Biometrika (?), **99**, 1, *pp.* 1–10

Submitted to Biometrika

Printed in Great Britain

One Pseudo-Sample is Enough in Approximate Bayesian Computation MCMC

BY LUKE BORNN, NATESH PILLAI

Department of Statistics, Harvard University

bornn@stat.harvard pillai@stat.harvard.edu

AARON SMITH

Department of Mathematics and Statistics, University of Ottawa

smith.aaron.matthew@gmail.com

AND DAWN WOODARD

Department of Operations Research and Information Engineering, Cornell University

woodard@cornell.edu

SUMMARY

We analyze the efficiency of approximate Bayesian computation (ABC), which approximates the likelihood function by drawing pseudo-samples from the model. We address both the rejection sampling and Markov chain Monte Carlo versions of ABC, presenting the surprising result that multiple pseudo-samples typically do not improve the efficiency of the algorithm as compared to employing a high-variance estimate computed using a single pseudo-sample. This result means that it is unnecessary to tune the number of pseudo-samples, and is in contrast to particle MCMC methods, in which many particles are often required to provide sufficient accuracy.

Some key words: Approximate Bayesian Computation; Markov chain Monte Carlo; Pseudo-marginal

1. INTRODUCTION

Approximate Bayesian computation (ABC) refers to a family of algorithms for Bayesian inference, which address situations where the likelihood function is intractable to evaluate but where one can sample according to the model. These methods are now widely used in population genetics, systems biology, ecology, and other areas, and are implemented in an array of popular software packages. See Rubin (1984); Tavare et al. (1997), and Pritchard et al. (1999) for early examples of ABC methods, and Marin et al. (2012) for a more recent review. Let $\boldsymbol{\theta} \in \Theta$ be the parameter of interest with prior density $\pi(\boldsymbol{\theta})$, $\boldsymbol{y}_{\text{obs}} \in \mathcal{Y}$ be the observed data and $p(\boldsymbol{y}|\boldsymbol{\theta})$ be the model. The simplest version of an ABC algorithm first generates a sample $\boldsymbol{\theta'} \sim \pi(\boldsymbol{\theta})$ from the prior distribution. Next, it obtains a *pseudo-sample* $\boldsymbol{y}_{\boldsymbol{\theta'}} \sim p(\cdot|\boldsymbol{\theta'})$. Conditional on $\boldsymbol{y}_{\boldsymbol{\theta'}} = \boldsymbol{y}_{\text{obs}}$, the distribution of $\boldsymbol{\theta'}$ is the posterior distribution $\pi(\boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}}) \propto \pi(\boldsymbol{\theta})p(\boldsymbol{y}_{\text{obs}}|\boldsymbol{\theta})$. For all but the most trivial discrete problems, the probability that $\boldsymbol{y}_{\boldsymbol{\theta'}} = \boldsymbol{y}_{\text{obs}}$ is either zero or very small. Thus the above condition of exact matching of pseudo-data to the observed data is typically relaxed to $\|\eta(\boldsymbol{y}_{\text{obs}}) - \eta(\boldsymbol{y}_{\boldsymbol{\theta'}})\| < \epsilon$, where η is a low-dimensional summary statistic (for parameter estimation, a good choice for η, if possible, is a set of sufficient statistics), $\|\cdot\|$ is a distance measure, and $\epsilon > 0$ is a tolerance level. The above algorithm gives samples from the target measure π_ϵ

that is proportional to $\left[\pi(\boldsymbol{\theta}) \int \mathbf{1}_{\{\|\eta(\boldsymbol{y}_{\text{obs}})-\eta(\boldsymbol{y})\|<\epsilon\}} p(\boldsymbol{y}|\boldsymbol{\theta}) d\boldsymbol{y}\right]$. The idea is that if the tolerance ϵ is small enough and the statistic(s) η good enough, then π_ϵ is a good approximation to $\pi(\boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}})$.

A generalized version of this method (Wilkinson, 2013) is given in Algorithm 1, where $K(\eta)$ is an unnormalized probability density function, M is an arbitrary number of pseudo-samples, and c is a constant satisfying $c \geq \sup_\eta K(\eta)$.

Algorithm 1. Generalized ABC

for $t = 1$ *to* n **do**
  repeat
    Generate $\boldsymbol{\theta}'$ from the prior $\pi(\cdot)$;
    Generate $\boldsymbol{y}_{i,\boldsymbol{\theta}'}$ for $i = 1, \ldots, M$ independently from the model $p(\cdot|\boldsymbol{\theta}')$;
    Generate u uniformly on $[0, 1]$;
  until $u < \frac{1}{cM} \sum_{i=1}^{M} K(\eta(\boldsymbol{y}_{\text{obs}}) - \eta(\boldsymbol{y}_{i,\boldsymbol{\theta}'}))$;
  Set $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}'$;

