Proceedings of the 2012 Winter Simulation Conference
C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, eds.

ON THE CHOICE OF MCMC KERNELS FOR APPROXIMATE BAYESIAN COMPUTATION WITH SMC SAMPLERS

Anthony Lee

University of Warwick
Coventry CV4 7AL
UNITED KINGDOM

ABSTRACT

Approximate Bayesian computation (ABC) is a class of simulation-based statistical inference procedures that are increasingly being applied in scenarios where the likelihood function is either analytically unavailable or computationally prohibitive. These methods use, in a principled manner, simulations of the output of a parametrized system in lieu of computing the likelihood to perform parametric Bayesian inference. Such methods have wide applicability when the data generating mechanism can be simulated. While approximate, they can usually be made arbitrarily accurate at the cost of computational resources. In fact, computational issues are central to the successful use of ABC in practice. We focus here on the use of sequential Monte Carlo samplers for ABC and in particular on the choice of Markov chain Monte Carlo kernels used to drive their performance, investigating the use of kernels whose mixing properties are less sensitive to the quality of the approximation than standard kernels.

1 INTRODUCTION

In parametric Bayesian inference, one associates with some data $y \in \mathbb{R}^d$ a likelihood function f that is the probability density function of the data y conditional upon the value of an unknown parameter $\theta \in \Theta$. In order to make probabilistic assertions about the possible values of θ, it is modeled as a random variable with a prior distribution defined by a density p such that the joint density of θ and y can be written $p(\theta, y) = p(\theta) f(y|\theta)$. The conditional or posterior density of θ given the observed data y is then given by Bayes' rule

$$p(\theta|y) = \frac{p(\theta) f(y|\theta)}{\int p(\theta') f(y|\theta') d\theta'}. \tag{1}$$

Many items of interest in Bayesian inference are posterior expectations of functions of θ, i.e.

$$\int \phi(\theta) p(\theta|y) d\theta, \tag{2}$$

for some function ϕ. One approach to estimating such expectations when they are analytically unavailable is by using Monte Carlo methods, which approximate (2) using random variables. In general, such methods require the ability to compute $p(\theta|y)$ point-wise up to a normalizing constant, i.e. one can compute the numerator in (1) but not necessarily the denominator.

1.1 Approximate Bayesian Computation

Approximate Bayesian computation (ABC) is largely concerned with the scenario where f cannot be computed or is too expensive to compute but where one is able to simulate, for any $\theta \in \Theta$, from the distribution with density $f(\cdot|\theta)$ (Tavaré et al. 1997; Pritchard et al. 1999). In such cases, not being able to evaluate $p(\theta|y)$ up to a normalizing constant renders many general-purpose Monte Carlo schemes

inapplicable but one can perform Bayesian inference using an artificial likelihood f^ε in lieu of the original likelihood f where $\varepsilon > 0$ determines the quality of the approximation. This likelihood is of the form

$$f^\varepsilon(y|\theta) = \int f(x|\theta)\xi^\varepsilon(y|x)dx,$$

where $\xi^\varepsilon(\cdot|x)$ is a probability density function with a large concentration of mass near x. For the entirety of this paper we consider the choice

$$\xi^\varepsilon(y|x) = \mathrm{Vol}(\varepsilon)^{-1}\mathbf{1}_{B_\varepsilon(x)}(y),$$

where $B_\varepsilon(x)$ denotes a metric ball of radius ε and $\mathrm{Vol}(\varepsilon)$ is the volume of a metric ball of radius ε. This choice can be interpreted as $f^\varepsilon(y|\theta) = \mathrm{Vol}(\varepsilon)^{-1}\int_{B_\varepsilon(y)} f(x|\theta)dx$, which closely approximates $f(y|\theta)$ for small ε, under appropriate assumptions on the smoothness of $f(x|\theta)$ when x is close to y. The quantity $\int_{B_\varepsilon(y)} f(x|\theta)dx$ can be thought of as the probability of "hitting" the ball $B_\varepsilon(y)$ when sampling from the likelihood function with parameter θ, and we refer to this loosely as "hitting the data" y with the dependence on ε implicit. Because we are using an approximation to the likelihood, f^ε, the approximate posterior is given by

$$p^\varepsilon(\theta|y) = \frac{p(\theta)f^\varepsilon(y|\theta)}{\int p(\theta')f^\varepsilon(y|\theta')d\theta'}.$$

