate $h(\theta)$ and $h(\zeta^*)$ by calculating local weighted averages of $\mathbf{1}_{\{\tilde{\delta}_n<\epsilon\}}$ for $\tilde{\zeta}_n$ that are close to θ or ζ^*. To obtain convergence results for the proposed methods, we consider the following assumptions:

(B1) Θ is a compact set.

(B2) $q(\theta) > 0$ continuous density of independent proposal distribution.

(B3) $p(\theta) > 0$ continuous density of prior distribution.

(B4) $h(\theta)$ continuous function of θ.

(B5) In KNN estimation assume that $K(N) = \sqrt{N}$ with uniform or linear weights.

(B6) $E[s^j|\theta]$ and $E[s^j s^k|\theta]$ are continuous functions of θ for every $1 \le j,k \le p$ with s^j representing jth component of summary statistic s.

(B7) $Var[s^j|\theta]$ and $Var[s^j s^k|\theta]$ are bounded functions.

(B8) $|\Sigma_\theta| > a_0$ where $\Sigma_\theta = Var(s|\theta)$ for every $\theta \in \Theta$.

The theoretical justification of the methods also relies on the following two theorems.

Theorem 6.4 (Uniform Consistency of kNN – Cheng, 1984)**.** *Given independent $\{\tilde{\zeta}_n, \tilde{\delta}_n\}_{n=1}^N$, let Θ be support of distribution of $\tilde{\zeta}$, $h(\tilde{\zeta}) = E(\tilde{\delta}|\tilde{\zeta})$ and $\hat{h}_N(\tilde{\zeta}) = \sum_{j=1}^N W_{Nj}\tilde{\delta}_j$ (kNN estimator) (here j are permuted indices that order distances between $\tilde{\zeta}_n$ and $\tilde{\zeta}$ from smallest to largest). Suppose weights W_{Nj} satisfy* *(i)* $\sum_{j=1}^{N} W_{Nj} = 1$,

(ii) $W_{Nj} = 0$ *for* $j > K$, *and* $K = K(N)$ *with* $K \to \infty$ *and* $K/N \to 0$,

(iii) $\sup_N K \max_j W_{Nj} < \infty$.

If

(i) Θ *is compact,*

(ii) $h(\tilde{\zeta})$ *is continuous function,*

(iii) $Var(\tilde{\delta}|\tilde{\zeta})$ *is bounded random variable,*

(iv) $K(N)$ *satisfies* $K/\sqrt{N}\log(N) \to \infty$,

then $\sup_{\tilde{\zeta}\in\Theta} |\hat{h}_N(\tilde{\zeta}) - h(\tilde{\zeta})| \to 0$ *with probability 1.*

Theorem 6.5 (Independent Metropolis sampler - Mengersen et al. (1996))**.** *Suppose* $\theta^{(t)}$ *is a MH Markov chain with invariant distribution* $\pi(\theta)$*, independent proposal* $q(\theta)$ *and acceptance probabilities* $a(\theta, \zeta^*) = \min\left\{1, \frac{\pi(\zeta^*)q(\theta)}{\pi(\theta)q(\zeta^*)}\right\}$.

If there exists $\beta > 0$ *such that* $q(\theta)/\pi(\theta) > \beta$ *for all* $\theta \in \Theta$*, then the algorithm is uniformly ergodic so that* $\|P^n(\theta,\cdot) - \pi\|_{TV} < (1-\beta)^n$ *(here* $P^n(\theta,\cdot)$ *is conditional distribution of* $\theta^{(n)}$ *given* $\theta^{(0)} = \theta$*).*

We can now state the main result:

Theorem 6.6 (Ergodicity of AABC)**.** *Consider the proposed AABC sampler with* ϵ *threshold and let:* $p(\theta)$ *denote the prior measure on* Θ, $\mathcal{Z}_N$ *denote simulated pairs* $\{\tilde{\zeta}_n, \mathbf{1}_{\{\tilde{\delta}_n<\epsilon\}}\}_{n=1}^{N}$ *with* $\tilde{\zeta}_n \sim q(\zeta)$ $\forall n$*. Assume* ***(B1)–(B5)*** *hold. Then for sufficiently large* N *(number of past simulations) and* M *(number of chain iterations), assumptions* ***(A1)–(A3)*** *are satisfied and the results established in Theorems 6.2 and 6.3 follow.*

The idea behind the proof relies on the results stated here. First Theorem 6.4 guarantees that the kNN estimates converge uniformly in probability for all points in the compact parameter space. The result in Theorem 6.1 yields that the perturbed kernels approach the exact kernel for sufficiently large history $\mathcal{Z}_N$. Since the independent proposal sampler is used in the AABC/ABSL algorithm, the Doeblin condition of the exact kernel follows from Theorem 6.5. Thus, it can be shown that the assumptions **(A1)–(A3)** are satisfied and the conclusions of Theorems 6.2 and 6.3 easily follow.

The theorem states that under assumptions **(B1)–(B5)** the proposed approximate ABC method has the necessary convergence properties, i.e. the distribution of the average of Markov chain states converges in total variation to the true target distribution and the sample average of any bounded function converges in mean square to the true expectation. Generally **(B1)** is a strong assumption, but in practice the parameter space can be restricted to a closed and bounded region that satisfies this assumption. **(B2)** and

(B5) are immediately satisfied by the construction since, for the proposed algorithm, the multivariate normal distribution is used as an independent proposal distribution and we set $K(N) = \sqrt{N}$. The continuity of the prior density function **(B3)** generally can be satisfied since it is in the control of the analyst. All these assumptions are met by the simulation studies in Section 5. **(B4)** is harder to check since $h(\theta) = P(\delta < \epsilon|\theta)$ is an unknown function, but it is intuitive to accept it since generally there is no reason for $h(\theta)$ to make a sudden jump or drop in its values when θ changes by a small amount especially when the parameter space is compact.

The main result for ABSL is similar but requires three additional assumptions about the behavior of the summary statistics. The idea of the proof is similar to Theorem 6.6.

Corollary 6.1 (Ergodicity of ABSL)**.** *Assume that **(B1)**–**(B8)** hold. Let $p(\theta)$ be the prior distribution on Θ, $h(\theta) = \mathcal{N}(s_0; \mu_\theta, \Sigma_\theta)$, and $\mathcal{Z}_N$ the set of simulated pairs $\{\tilde{\zeta}_n, \{\tilde{s}_n^{(j)}\}_{j=1}^m\}_{n=1}^N$. Then for sufficiently large N (number of past simulations) and M (number of chain iterations), assumptions **(A1)**–**(A3)** are satisfied and the results established in Theorems 6.2 and 6.3 follow.*

The continuity of the expectations and covariances of the summary statistics, **(B6)**, is another hard-to-check assumption. Similarly, **(B7)** and **(B8)** are expected to hold due to the compactness of the parameter space (**(B1)**). We therefore assume that **(B1)**–**(B8)** are met for all the simulation scenarios in Section 5 and the data analysis of the real data set in the Supplementary Material (Levi and Craiu, 2020). All the proofs of the above theorems and corollaries can be found in the Supplementary Material.

7 Discussion and Future Work

In this paper we propose to speed up generic ABC-MCMC and BSL algorithms by reusing past simulations. This approach significantly accelerates the process and can be very useful for models where simulation of a pseudo data set is computationally expensive. We have presented theoretical arguments and sufficient assumptions for convergence properties of the perturbed chain. The performance of these strategies were examined via a series of simulations under different models. All simulations summaries show that the proposed methods significantly improve mixing and efficiency of the chain. When the likelihood is intractable, a researcher can choose either AABC or ABSL for inferential purposes. If the summary statistics can be reasonably trusted to follow a multivariate normal distribution for each parameter value, we recommend using ABSL over generic BSL as it is an order of magnitude faster and does not require the selection of a threshold ϵ. On the other hand, when one has no particular reason to assume a Gaussian distribution for the summary statistic, AABC should be preferred.