Using the "uniform kernel" $K(\eta) = \mathbf{1}_{\{\|\eta\|<\epsilon\}}$ and taking $M = c = 1$ yields the version described above. Algorithm 1 yields samples $\boldsymbol{\theta}^{(t)}$ from the kernel-smoothed posterior density

$$\pi_K(\boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}}) \propto \pi(\boldsymbol{\theta}) \int K\left(\eta(\boldsymbol{y}_{\text{obs}}) - \eta(\boldsymbol{y})\right) p(\boldsymbol{y}|\boldsymbol{\theta})\, d\boldsymbol{y}. \tag{1}$$

Although Algorithm 1 is nearly always presented in the special case with $M = 1$ (Wilkinson, 2013), it is easily verified that using $M > 1$ still yields samples from $\pi_K(\boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}})$. Since our focus is on the algorithmic aspects of ABC, we will not be concerned with the choice of η and K; see instead Robert et al. (2011); Fearnhead & Prangle (2012) and Marin et al. (2014).

Many variants of Algorithm 1 exist. Algorithm 2 is a Markov chain Monte Carlo (MCMC) version, constructed using essentially any transition kernel q on Θ (Marjoram et al., 2003; Wilkinson, 2013; Andrieu & Vihola, 2014); like Algorithm 1 it uses the nonnegative, unbiased estimator

$$\hat{\pi}_{K,M}(\boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}}) \equiv \pi(\boldsymbol{\theta}) \frac{1}{M} \sum_{i=1}^{M} K\left(\eta(\boldsymbol{y}_{\text{obs}}) - \eta(\boldsymbol{y}_{i,\boldsymbol{\theta}})\right) \tag{2}$$

of (an unnormalized version of) $\pi_K(\boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}})$. Under mild conditions on K and $p(\boldsymbol{y}|\boldsymbol{\theta})$, the kernel density estimator (2) is consistent and so as $M \to \infty$, the (fixed-length) sample paths obtained from Algorithm 2 converge to those of the usual Metropolis-Hastings algorithm where $\pi_K(\boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}})$ is used for calculating the acceptance ratio instead of $\hat{\pi}_{K,M}$. Despite the approximation (2), the Markov chain converges to the distribution $\pi_K(\boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}})$ (Andrieu & Vihola, 2014).

We address the effect of the choice of M on the efficiency of Algorithms 1 and 2. Increasing the number of pseudo-samples M improves the accuracy of the estimator (2), which one might think could improve the efficiency of Algorithms 1 and 2. Indeed, this is suggested by the fact that if $\hat{\pi}_{K,M}$ is replaced by the exact target density π_K in Algorithm 2 (yielding a Metropolis-Hastings chain with target π_K), the accuracy of MCMC estimators improves (see Section 2). We show, surprisingly, that the choice $M = 1$ minimizes the running time of Algorithm 1 and yields a running time within a factor of two of optimal for Algorithm 2. We use natural definitions of running time in the contexts of serial and parallel computing, which are natural extensions of those used by other authors (see Pitt et al. (2012); Sherlock et al. (2013); Doucet et al. (2014)) and which capture the time required to obtain a particular Monte Carlo accuracy. Our definitions are appropriate when drawing pseudo-samples is more computationally intensive than the other

steps in Algorithms 1 and 2, and when the expected computational cost of drawing each pseudo-sample is the same, i.e. when there is no computational discounting due to having pre-computed relevant quantities.

Algorithm 2. ABC-MCMC

Initialize $\boldsymbol{\theta}^{(0)}$ and set $T^{(0)} = \hat{\pi}_{K,M}(\boldsymbol{\theta}^{(0)}|\boldsymbol{y}_{\text{obs}})$, as in (2);
for $t = 1$ *to* n **do**
  Generate $\boldsymbol{\theta'}$ from $q(\cdot|\boldsymbol{\theta}^{(t-1)})$;
  Generate $\boldsymbol{y}_{i,\boldsymbol{\theta'}}$ for $i = 1, \ldots, M$ independently from the model $p(\cdot|\boldsymbol{\theta'})$;
  Compute $T' = \hat{\pi}_{K,M}(\boldsymbol{\theta'}|\boldsymbol{y}_{\text{obs}})$;
  Generate u uniformly on $[0, 1]$;
  if $u \leq \frac{T' q(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta'})}{T^{(t-1)} q(\boldsymbol{\theta'}|\boldsymbol{\theta}^{(t-1)})}$ **then**
    Set $\left(\boldsymbol{\theta}^{(t)}, T^{(t)}\right) = (\boldsymbol{\theta'}, T')$;
  else
    Set $\left(\boldsymbol{\theta}^{(t)}, T^{(t)}\right) = \left(\boldsymbol{\theta}^{(t-1)}, T^{(t-1)}\right)$;

For Algorithm 1 our result (Lemma 1) is simple to show, but we are not aware of it having been pointed out. It may have been observed empirically, perhaps explaining why Algorithm 1 is nearly always described with $M = 1$. On the other hand, a number of authors have recommended choosing $M > 1$ in Algorithm 2, because this improves the accuracy of $\hat{\pi}_{K,M}(\boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}})$. Andrieu & Vihola (2014) showed that increasing M decreases the autocorrelation of the Markov chain, and improves the accuracy of the resulting Monte Carlo estimators for a fixed number of Markov chain iterations. However, increasing M also increases the running time of each iteration of the chain (if the M pseudo-samples are drawn serially), or increases the number of processors assigned to draw samples (if the pseudo-samples are drawn in parallel). It is not immediately clear how to select M to optimize this computational tradeoff, in the sense of minimizing the running time required to obtain a desired Monte Carlo accuracy. The problem is particularly hard because the accuracy of Markov chain Monte Carlo estimators depends on the amount of autocorrelation of the Markov chain, which itself depends in a complex way on the characteristics of the target distribution and the construction of the Markov chain (Woodard et al., 2009).