While $f^\varepsilon(y|\theta)$ also cannot be computed point-wise, there exist many simulation-based methods for estimating expectations of the form

$$\int \phi(\theta)p^\varepsilon(\theta|y)d\theta, \tag{3}$$

some of which are presented in Sections 1.2 -1.4. A more comprehensive, recent survey of developments in ABC can be found in Marin et al. (2012).

In cases where the data obtained, $\tilde{y}$, is high-dimensional, it is often computationally advantageous to summarize the data using a summary statistic $S(\tilde{y})$ that is of lower dimension than $\tilde{y}$. We omit the explicit use of summary statistics here, noting that this is equivalent to performing inference when only the summary $y = S(\tilde{y})$ is the observed data and we utilize auxiliary variables $x = S(\tilde{x})$ in approximating the posterior. While summarization in this manner makes inference more computationally tractable, it is inevitably associated with an additional loss of information except when the summary statistic is sufficient for the data. As an example, consider the situation where one has m i.i.d. observations $\tilde{y}_{1:m}$with likelihood $\tilde{f}$. In an ABC approximation we can use $y = S(\tilde{y}_{1:m})$ as observed data, discarding all other information contained in $\tilde{y}_{1:m}$. Then $f(y|\theta) = \int \delta_{S(\tilde{x}_{1:m})}(y)\prod_{i=1}^m \tilde{f}(\tilde{x}_i|\theta)d\tilde{x}_{1:m}$ is the associated likelihood of the summary statistic y with parameter θ, and we can always simulate according to f if we can simulate according to $\tilde{f}$ and we can compute $S(\cdot)$. It is worth noting that some choices of S may render some components of θ unidentifiable even if those components are identifiable under the true posterior involving $\tilde{f}$.

1.2 Simple Monte Carlo Methods for ABC

The ability to simulate according to f allows one to use a variety of Monte Carlo methods to approximate (3). The key behind such approximations is noting that the utilization of $f(\cdot|\theta)$ as a proposal density in a Monte Carlo scheme allows one to bypass the computation of both $f(y|\theta)$ and $f^\varepsilon(y|\theta)$. This involves defining as a target density $\pi^\varepsilon(\theta,x) = p^\varepsilon(\theta,x|y) \propto p(\theta)f(x|\theta)\xi^\varepsilon(y|x)$, the posterior density of θ and the auxiliary variable x where it should be clear that $\int p^\varepsilon(\theta,x|y)dx = p^\varepsilon(\theta|y)$.

One can obtain an estimate of (3) by using self-normalized importance sampling. Here, one defines a proposal density q^ε with $\pi^\varepsilon(\theta,x) > 0 \implies q^\varepsilon(\theta,x) > 0$ and proceeds to sample $(\theta^{(i)},x^{(i)}) \sim q^\varepsilon(\cdot)$ i.i.d.

for $i \in \{1,\ldots,n\}$ and compute the weighted average of evaluations of ϕ:

$$\hat{I}_{IS} = \left[\sum_{j=1}^{n} w(\theta^{(j)},x^{(j)})\right]^{-1} \sum_{i=1}^{n} \phi(\theta^{(i)}) w(\theta^{(i)},x^{(i)}),$$