Further work needs to be done in order to extend the application of these ideas to more complex models. First, in the current implementation it is assumed that the dimension of the parameter space must be small or moderate since it is well known that the kNN approach is unreliable in higher dimensions due to the curse of dimensionality. Second, the size of the history set, i.e. the set of past samples $\mathcal{Z}_N$, increases as the chain progresses, which can create memory issues when the number of MCMC iterations, M, is very large. In our experiments we have used 50,000 to 100,000 iterations and have not encountered any memory problems. Third, kNN is a non-parametric method which needs to calculate the distances from the proposed sample ζ^* to all the samples in the historical set, $\mathcal{Z}_N$. Not surprisingly, this procedure becomes computationally expensive as the number of iterations (and thus cardinality of $\mathcal{Z}_N$) increases. So the advantage of these methods may diminish for a large scale MCMC. To remedy this, we can implement a more efficient kNN which uses KD-tree for faster distance computations (Sherlock et al., 2017) and/or stop adding the new members to $\mathcal{Z}_N$ after some large N. Finally, it is clear that the independent Metropolis sampling kernel used throughout the paper might not be efficient, especially if the posterior is far from being a Gaussian. It turns out that random walk kernels (which generally have good performance as shown in the Supplementary Material (Levi and Craiu, 2020)) could be a better alternative, but require further theoretical developments. We hope that by drawing attention to the alternative approaches advanced in the paper, we will spur the Bayesian community's interest for developing strategies that are economical and adaptive for approximate Bayesian computation methods.

Supplementary Material

Supplementary Material for "Finding our Way in the Dark: Approximate MCMC for Approximate Bayesian Methods" (DOI: 10.1214/20-BA1250SUPP; .pdf).

References

An, Z., Nott, D. J., and Drovandi, C. (2018). "Robust Bayesian Synthetic Likelihood via a Semi-Parametric Approach." *arXiv preprint arXiv:1809.05800*. MR4065218. doi: https://doi.org/10.1007/s11222-019-09904-x. 202

Andrieu, C., Doucet, A., and Holenstein, R. (2010). "Particle Markov chain Monte Carlo methods." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3): 269–342. MR2758115. doi: https://doi.org/10.1111/j.1467-9868.2009.00736.x. 195, 197, 207

Andrieu, C. and Roberts, G. O. (2009). "The pseudo-marginal approach for efficient Monte Carlo computations." *The Annals of Statistics*, 37(2): 697–725. MR2502648. doi: https://doi.org/10.1214/07-AOS574. 197, 201

Andrieu, C. and Vihola, M. (2015). "Convergence properties of pseudo-marginal Markov chain Monte Carlo algorithms." *The Annals of Applied Probability*, 25(2): 1030–1077. MR3313762. doi: https://doi.org/10.1214/14-AAP1022. 201

Balakrishnan, S., Madigan, D., et al. (2006). "A one-pass sequential Monte Carlo method for Bayesian analysis of massive datasets." *Bayesian Analysis*, 1(2): 345–361. MR2221268. doi: https://doi.org/10.1214/06-BA112. 194

Baragatti, M. and Pudlo, P. (2014). "An overview on approximate Bayesian computation." In *ESAIM: Proceedings*, volume 44, 291–299. EDP Sciences. MR3178623. doi: https://doi.org/10.1051/proc/201444018. 194

Bardenet, R., Doucet, A., and Holmes, C. (2014). "Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach." In *International Conference on Machine Learning (ICML)*, 405–413. 194

Beaumont, M. A. (2003). "Estimation of population growth or decline in genetically monitored populations." *Genetics*, 164(3): 1139–1160. 201

Bentley, J. L. (1975). "Multidimensional binary search trees used for associative searching." *Communications of the ACM*, 18(9): 509–517. 201