Several authors have drawn the conclusion that the approximately optimal value of M in Algorithm 2 is obtained by tuning M to achieve a particular variance for the estimator $\hat{\pi}_{K,M}(\boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}})$ (Pitt et al., 2012; Sherlock et al., 2013; Doucet et al., 2014). We demonstrate that this tuning process is unnecessary, since near-optimal efficiency is obtained by using low-cost, high-variance likelihood estimates based on a single pseudo-sample (Proposition 3 and Corollary 4). This result assumes only that the kernel $K(\eta)$ is the uniform kernel $\mathbf{1}_{\{\|\eta\|<\epsilon\}}$ (the most common choice).

Our result is in contrast to particle MCMC methods (Andrieu et al., 2010), where it is well known that multiple particles are often required to provide sufficient accuracy, with Flury & Shephard (2011) demonstrating that often millions of particles are required. This difference between particle MCMC and ABC-MCMC is largely due the interacting nature of the particles in the filter, allowing for better path samples.

The intuition behind our result (in the serial computation case) is that the computational effort of constructing $\hat{\pi}_{K,M}$ with $M > 1$ pseudo-samples can instead be used to propose M values of $\boldsymbol{\theta}$ and perform the accept-reject step with the less accurate estimator $\hat{\pi}_{K,1}$ constructed from a single pseudo-sample. With these M proposals, the chain using $\hat{\pi}_{K,1}$ can take multiple steps if multiple good pseudo-samples are drawn, while the chain using $\hat{\pi}_{K,M}$ can only take one step per M pseudo-samples.

We also give a simulation study that supports our theoretical results, and explores the case where pseudo-samples after the first have a reduced computational cost. Our proofs for Algorithm 2 are based on tools developed by Andrieu & Vihola (2014) that bound the relative efficiency of two *pseudo-marginal* algorithms (a class of algorithms that includes Algorithm 2). In particular, they involve comparing the distribution of the error in the estimated target density $\hat{\pi}_{K,M}$, between ABC-MCMC chains with two different values of M. We also provide an extension (Theorem 2) of the methods of Andrieu & Vihola (2014), which may be useful in characterizing other aspects of the efficiency of ABC-MCMC, or the efficiency of other types of pseudo-marginal MCMC methods.

2. EFFICIENCY OF ABC AND ABC-MCMC

For a measure μ on space $\mathcal{X}$ let $\mu(f) \equiv \int f(x)\mu(dx)$ be the expectation of a real-valued function f with respect to μ, let $L^2(\mu) = \{f : \mu(f^2) < \infty\}$ be the space of functions with finite variance, and let $\langle f, g\rangle_\mu \equiv \int f(x)g(x)\mu(dx)$ be the usual inner product for $f, g \in L^2(\mu)$. For any reversible Markov transition kernel H (as in Algorithm 2) with stationary distribution μ, any function $f \in L^2(\mu)$, and Markov chain $X^{(t)}$ evolving according to H with $X^{(0)} \sim \mu$, the Markov chain Monte Carlo estimator of $\mu(f)$ is $\bar{f}_n \equiv \frac{1}{n}\sum_{t=1}^n f(X^{(t)})$. The error of this estimator can be measured by the *normalized asymptotic variance*:

$$v(f, H) = \lim_{n\to\infty} n\mathrm{Var}_H\left(\bar{f}_n\right) \tag{3}$$

which is closely related to the autocorrelation of the samples $f(X^{(t)})$ (Tierney, 1998).

If H is *geometrically ergodic*, $v(f, H)$ is guaranteed to be finite for all $f \in L^2(\mu)$, while in the non-geometrically ergodic case $v(f, H)$ may or may not be finite (Roberts & Rosenthal, 2008). In the case of infinite asymptotic variance our results still hold but are not informative. The fact that our results do not require geometric ergodicity distinguishes them from many results on efficiency of MCMC methods (Guan & Krone, 2007; Woodard et al., 2009).

In the special case where the $X^{(t)}$ are drawn independently according to μ via an algorithm H, the quantity $n\mathrm{Var}_H\left(\bar{f}_n\right) = \langle f - \mu(f),\, f - \mu(f)\rangle_\mu$ does not depend on n or H (Robert & Casella, 2010). We denote this quantity by $v(f)$, keeping in mind that its interpretation as the error of the Monte Carlo estimator holds even for finite n.