where $w(\theta,x) = p(\theta)f(x|\theta)\xi^{\varepsilon}(y|x)/q^{\varepsilon}(\theta,x)$ is the importance weight associated with (θ,x). For the special case where $q^{\varepsilon}(\theta,x) = g(\theta)f^{\varepsilon}(x|\theta)$, the weights are of the form $w(\theta,x) = p(\theta)\xi^{\varepsilon}(y|x)/g(\theta)$, which does not require the computation of $f(x|\theta)$. Rejection sampling according to π^{ε} is also possible using the same type of proposal density q^{ε} and can be used to compute a classical Monte Carlo estimate

$$\hat{I}_{MC} = \frac{1}{n}\sum_{i=1}^{n} \phi(\theta^{(i)}),$$

where $\theta^{(i)} \sim \pi^{\varepsilon}(\cdot)$ for $i \in \{1,\ldots,n\}$. In this case, one samples $(\theta,x) \sim q^{\varepsilon}(\cdot)$ and "accepts" it with probability $\frac{p(\theta)\xi^{\varepsilon}(y|x)}{Cg(\theta)}$, with $C \geq \sup_{(\theta,x)}\left\{\frac{p(\theta)\xi^{\varepsilon}(y|x)}{g(\theta)}\right\}$. Each accepted sample is distributed exactly according to π^{ε}. In the special case where $g = p$, one can accept (θ,x) if $\mathbf{1}_{B_{\varepsilon}(x)}(y) = 1$, i.e. we have "hit" the data.

While such estimates are theoretically justified, they can in practice require too much computational effort to obtain estimates that are sufficiently close to (3) for use in inference. In particular, since $\mathrm{Vol}(\varepsilon)$ decreases as $O(\varepsilon^d)$, the probability of hitting the data can be very low so that most of the importance weights are 0. While this cannot be circumvented in general, more advanced approaches that utilize Markov chain Monte Carlo (MCMC) and/or sequential Monte Carlo (SMC) try to ensure that the algorithm used to estimate (3) spends more time in regions of Θ where the probability of hitting the data is high.

1.3 Markov Chain Monte Carlo

An MCMC approach involves simulating a Markov chain $(\theta^{(1)},x^{(1)}),(\theta^{(2)},x^{(2)}),\ldots,(\theta^{(m)},x^{(m)})$ for m steps with equilibrium distribution π^{ε} from some initial point $(\theta^{(0)},x^{(0)})$ and estimating (3) using the time average

$$\frac{1}{m-b}\sum_{i=1}^{m-b} \phi(\theta^{(i)}), \tag{4}$$

for some fixed "burn-in" $b < m$. Such a Markov chain can be simulated using a Metropolis-Hastings (Metropolis et al. (1953); Hastings (1970)) kernel K_{ε} that uses a proposal density $q^{\varepsilon}((\theta',z)|(\theta,x)) = g(\theta'|\theta)f(z|\theta')$. Algorithm 1 describes how to sample from the kernel $K_{\varepsilon}((\theta,x),\cdot)$. This kernel was proposed in Marjoram et al. (2003).

Algorithm 1 Metropolis-Hastings ABC kernel

At (θ,x):

1. Sample $\theta' \sim g(\cdot|\theta)$.
2. Sample $z \sim f(\cdot|\theta')$.
3. With probability
$$\min\left\{1, \frac{p(\theta')g(\theta|\theta')}{p(\theta)g(\theta'|\theta)}\mathbf{1}_{B_{\varepsilon}(z)}(y)\right\},$$
output (θ',z). Otherwise, output (θ,x).

This simple MCMC kernel does have the property that more computational effort is spent in regions of Θ with high mass under π^{ε}. However, its use is somewhat hindered by the fact that ε needs to be small

to control the closeness of the artificial likelihood f^{ε} to the true likelihood f and small ε implies that the probability of hitting the data is also small. In practice, this manifests itself in the chain getting "stuck" for many iterations, i.e. for large values of k and some i, $\theta^{(i)} = \theta^{(i+1)} = \cdots = \theta^{(i+k)}$ and the estimate (4) can be unacceptably far from (3) for reasonable values of m.