Biau, G. and Devroye, L. (2015). *Lectures on the nearest neighbor method*. Springer. MR3445317. doi: https://doi.org/10.1007/978-3-319-25388-6. 200

Boland, A., Friel, N., Maire, F., et al. (2018). "Efficient MCMC for Gibbs random fields using pre-computation." *Electronic Journal of Statistics*, 12(2): 4138–4179. MR3890764. doi: https://doi.org/10.1214/18-EJS1504. 194

Bornn, L., Pillai, N., Smith, A., and Woodard, D. (2014). "One pseudo-sample is enough in approximate Bayesian computation MCMC." *Biometrika*, 99(1): 1–10. 197

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011). *Handbook of Markov chain Monte Carlo*. CRC Press. MR2742422. doi: https://doi.org/10.1201/b10905. 194

Cheng, P. E. (1984). "Strong consistency of nearest neighbor regression function estimators." *Journal of Multivariate Analysis*, 15(1): 63–72. MR0755816. doi: https://doi.org/10.1016/0047-259X(84)90067-8. 200, 214

Craiu, R. V. and Rosenthal, J. S. (2014). "Bayesian computation via Markov chain Monte Carlo." *Annual Review of Statistics and Its Application*, 1: 179–201. 193

Dette, H., Hallin, M., Kley, T., Volgushev, S., et al. (2015). "Of copulas, quantiles, ranks and spectra: An L_1-approach to spectral analysis." *Bernoulli*, 21(2): 781–831. MR3338647. doi: https://doi.org/10.3150/13-BEJ587. 210

Drovandi, C. C. (2018). "ABC and indirect inference." In *Handbook of Approximate Bayesian Computation*, 179–209. Chapman and Hall/CRC. MR3889284. 194

Drovandi, C. C., Grazian, C., Mengersen, K., and Robert, C. (2018a). "Approximating the Likelihood in ABC." In *Handbook of Approximate Bayesian Computation*, 321–368. Chapman and Hall/CRC. MR3889289. 194, 201

Drovandi, C. C., Moores, M. T., and Boys, R. J. (2018b). "Accelerating pseudo-marginal MCMC using Gaussian processes." *Computational Statistics & Data Analysis*, 118: 1–17. MR3715260. doi: https://doi.org/10.1016/j.csda.2017.09.002. 198, 201, 202

Entezari, R., Craiu, R. V., and Rosenthal, J. S. (2018). "Likelihood inflating sampling algorithm." *Canadian Journal of Statistics*, 46(1): 147–175. MR3767170. doi: https://doi.org/10.1002/cjs.11343. 194

Everitt, R. G. (2017). "Bootstrapped synthetic likelihood." *arXiv preprint arXiv:1711.05825*. 202

Fearnhead, P. and Prangle, D. (2012). "Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation."

Journal of the Royal Statistical Society: Series B (Statistical Methodology), 74(3): 419–474. MR2925370. doi: https://doi.org/10.1111/j.1467-9868.2011.01010.x. 196

Filippi, S., Barnes, C. P., Cornebise, J., and Stumpf, M. P. (2013). "On optimality of kernels for approximate Bayesian computation using sequential Monte Carlo." *Statistical Applications in Genetics and Molecular Biology*, 12(1): 87–107. MR3044402. doi: https://doi.org/10.1515/sagmb-2012-0069. 197

Fix, E. and Hodges, J. (1951). "Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties USAF School of Aviation Medicine, Randolph Field." Technical report, Texas, Tech. Report 4. 200

Friedman, J. H., Bentley, J. L., and Finkel, R. A. (1977). "An algorithm for finding best matches in logarithmic expected time." *ACM Transactions on Mathematical Software (TOMS)*, 3(3): 209–226. 201

Järvenpää, M., Gutmann, M. U., Pleska, A., Vehtari, A., Marttinen, P., et al. (2018). "Efficient acquisition rules for model-based approximate Bayesian computation." *Bayesian Analysis*. MR3934099. doi: https://doi.org/10.1214/18-BA1121. 198