We will characterize the running time of Algorithms 1 and 2 in two contexts: first, the case where the computations to draw the pseudo-samples are done serially, and second, the case where they are done with maximum parallelization, namely across M processors. Using (3), the variance of $\bar{f}_n$ from a single Markov chain H is roughly $v(f, H)/n$, so to achieve variance $\delta > 0$ in the serial context we need $v(f, H)/\delta$ iterations. Similarly, the variance of $\bar{f}_n$ from M parallel Markov chains H is roughly $v(f, H)/(nM)$, so to achieve variance $\delta > 0$ in the parallel context we need $v(f, H)/(\delta M)$ iterations of each Markov chain.

We assume that drawing pseudo-samples is the slowest part of the computation, and that drawing each pseudo-sample takes the same amount of time on average (as also assumed in Pitt et al. 2012, Sherlock et al. 2013, Doucet et al. 2014). Let Q_M be the transition kernel of Algorithm 2, when M samples are used to construct the estimate $\hat{\pi}_{K,M}$ at every step. Then the running time of Q_M in the serial context can be measured as the number of iterations times the number of pseudo-samples drawn in each iteration, namely $C^{\text{ser}}_{f,Q_M,\delta} \equiv Mv(f, Q_M)/\delta$. In the context of parallel computation across M processors, we compare two competing approaches that utilize all the processors. These are: (a) a single chain with transition kernel Q_M, where the $M > 1$ pseudo-samples in each iteration are drawn independently across M processors; and (b) M parallel chains with transition kernel Q_1. The running time of these methods can be measured as the num-

ber of required Markov chain iterations to obtain accuracy δ, namely $C^{\text{M-par}}_{f,Q_M,\delta} \equiv v(f,Q_M)/\delta$ utilizing method (a) and $C^{\text{M-par}}_{f,Q_1,\delta} \equiv v(f,Q_1)/(\delta M)$ utilizing method (b).

For Algorithm 1, denoted by R_M, the running time is defined analogously. However, we must account for the fact that each iteration of R_M yields one accepted value of $\boldsymbol{\theta}$, which may require multiple proposed values of $\boldsymbol{\theta}$ (along with the associated computations, including drawing pseudo-samples). The number of proposed values of $\boldsymbol{\theta}$ to get one acceptance has a geometric distribution with mean equal to the inverse of the marginal probability $p_{acc}(R_M)$ of accepting a proposed value of $\boldsymbol{\theta}$. So, similarly to Q_M, the running time of R_M in the context of serial computing can be measured as $C^{\text{ser}}_{f,R_M,\delta} \equiv Mv(f)/(\delta\, p_{acc}(R_M))$, and the computation time in the context of parallel computing can measured as $C^{\text{M-par}}_{f,R_M,\delta} \equiv v(f)/(\delta\, p_{acc}(R_M))$ utilizing method (a) and $C^{\text{M-par}}_{f,R_1,\delta} \equiv v(f)/(\delta M p_{acc}(R_1))$ utilizing method (b).

Using these definitions, we first establish that $M = 1$ is optimal for ABC.

LEMMA 1. *The marginal acceptance probability of ABC (Algorithm 1) does not depend on M. For $M > 1$ the running times $C^{ser}_{f,R_M,\delta}$ and $C^{M\text{-}par}_{f,R_M,\delta}$ of ABC in the serial and parallel contexts, respectively, satisfy*

$$C^{ser}_{f,R_M,\delta} = MC^{ser}_{f,R_1,\delta} \qquad \textit{and} \qquad C^{M\text{-}par}_{f,R_M,\delta} = MC^{M\text{-}par}_{f,R_1,\delta} \tag{4}$$

for any $f \in L^2(\pi_K)$ and any $\delta > 0$.

Proof. The marginal acceptance rate of Algorithm 1 is

$$\int \pi(\boldsymbol{\theta}) \left[\prod_{i=1}^{M} p(\boldsymbol{y}_{i,\boldsymbol{\theta}}|\boldsymbol{\theta})\right] \left[\frac{1}{cM}\sum_{i=1}^{M} K(\eta(\boldsymbol{y}_{\text{obs}}) - \eta(\boldsymbol{y}_{i,\boldsymbol{\theta}}))\right] d\boldsymbol{\theta}\, d\boldsymbol{y}_{1,\boldsymbol{\theta}} \ldots d\boldsymbol{y}_{M,\boldsymbol{\theta}}$$
$$= \frac{1}{c}\int \pi(\boldsymbol{\theta}) p(\boldsymbol{y}|\boldsymbol{\theta}) K(\eta(\boldsymbol{y}_{\text{obs}}) - \eta(\boldsymbol{y}))\, d\boldsymbol{\theta}\, d\boldsymbol{y}$$

which does not depend on M. The results for the running times follow immediately. □

Next we address ABC-MCMC. We show below that Algorithm 2 is less accurate (in terms of asymptotic variance) than the corresponding Metropolis-Hastings algorithm that uses the exact target distribution π_K. This suggests that accurately approximating the likelihood by using a large M in Algorithm 2 might lead to improved performance, as compared to using smaller values of M. In this section we conclude that this is not the case when the uniform kernel $K(\eta) = \mathbf{1}_{\{\|\eta\|<\epsilon\}}$ is used. We demonstrate that selecting $M > 1$ in Algorithm 2 is never much better than choosing $M = 1$, show that choosing $M = 1$ can be substantially better in some situations, and conclude that one should choose $M = 1$. It is of interest to extend our results to other kernels.