1.4 Sequential Monte Carlo

One approach to alleviating the issue with a direct MCMC approach is to utilize a sequential Monte Carlo (SMC) approach, which gradually decreases ε from some large value ε_1 to some ε_T that has been deemed small enough for the approximate posterior π^{ε_T} to be sufficiently close to the exact posterior. This can be done using a sequential Monte Carlo sampler (Del Moral, Doucet, and Jasra 2006) as in Sisson, Fan, and Tanaka (2007), Beaumont et al. (2009), Del Moral, Doucet, and Jasra (2012). The first two can also be seen as population Monte Carlo (PMC) approaches (Cappé et al. 2004). In both cases, one defines the decreasing sequence $\{\varepsilon_t\}_{t=1}^{T}$ where ε_1 is large and ε_T is the final value of ε. This in turn defines a sequence of posterior densities $\pi_t = \pi^{\varepsilon_t}$ for $t \in \{1,\ldots,T\}$, which are approximated in turn by the sampler using weighted particles. The distinct SMC methods that are commonly used differ in how the particles at each time are propagated. One special case of the SMC sampler methodology allows one to use a π_t-invariant MCMC kernel to "move" each particle whilst in the PMC approach, the particles are often sampled using a rejection sampler. The former is described in Algorithm 2, which is in the spirit of Del Moral, Doucet, and Jasra (2012) and the latter in Algorithm 3, which was proposed in Beaumont et al. (2009) (see also Toni et al. (2009) for a nearly identical method without the adaptive parameters $\tau_{2:T}$).

Algorithm 2 SMC Sampler with MCMC kernels

1. At $t = 0$:
   (a) For $i \in \{1,\ldots,n\}$, sample $(\theta_0^{(i)}, x_0^{(i)}) \sim \pi_1(\cdot)$ i.i.d. (using rejection).
2. At $t \in \{1,\ldots,T\}$:
   (a) For $i \in \{1,\ldots,n\}$, set $w_{t-1}^{(i)} = \mathbf{1}_{B_{\varepsilon_t}(x_{t-1}^{(i)})}(y)$.
   (b) Select n indices $A_{t-1}^{(1:n)}$ according to the weights $w_{t-1}^{(1:n)}$ such that for each $i \in \{1,\ldots,n\}$,
   $$\mathbb{E}\left[\sum_{j=1}^{n} \mathbb{I}[A_{t-1}^{(j)} = i]\right] = \frac{n w_{t-1}^{(i)}}{\sum_{j=1}^{n} w_{t-1}^{(j)}}.$$
   (c) For $i \in \{1,\ldots,n\}$, sample $(\theta_t^{(i)}, x_t^{(i)}) \sim K_{\varepsilon_t}((\theta_{t-1}^{(A_{t-1}^{(i)})}, x_{t-1}^{(A_{t-1}^{(i)})}), \cdot)$.
3. Compute $\hat{I}_{SMC_1} = \frac{1}{n}\sum_{i=1}^{n} \phi(\theta_T^{(i)})$.

The resampling step, 2(b), in Algorithm 2 can be interpreted as requiring that the expected number of time t offspring of particle i at time $t-1$, $(\theta_{t-1}^{(i)}, x_{t-1}^{(i)})$, is $\frac{n w_{t-1}^{(i)}}{\sum_{j=1}^{n} w_{t-1}^{(j)}}$ for every $i \in \{1,\ldots,n\}$. A variety of schemes satisfy this requirement (see, e.g., Douc, Cappé, and Moulines (2005)), and residual resampling is used for the examples in Section 3. Note that in Algorithm 3 one can sample according to the density

$$q(\theta'|\theta^{(1:n)}, w^{(1:n)}, \varepsilon, \tau) \propto f^{\varepsilon}(y|\theta') \sum_{i=1}^{n} w^{(i)} g_{\tau}(\theta'|\theta^{(i)})$$