Johndrow, J. E. and Mattingly, J. C. (2017). "Error bounds for Approximations of Markov chains used in Bayesian Sampling." *arXiv preprint arXiv:1711.05382*. 194, 211

Johndrow, J. E., Mattingly, J. C., Mukherjee, S., and Dunson, D. (2015a). "Approximations of Markov chains and high-dimensional Bayesian inference." *arXiv preprint arXiv:1508.03387*. 198

Johndrow, J. E., Mattingly, J. C., Mukherjee, S., and Dunson, D. (2015b). "Optimal approximating Markov chains for Bayesian inference." *arXiv preprint arXiv:1508.03387*. 194, 211, 212, 213

Kolaczyk, E. D. and Csárdi, G. (2014). *Statistical analysis of network data with R*, volume 65. Springer. MR3288852. doi: https://doi.org/10.1007/978-1-4939-0983-4. 195

Lee, A. (2012). "On the choice of MCMC kernels for approximate Bayesian computation with SMC samplers." In *Proceedings of the 2012 Winter Simulation Conference (WSC)*, 1–12. IEEE. 197

Lee, A., Andrieu, C., and Doucet, A. (2012). "Discussion of constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation." *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 74(3): 449–450. 196

Levi, E. and Craiu, R. V. (2020). "Supplementary Material of "Finding our Way in the Dark: Approximate MCMC for Approximate Bayesian Methods"." *Bayesian Analysis.* doi: https://doi.org/10.1214/20-BA1250SUPP. 195, 201, 203, 204, 206, 207, 208, 210, 216, 217

Lux, T. and Marchesi, M. (2000). "Volatility clustering in financial markets: a microsimulation of interacting agents." *International Journal of Theoretical and*

Applied Finance, 3(04): 675–702. MR1795484. doi: https://doi.org/10.1142/S0219024900000826. 208

Maclaurin, D. and Adams, R. P. (2015). "Firefly Monte Carlo: Exact MCMC with subsets of data." In *Twenty-Fourth International Joint Conference on Artificial Intelligence*. 194

Marin, J.-M., Pillai, N. S., Robert, C. P., and Rousseau, J. (2014). "Relevant statistics for Bayesian model choice." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(5): 833–859. MR3271169. doi: https://doi.org/10.1111/rssb.12056. 196

Marin, J.-M., Pudlo, P., Robert, C. P., and Ryder, R. J. (2012). "Approximate Bayesian computational methods." *Statistics and Computing*, 22(6): 1167–1180. MR2992292. doi: https://doi.org/10.1007/s11222-011-9288-2. 194

Marjoram, P., Molitor, J., Plagnol, V., and Tavaré, S. (2003). "Markov chain Monte Carlo without likelihoods." *Proceedings of the National Academy of Sciences*, 100(26): 15324–15328. 196

Meeds, E. and Welling, M. (2014). "GPS-ABC: Gaussian process surrogate approximate Bayesian computation." *arXiv preprint arXiv:1401.2838*. 202

Mengersen, K. L., Tweedie, R. L., et al. (1996). "Rates of convergence of the Hastings and Metropolis algorithms." *The Annals of Statistics*, 24(1): 101–121. MR1389882. doi: https://doi.org/10.1214/aos/1033066201. 215

Mitrophanov, A. Y. (2005). "Sensitivity and convergence of uniformly ergodic Markov chains." *Journal of Applied Probability*, 42(4): 1003–1014. MR2203818. doi: https://doi.org/10.1239/jap/1134587812. 194, 198

Nolan, J. P. (2003). "Modeling financial data with stable distributions." In *Handbook of heavy tailed distributions in finance*, 105–130. Elsevier. 195

Pillai, N. S. and Smith, A. (2014). "Ergodicity of approximate MCMC chains with applications to large data sets." *arXiv preprint arXiv:1405.0182*. 211

Prangle, D. (2015). "Summary statistics in approximate Bayesian computation." *arXiv preprint arXiv:1512.05633*. MR3889282. 196