A potential concern regarding Algorithm 2 is raised by Lee & Latuszynski (2013), who point out that this algorithm is not geometrically ergodic when q is a local proposal distribution, such as a random walk proposal. This is due to the fact that in the tails of the distribution π_K, the pseudo-data $\boldsymbol{y}_{i,\boldsymbol{\theta}}$ are very different from $\boldsymbol{y}_{\text{obs}}$ and so the proposed moves are rarely accepted. This issue can be fixed, however, in several ways (see (Lee & Latuszynski, 2013) for a sophisticated fix).

Algorithm 2 is a special case of the pseudo-marginal algorithm (Andrieu & Roberts, 2009; Andrieu & Vihola, 2014). Pseudo-marginal algorithms are based on the fact that if one cannot evaluate the target density up to a normalizing constant (as in ABC, with target density π_K), one can substitute in an unbiased (and nonnegative) estimate of that target density into the Metropolis-Hastings acceptance ratio and the Markov chain will still have that target distribution as its stationary distribution. Thus, to sample from the distribution π_K using a Metropolis-Hastings algorithm, we may use $\hat{\pi}_{K,M}$ instead in the acceptance ratio and still have target measure $\pi_K(\boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}})$.

This is the formal justification for Algorithm 2. Recalling that Q_M is the transition kernel associated with Algorithm 2, note that Q_M is a Markov chain on the augmented state space $\Theta \times \mathbb{R}^+$ and not necessarily Markov in Θ (Andrieu & Vihola, 2014).

Our main tool in analyzing Algorithm 2 will be the results of Leskelä & Vihola (2014) and Andrieu & Vihola (2014). Two random variables X and Y are *convex ordered* if $\mathbb{E}[\phi(X)] \leq \mathbb{E}[\phi(Y)]$ for any convex function ϕ; we denote this relation by $X \leq_{cx} Y$. Let H_1, H_2 be the transition kernels of two pseudo-marginal algorithms with the same proposal kernel q and the same target marginal distribution μ on space Ω. Denote by $T_{1,x}$ and $T_{2,x}$ the (unbiased) estimators of the unnormalized target used by H_1 and H_2 respectively; these correspond to the random quantity $\hat{\pi}_{K,M}(x)$ in our Algorithm 2. Recall the normalized asymptotic variance from (3). For any function $f : \Omega \to \mathbb{R}$, let $v(f, H)$ be the normalized asymptotic variance of the function $f(x, y) \equiv f(x)$. Then if $T_{1,x} \leq_{cx} T_{2,x}$ for all x, Theorem 3 of Andrieu & Vihola (2014) shows that $v(f, H_1) \leq v(f, H_2)$ for all $f \in L^2(\mu)$.

Next we use this tool to point out that "likelihood-free" MCMC (Algorithm 2) is less accurate than the likelihood-based version, when the likelihood can be calculated directly. Let Q_∞ denote the Metropolis-Hastings chain on Θ with proposal kernel q and target π_K. Corollary 1 follows from Theorem 3 of Andrieu & Vihola (2014).

Corollary 1. For any $f \in L^2(\pi_K)$ and any $M \geq 1$ we have $v(f, Q_M) \geq v(f, Q_\infty)$. □

Proof. Q_∞ is a special case of a pseudo-marginal algorithm with target marginal distribution π_K and proposal q, where the estimate of $\pi_K(\boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}})$ is a point mass at the true value. Q_M is also a pseudo-marginal algorithm with target marginal distribution π_K and proposal q. A point mass at $\pi_K(\boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}})$ is convex upper-bounded by any random variable that has expectation $\pi_K(\boldsymbol{\theta}|\boldsymbol{y}_{\text{obs}})$, as follows from Theorem 3 of Leskelä & Vihola (2014). The result then follows from Theorem 3 of Andrieu & Vihola (2014). □

In the supplentary materials we show that this result also holds for the alternative version of ABC-MCMC described in Wilkinson (2013).

Although Corollary 1 might suggest that it is advantageous to use a large value of M in Algorithm 2, we will show that this is not the case. To do this, we first give a general result related to Theorem 3 of Andrieu & Vihola (2014). For any $0 \leq \alpha < 1$ and unbiased estimator T_x of the unnormalized target, define the estimator

$$T_{x,\alpha} = \begin{cases} 0 & \text{with probability } \alpha \\ \frac{1}{1-\alpha} T_x & \text{otherwise} \end{cases} \tag{5}$$

that is "handicapped" relative to T_x in the sense that it estimates the target density to be zero with probability α, and otherwise uses T_x (adjusted so that $T_{x,\alpha}$ is unbiased). Theorem 2 shows that convex ordering of $T_{1,x}$ and $T_{2,x,\alpha}$ is enough to obtain a relative bound on the asymptotic variances associated with the pseudo-marginal algorithms H_1 and H_2 that have common proposal q and target marginal distribution μ, and use $T_{1,x}$ and $T_{2,x}$ as estimators.