via rejection: one first samples $\theta' \sim \sum_{i=1}^{n} w^{(i)} g_{\tau}(\cdot|\theta^{(i)})$ and an auxiliary $x \sim f(\cdot|\theta')$ and accepts if $y \in B_{\varepsilon}(x)$. One can think of $\tilde{p}(\theta'; \theta^{(1:n)}, w^{(1:n)}, \tau) = \sum_{i=1}^{n} w^{(i)} g_{\tau}(\theta'|\theta^{(i)})$ as an artificial prior and so the importance

weights correct for the discrepancy between the true prior p and the artificial prior $\tilde{p}$. The choice of τ_t in this scheme is used to try to tailor the proposal density q using information contained in the samples up to time $t-1$. In the remainder of this paper, SMC sampler will generally refer to Algorithm 2 and PMC sampler will be used to refer to Algorithm 3.

Algorithm 3 SMC using a rejection proposal

1. At $t = 1$
   (a) For $i \in \{1,\ldots,n\}$, sample $\theta_0^{(i)} \sim \pi_1(\cdot)$ i.i.d. (using rejection).
   (b) For $i \in \{1,\ldots,n\}$, set $w_0^{(i)} = \frac{1}{n}$.
2. At $t \in \{2,\ldots,T\}$:
   (a) Let τ_t be some function of $(\theta_{t-1}^{(1:n)}, w_{t-1}^{(1:n)})$.
   (b) For $i \in \{1,\ldots,n\}$, sample $\theta_t^{(i)} \sim q(\cdot|\theta_{t-1}^{(1:n)}, w_{t-1}^{(1:n)}, \varepsilon_t, \tau_t)$ where

$$q(\theta'|\theta^{(1:n)}, w^{(1:n)}, \varepsilon, \tau) = \frac{f^{\varepsilon}(y|\theta')\sum_{i=1}^{n} w^{(i)} g_{\tau}(\theta'|\theta^{(i)})}{\int f^{\varepsilon}(y|\bar{\theta})\sum_{i=1}^{n} w^{(i)} g_{\tau}(\bar{\theta}|\theta^{(i)})d\bar{\theta}}.$$

   (c) For $i \in \{1,\ldots,n\}$, set $\tilde{w}_t^{(i)} = \frac{p(\theta_t^{(i)})}{\sum_{j=1}^{n} w_{t-1}^{(j)} g_{\tau_t}(\theta_t^{(i)}|\theta_{t-1}^{(j)})}$.
   (d) For $i \in \{1,\ldots,n\}$, set $w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^{n} \tilde{w}_t^{(j)}}$.
3. Compute $\hat{I}_{SMC_2} = \sum_{i=1}^{n} \phi(\theta_T^{(i)}) w_T^{(i)}$.

The two approaches are quite different, despite both having a sequential structure targeting the same sequence of auxiliary distributions. The main advantage of Algorithm 2 is that it has $O(nT)$ cost, while its main disadvantage is that typically the MCMC kernels K_{ε_t} mix more and more slowly as t increases, due to the difficulty of hitting the data as ε decreases. On the other hand, the main advantage of Algorithm 3 is that by utilizing rejection, one can be confident that the particles $\theta_t^{(1:n)}$ are adequately approximating π_t as long as the proposal density $q(\theta'|\theta_{t-1}^{(1:n)}, w_{t-1}^{(1:n)}, \varepsilon_t, \tau_t)$ has good enough coverage while its main disadvantages are that the cost of obtaining a sample from $q(\cdot|\theta_{t-1}^{(1:n)}, w_{t-1}^{(1:n)}, \varepsilon_t, \tau_t)$ increases as ε decreases, that step $2(b)$ is an $O(n^2)$ operation and that the weights at time T can have a large variance.

The focus of this paper is to highlight the possibility of using different types of MCMC kernel K_{ε} to drive Algorithm 2, with the particular aim of making the mixing properties of the kernel be nearly independent of ε as $\varepsilon \to 0$.