Price, L. F., Drovandi, C. C., Lee, A., and Nott, D. J. (2018). "Bayesian synthetic likelihood." *Journal of Computational and Graphical Statistics*, 27(1): 1–11. MR3788296. doi: https://doi.org/10.1080/10618600.2017.1302882. 194, 201, 202, 206

Quiroz, M., Villani, M., and Kohn, R. (2015). "Speeding up MCMC by efficient data subsampling." *Riksbank Research Paper Series*, (121). 194

Robert, C. P., Cornuet, J.-M., Marin, J.-M., and Pillai, N. S. (2011). "Lack of confidence in approximate Bayesian computation model choice." *Proceedings of the National Academy of Sciences*, 108(37): 15112–15117. 196

Roberts, G. O., Gelman, A., Gilks, W. R., et al. (1997). "Weak convergence and optimal scaling of random walk Metropolis algorithms." *The Annals of Applied Probabil-*

ity, 7(1): 110–120. MR1428751. doi: https://doi.org/10.1214/aoap/1034625254. 198

Roberts, G. O. and Rosenthal, J. S. (2001). "Optimal scaling for various Metropolis-Hastings algorithms." *Statistical Science*, 16(4): 351–367. MR1888450. doi: https://doi.org/10.1214/ss/1015346320. 198

Roberts, G. O., Rosenthal, J. S., and Schwartz, P. O. (1998). "Convergence properties of perturbed Markov chains." *Journal of applied probability*, 35(1): 1–11. MR1622440. doi: https://doi.org/10.1017/s0021900200014625. 211

Rue, H. and Held, L. (2005). *Gaussian Markov random fields: theory and applications*. Chapman and Hall/CRC. MR2130347. doi: https://doi.org/10.1201/9780203492024. 195

Schmitt, T. A., Schäfer, R., Dette, H., and Guhr, T. (2015). "Quantile correlations: Uncovering temporal dependencies in financial time series." *International Journal of Theoretical and Applied Finance*, 18(07): 1550044. MR3423181. doi: https://doi.org/10.1142/S0219024915500442. 210

Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., and McCulloch, R. E. (2016). "Bayes and big data: The consensus Monte Carlo algorithm." *International Journal of Management Science and Engineering Management*, 11(2): 78–88. 194

Sherlock, C., Golightly, A., and Henderson, D. A. (2017). "Adaptive, delayed-acceptance MCMC for targets with expensive likelihoods." *Journal of Computational and Graphical Statistics*, 26(2): 434–444. MR3640199. doi: https://doi.org/10.1080/10618600.2016.1231064. 198, 201, 217

Shestopaloff, A. Y. and Neal, R. M. (2013). "MCMC for non-linear state space models using ensembles of latent sequences." *arXiv preprint arXiv:1305.0320*. MR3564012. 195, 207

Sisson, S., Fan, Y., and Beaumont, M. (2018a). "Overview of approximate Bayesian computation." *arXiv preprint arXiv:1802.09720*. MR3939526. doi: https://doi.org/10.1146/annurev-statistics-030718-105212. 194

Sisson, S. A., Fan, Y., and Beaumont, M. (2018b). *Handbook of Approximate Bayesian Computation*. Chapman and Hall/CRC. 197

Sisson, S. A., Fan, Y., and Tanaka, M. M. (2007). "Sequential Monte Carlo without likelihoods." *Proceedings of the National Academy of Sciences*, 104(6): 1760–1765. MR2301870. doi: https://doi.org/10.1073/pnas.0607208104. 197

Wang, X. and Dunson, D. B. (2013). "Parallelizing MCMC via Weierstrass sampler." *arXiv preprint arXiv:1312.4605*. 194

Wilkinson, R. D. (2014). "Accelerating ABC methods using Gaussian processes." *arXiv preprint arXiv:1401.1436*. 198, 202

Wood, S. N. (2010). "Statistical inference for noisy nonlinear ecological dynamic systems." *Nature*, 466(7310): 1102. 194, 201, 206, 207