Theorem 2. Assume that H_2 is nonnegative definite. For any $0 \leq \alpha < 1$, if $T_{1,x} \leq_{cx} T_{2,x,\alpha}$ then for any $f \in L^2(\mu)$ we have

$$v(f, H_1) \leq \frac{1+\alpha}{1-\alpha} v(f, H_2).$$

Theorem 2 is proven in the supplementary materials, and assumes that the transition kernel H_2 is nonnegative definite; this is a common technical assumption in analyses of the efficiency of Markov chains (Woodard et al., 2009; Narayanan & Rakhlin, 2010), and is done for analytical

convenience. It can be achieved without affecting the relative efficiencies of H_1 and H_2, by incorporating a "holding probability" (chance of proposing to stay in the same location) of 1/2 into the proposal kernel q (Woodard et al., 2009).

Theorem 2 can be applied to the context of Algorithm 2, to obtain a useful bound for the case of the uniform kernel estimator. Proposition 3 is proven in the supplementary materials.

Proposition 3. If $K(\eta) = \mathbf{1}_{\|\eta\| \leq \epsilon}$ for some $\epsilon > 0$, and if the transition kernel Q_M of Algorithm 2 is nonnegative definite, then for any $f \in L^2(\pi_K)$ we have

$$v(f, Q_1) \leq 2M v(f, Q_M). \tag{6}$$

If additionally $\int \mathbf{1}_{\{\|\eta(\boldsymbol{y}_{\text{obs}}) - \eta(\boldsymbol{y})\| \leq \epsilon\}} p(\boldsymbol{y}|\boldsymbol{\theta}) d\boldsymbol{y} > a$ for all $\boldsymbol{\theta}$ and some $a > 0$, then

$$v(f, Q_1) \leq \left(\frac{2}{a} + 2\right) v(f, Q_M)$$

uniformly in M. □

We note that the existence of such an $a > 0$ is common whenever the state space of the chain is compact, the chain has no absorbing state and all likelihoods and distributions involved in the integral are sufficiently regular.

Proposition 3 implies that it's only possible to get an efficiency gain of two times from running Q_M rather than Q_1.

Corollary 4. Using the uniform kernel and assuming that Q_M is nonnegative definite, the running time of Q_1 is at most two times longer than that of Q_M, for both serial and parallel computation:

$$C^{\text{ser}}_{f,Q_1,\delta} \leq 2C^{\text{ser}}_{f,Q_M,\delta} \qquad \text{and} \qquad C^{\text{M-par}}_{f,Q_1,\delta} \leq 2C^{\text{M-par}}_{f,Q_M,\delta}.$$

If additionally $\int \mathbf{1}_{\{\|\eta(\boldsymbol{y}_{\text{obs}}) - \eta(\boldsymbol{y})\| \leq \epsilon\}} p(\boldsymbol{y}|\boldsymbol{\theta}) d\boldsymbol{y} > a > 0$ for all $\boldsymbol{\theta}$ then

$$C^{\text{ser}}_{f,Q_1,\delta} \leq \left(\frac{2 + 2a}{cM}\right) C^{\text{ser}}_{f,Q_M,\delta} \qquad \text{and} \qquad C^{\text{M-par}}_{f,Q_1,\delta} \leq \left(\frac{2 + 2a}{cM}\right) C^{\text{M-par}}_{f,Q_M,\delta},$$

i.e., Q_1 is order M times faster than Q_M. □

So for the uniform kernel there is never a strong reason to use Q_M over Q_1, and there can be a strong reason to use Q_1 over Q_M.

2·1. *Simulation study*

We now demonstrate these results through a simple simulation study, showing that choosing $M > 1$ is seldom beneficial. We consider the model $\boldsymbol{y}|\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\theta}, \sigma_y^2)$ for a single observation $\boldsymbol{y}$, where σ_y is known and $\boldsymbol{\theta}$ is given a standard normal prior, $\boldsymbol{\theta} \sim \mathcal{N}(0, 1)$. We apply Algorithm 2 using proposal distribution $q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t-1)}) = \mathcal{N}(\boldsymbol{\theta}; 0, 1)$, summary statistic $\eta(\boldsymbol{y}) = \boldsymbol{y}$, and $K(\eta) = \mathbf{1}_{\{\|\eta\| < \epsilon\}}$ equal to the uniform kernel with bandwidth ϵ. Each iteration of this algorithm involves simulating $\boldsymbol{\theta}'$ from $q(\cdot|\boldsymbol{\theta}^{(t-1)})$, generating pseudo-samples $\boldsymbol{y}_{1,\boldsymbol{\theta}'}, \ldots, \boldsymbol{y}_{M,\boldsymbol{\theta}'}|\boldsymbol{\theta}' \sim \mathcal{N}(\boldsymbol{\theta}', \sigma_y)$, and accepting or rejecting the move based on the approximated likelihood

$$\frac{1}{M} \sum_{i=1}^{M} \mathbf{1}_{\{\|\boldsymbol{y}_{\text{obs}} - \boldsymbol{y}_{i,\boldsymbol{\theta}'}\| < \epsilon\}}. \tag{7}$$