2 ROBUST MCMC KERNELS FOR ABC

We investigate the use of "adaptive" MCMC kernels that have more robust properties as $\varepsilon \to 0$. A particular motivation is that the analysis of SMC samplers to date often assumes the use of geometrically ergodic MCMC kernels with similar ergodic behavior (see, e.g., Jasra and Doucet (2008); Whiteley (2012)), i.e. they mix well at every $t \in \{1,\ldots,T\}$, although this assumption is not typically satisfied in practice and is not met in most cases for ABC SMC samplers as $\varepsilon \to 0$ with a simple MCMC kernel. Schweizer (2012) has sought to weaken this assumption by considering kernels that only mix well "locally" but this property again is unlikely to hold in most interesting ABC applications. This encourages the construction of MCMC kernels that do have similar behavior as $\varepsilon \to 0$, although it does not imply that this is necessary for the SMC methods to provide good estimates.

2.1 A 1-hit Kernel

We first propose to use an MCMC kernel presented in Lee, Andrieu, and Doucet (2012b) and based on Lee, Andrieu, and Doucet (2012a), the 1-hit ABC-MCMC kernel, which can be sampled from using Algorithm 4. The robustness of this kernel comes from the fact that one simulates according to the likelihood for both the current value of θ and the proposed value θ' until a hit is observed, which allows the Markov chain to have a reasonable chance of moving if the proposed value θ' has high posterior density. This is in contrast to the kernel in Algorithm 1, where as $\varepsilon \to 0$, the probability of obtaining a hit, and hence accepting θ', decreases rapidly. In fact, it can be shown for any $\varepsilon > 0$ the probability of, at θ, accepting the proposal θ' using Algorithm 4 can be expressed analytically as

$$\min\left\{1, \frac{p(\theta')g(\theta|\theta')}{p(\theta)g(\theta'|\theta)}\right\} \times \frac{f^\varepsilon(y|\theta')}{f^\varepsilon(y|\theta) + f^\varepsilon(y|\theta') - \mathrm{Vol}(\varepsilon) f^\varepsilon(y|\theta) f^\varepsilon(y|\theta')}.$$

Both the analytic probability of accepting a proposed move and that the 1-hit kernel satisfies detailed balance can be verified directly by noting that N is a geometric random variable.

Algorithm 4 1-hit MCMC kernel

At (θ, x):

1. Sample $\theta' \sim g(\cdot|\theta)$.
2. With probability $1 - \min\left\{1, \frac{p(\theta')g(\theta|\theta')}{p(\theta)g(\theta'|\theta)}\right\}$, output (θ, x). Otherwise,
   (a) For $i = 1, 2, \ldots$ sample $z^{(i)} \sim f(\cdot|\theta')$ and $x^{(i)} \sim f(\cdot|\theta)$ until $y \in B_\varepsilon(z^{(i)})$ and/or $y \in B_\varepsilon(x^{(i)})$. Let $N = i$.
   (b) If $y \in B_\varepsilon(z^{(N)})$ output $(\theta', z^{(N)})$. Otherwise, output (θ, x).

Proposition 1. *The 1-hit kernel satisfies detailed balance.*

Proof. The kernel can be viewed as, at (θ, x), proposing $\theta' \sim g(\cdot|\theta)$ and $z \sim p^\varepsilon(\cdot|y, \theta')$ with $p^\varepsilon(z|y, \theta') = \frac{f(z|\theta')\mathbf{1}_{B_\varepsilon(z)}(y)\mathrm{Vol}(\varepsilon)^{-1}}{f^\varepsilon(y|\theta')}$, and accepting with probability $\min\left\{1, \frac{p(\theta')g(\theta|\theta')}{p(\theta)g(\theta'|\theta)}\right\} \mathbb{P}[N_{\theta'} \leq N_\theta]$ where $N_{\theta'}$ and N_θ are independent geometric random variables with success probability $\mathrm{Vol}(\varepsilon) f^\varepsilon(y|\theta)$ and $\mathrm{Vol}(\varepsilon) f^\varepsilon(y|\theta')$ respectively. Their minimum is also a geometric random variable and we have

$$\mathbb{P}[N_{\theta'} \leq N_\theta] = \frac{f^\varepsilon(y|\theta')}{f^\varepsilon(y|\theta) + f^\varepsilon(y|\theta') - \mathrm{Vol}(\varepsilon) f^\varepsilon(y|\theta) f^\varepsilon(y|\theta')}.$$