We start by exploring the case where $\boldsymbol{y}_{\text{obs}} = 2$ and $\sigma_y = 1$, simulating the Markov chain for 5 million iterations. Figure 1 shows the acceptance rate per generated pseudo-sample as a function

of M. As can be seen, large M does not provide any benefit in terms of accepted $\boldsymbol{\theta}$ samples per generated pseudo-sample, supporting the result of Corollary 4. In fact, for large ϵ, increasing M provides no noticeable benefit, and can even worsen the number of accepted samples per generated pseudo-sample, which agrees with our theoretical results.

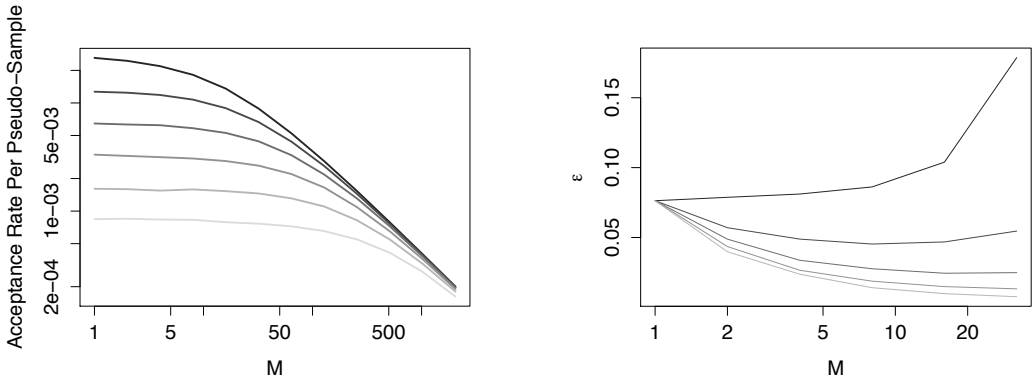

Fig. 1: Left: Acceptance rate per pseudo-sample. Lines correspond to $\epsilon = 0.5^2$ (black) through $\epsilon = 0.5^6$ (light grey), spaced uniformly on log scale. Right: Bias (ϵ) resulting from requiring that 0.4% of $\boldsymbol{\theta}$ samples are accepted per unit computational cost. Lines correspond to different computational savings for pseudo-samples beyond the first, ranging from $\delta = 1$ (black, no cost savings) to $\delta = 16$ (light grey).

In certain cases, there is an initial fixed computational cost to generating pseudo-samples, after which generating subsequent pseudo-samples is computationally inexpensive. If $\boldsymbol{y}$ is a length-n sample from a Gaussian process, for instance, there is an initial $O(n^3)$ cost to decomposing the covariance, after which each pseudo-sample may be generated at $O(n^2)$ cost. In Figure 1 we look at the ϵ which results from requiring that a fixed percentage (0.4%) of $\boldsymbol{\theta}$ samples are accepted per unit computational cost. In the non-discounted case, for example, this means that for $M = 1$ we require that 0.4% of samples are accepted, while for $M = 64$ we require that $0.4 \times 64 = 25.6\%$ of samples are accepted. In the case with discount factor $\delta > 1$ (representing a $\delta\times$ cost reduction for all pseudo-samples after the first), the pseudo-sampling cost is $(1 + (M - 1)/\delta)$, so we require that $0.4 \times (1 + (M - 1)/\delta)\%$ of $\boldsymbol{\theta}$ samples are accepted. For example, a discount of $\delta = 16$ with $M = 64$ requires that 2.0% of samples are accepted. Figure 1 shows that, for $\delta = 1$, larger M results in larger ϵ; in other words, for a fixed computational budget and no discount, $M = 1$ gives the smallest bias. For discounts $\delta > 1$, however, increasing M can indeed lead to reduced bias. See the supplementary materials for further simulations exploring discounting.

3. DISCUSSION

We have shown that when the likelihood is available, one should use standard MCMC methods rather than resorting to ABC. While likely not surprising to practitioners, this is the first such known rigorous result to our knowledge. Further, for the uniform kernel, we have shown that there is very little benefit in attempting to accurately approximate the likelihood by simulating more than one pseudo-sample per ABC iteration. Theoretical results suggest that using one sample is never significantly worse than using more, and our simulation results suggest that increasing M has a detrimental effect. In contrast to other pseudo-marginal methods, in particular particle MCMC (Andrieu et al., 2010) where M has an analogy to the number of particles, it is interesting here that the gains from increasing M do not significantly outweigh the added

computational cost. One exception is when additional samples have a reduced computational cost, and the ABC bandwidth is small relative to the likelihood variance. Extending our result to non-uniform kernels is an interesting open problem.

Our results are obtained by bounding the asymptotic variance of Monte Carlo estimates. For Algorithm 1, this is equivalent to bounding the finite-sample Monte Carlo variance. For Algorithm 2, our results take into account the autocorrelation of the Markov chain, and its effect on the asymptotic Monte Carlo variance. On the other hand, our results for Algorithm 2 do not take into account differences between the finite-sample variance and the asymptotic variance, such as the effect of the burn-in period. However, in practice there is typically no burn-in period for Algorithm 2, since it is usually initialized using samples from ABC (Marin et al., 2012).

REFERENCES

ANDRIEU, C., DOUCET, A. & HOLENSTEIN, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B* **72**, 269–342.

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

ANDRIEU, C. & VIHOLA, M. (2014). Establishing some order amongst exact approximations of MCMCs. *arXiv preprint, arXiv:1404.6909v1* .

DOUCET, A., PITT, M., DELIGIANNIDIS, G. & KOHN, R. (2014). Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *arXiv preprint, arXiv:1210.1871v3* .

FEARNHEAD, P. & PRANGLE, D. (2012). Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation. *Journal of the Royal Statistical Society: Series B* **74**, 419–474.

FLURY, T. & SHEPHARD, N. (2011). Bayesian inference based only on simulated likelihood: particle filter analysis of dynamic economic models. *Econometric Theory* **27**, 933–956.

GUAN, Y. & KRONE, S. M. (2007). Small-world MCMC and convergence to multi-modal distributions: from slow mixing to fast mixing. *Annals of Applied Probability* **17**, 284–304.

LEE, A. & LATUSZYNSKI, K. (2013). Variance bounding and geometric ergodicity of Markov chain Monte Carlo kernels for approximate Bayesian computation. *arXiv preprint arXiv:1210.6703* .

LESKELÄ, L. & VIHOLA, M. (2014). Conditional convex orders and measurable martingale couplings. *arXiv preprint, arXiv:1404.0999* .

MARIN, J.-M., PILLAI, N. S., ROBERT, C. P. & ROUSSEAU, J. (2014). Relevant statistics for Bayesian model choice. *Journal of the Royal Statistical Society: Series B* In press.

MARIN, J.-M., PUDLO, P., ROBERT, C. P. & RYDER, R. J. (2012). Approximate Bayesian computational methods. *Statistics and Computing* **22**, 1167–1180.

MARJORAM, P., MOLITOR, J., PLAGNOL, V. & TAVARÉ, S. (2003). Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences* **100**, 15324–15328.

NARAYANAN, H. & RAKHLIN, A. (2010). Random walk approach to regret minimization. In *Advances in Neural Information Processing Systems*.

PITT, M. K., SILVA, R. D. S., GIORDANI, P. & KOHN, R. (2012). On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. *Journal of Econometrics* **171**, 134–151.

PRITCHARD, J. K., SEIELSTAD, M. T., PEREZ-LEZAUN, A. & FELDMAN, M. W. (1999). Population growth of human Y chromosomes: a study of Y chromosome microsatellites. *Molecular Biology and Evolution* **16**, 1791–1798.

ROBERT, C. & CASELLA, G. (2010). *Monte Carlo Statistical Methods*. New York: Springer-Verlag, 2nd ed.

ROBERT, C. P., CORNUET, J.-M., MARIN, J.-M. & PILLAI, N. S. (2011). Lack of confidence in approximate Bayesian computation model choice. *Proceedings of the National Academy of Sciences* **108**, 15112–15117.

ROBERTS, G. O. & ROSENTHAL, J. S. (2008). Variance bounding Markov chains. *Annals of Applied Probability* **18**, 1201–1214.

RUBIN, D. B. (1984). Bayesianly justifiable and relevant frequency calculations for the applied statistician. *Annals of Statistics* **12**, 1151–1172.

SHERLOCK, C., THIERY, A. H., ROBERTS, G. O. & ROSENTHAL, J. S. (2013). On the efficiency of pseudo-marginal random walk Metropolis algorithms. *arXiv preprint arXiv:1309.7209* .

TAVARE, S., BALDING, D. J., GRIFFITHS, R. & DONNELLY, P. (1997). Inferring coalescence times from DNA sequence data. *Genetics* **145**, 505–518.

TIERNEY, L. (1998). A note on Metropolis-Hastings kernels for general state spaces. *Annals of Applied Probability* **8**, 1–9.

WILKINSON, R. D. (2013). Approximate Bayesian computation (ABC) gives exact results under the assumption of model error. *Statistical applications in genetics and molecular biology* **12**, 129–141.

WOODARD, D. B., SCHMIDLER, S. C. & HUBER, M. (2009). Conditions for rapid mixing of parallel and simulated tempering on multimodal distributions. *Annals of Applied Probability* **19**, 617–640.

[*Received April* 2012. *Revised September* 2012]