It can then be verified that detailed balance holds, i.e.

$$\begin{aligned}\pi^\varepsilon(\theta, x) g(\theta'|\theta) p^\varepsilon(z|y, \theta') \min\left\{1, \frac{p(\theta')g(\theta|\theta')}{p(\theta)g(\theta'|\theta)}\right\} \mathbb{P}[N_{\theta'} \leq N_\theta] \\ = \pi^\varepsilon(\theta', z) g(\theta|\theta') p^\varepsilon(x|y, \theta) \min\left\{1, \frac{p(\theta)g(\theta'|\theta)}{p(\theta')g(\theta|\theta')}\right\} \mathbb{P}[N_\theta \leq N_{\theta'}].\end{aligned}$$

□

2.2 Two r-hit Kernels

An alternative to the 1-hit kernel is to sample auxiliary data associated with θ and θ' until $r-1$ and r hits have been obtained respectively for some $r \in \mathbb{Z}$ with $r \geq 2$. Algorithm 5 describes such a kernel, which may

be beneficial in circumstances when the term $\frac{p(\theta')g(\theta|\theta')}{p(\theta)g(\theta'|\theta)}$ varies considerably. In this case the discrepancy between N and N' can, e.g., allow one to accept with higher probability a proposed θ' that has a much higher likelihood than θ but a lower value of $p(\theta')g(\theta|\theta')$. However, it can be shown that if $\frac{p(\theta')g(\theta|\theta')}{p(\theta)g(\theta'|\theta)}$ is always equal to 1, then the 2-hit kernel and the 1-hit kernel are identical, the algorithms presenting only different ways of sampling from the same kernel. Since the 2-hit algorithm is more expensive computationally, the 1-hit algorithm is to be preferred in such situations. The r-hit kernel can be justified by noting that both N and N' are negative binomial random variables.

Algorithm 5 r-hit MCMC kernel ($r \geq 2$)

At (θ, x):

1. Sample $\theta' \sim g(\cdot|\theta)$.
2. For $i = 1, 2, \ldots$ sample $z^{(i)} \sim f(\cdot|\theta')$ until $\sum_{j=1}^{i} \mathbf{1}_{B_\varepsilon(z^{(i)})}(y) = r$. Let $N' = i$.
3. Sample L uniformly from the set $\{j \in \{1, \ldots, N'-1\} : \mathbf{1}_{B_\varepsilon(z^{(j)})}(y) = 1\}$.
4. For $i = 1, 2, \ldots$ sample and $x^{(i)} \sim f(\cdot|\theta)$ until $\sum_{j=1}^{i} \mathbf{1}_{B_\varepsilon(x^{(i)})}(y) = r-1$. Let $N = i$.
5. With probability
$$\min\left\{1, \frac{p(\theta')g(\theta|\theta')}{p(\theta)g(\theta'|\theta)} \times \frac{N}{N'-1}\right\},$$
output $(\theta', z^{(L)})$. Otherwise output (θ, x).

Proposition 2. *The r-hit kernel satisfies detailed balance.*

Proof. The kernel can be viewed as, at (θ, x), proposing $\theta' \sim g(\cdot|\theta)$ and $z \sim p^\varepsilon(\cdot|y, \theta')$ and accepting with probability $\mathbb{E}\left[\min\left\{1, \frac{p(\theta')g(\theta|\theta')}{p(\theta)g(\theta'|\theta)} \times \frac{N_{r-1,\theta}}{N_{r,\theta'}-1}\right\}\right]$ where $N_{r-1,\theta}$ and $N_{r,\theta'}$ are negative binomial random variables associated with the number of trials required to obtain $r-1$ and r successes, respectively, with success parameters $\mathrm{Vol}(\varepsilon) f^\varepsilon(y|\theta)$ and $\mathrm{Vol}(\varepsilon) f^\varepsilon(y|\theta')$. One can verify that detailed balance holds, i.e.

$$\pi^\varepsilon(\theta, x) g(\theta'|\theta) p^\varepsilon(z|y, \theta') \mathbb{E}\left[\min\left\{1, \frac{p(\theta')g(\theta|\theta')}{p(\theta)g(\theta'|\theta)} \times \frac{N_{r-1,\theta}}{N_{r,\theta'}-1}\right\}\right]$$
$$= \pi^\varepsilon(\theta', z) g(\theta|\theta') p^\varepsilon(x|y, \theta) \mathbb{E}\left[\min\left\{1, \frac{p(\theta)g(\theta'|\theta)}{p(\theta')g(\theta|\theta')} \times \frac{N_{r-1,\theta'}}{N_{r,\theta}-1}\right\}\right].$$

□

A possible issue with both the 1-hit and the r-hit kernels is that they rely on the proposal density g being reasonably good. An alternative kernel is proposed in Algorithm 6, which may be better suited to problems in which it is difficult to construct a good proposal density g, since one is not restricted to considering only one possible value of θ'. This kernel also requires $r \geq 2$.

Proposition 3. *The r-hit kernel with multiple θ proposals satisfies detailed balance.*

Proof. The kernel can be viewed as, at (θ, x), proposing $(\theta', z) \sim \tilde{q}^\varepsilon(\cdot|y, \theta)$ where

$$\tilde{q}^\varepsilon(\theta', z|y, \theta) = \frac{g(\theta'|\theta) p^\varepsilon(z|y, \theta')}{\int g(\bar{\theta}|\theta) f^\varepsilon(y|\bar{\theta}) d\bar{\theta}},$$

and accepting with probability $\mathbb{E}\left[\min\left\{1, \frac{p(\theta')g(\theta|\theta')}{p(\theta)g(\theta'|\theta)} \times \frac{M_{r-1,\theta'}}{M_{r,\theta}-1}\right\}\right]$ where $M_{r-1,\theta'}$ and $M_{r,\theta}$ are negative binomial random variables associated with the number of trials required to obtain $r-1$ and r successes,

Algorithm 6 r-hit MCMC kernel with multiple θ proposals ($r \geq 2$)

At (θ, x):

1. For $i = 1, 2, \ldots$ sample $\theta'^{(i)} \sim g(\cdot|\theta)$ and $z^{(i)} \sim f(\cdot|\theta'^{(i)})$ until $\sum_{j=1}^{i} \mathbf{1}_{B_\varepsilon(z^{(i)})}(y) = r$. Let $N' = i$.
2. Sample L uniformly from the set $\{j \in \{1, \ldots, N'-1\} : \mathbf{1}_{B_\varepsilon(z^{(j)})}(y) = 1\}$.
3. For $i = 1, 2, \ldots$ sample $\theta^{(i)} \sim g(\cdot|\theta'^{(L)})$ and $x^{(i)} \sim f(\cdot|\theta^{(i)})$ until $\sum_{j=1}^{i} \mathbf{1}_{B_\varepsilon(x^{(i)})}(y) = r-1$. Let $N = i$.
4. With probability
$$\min\left\{1, \frac{p(\theta'^{(L)})g(\theta|\theta'^{(L)})}{p(\theta)g(\theta'^{(L)}|\theta)} \times \frac{N}{N'-1}\right\},$$
output $(\theta'^{(L)}, z^{(L)})$. Otherwise output (θ, x).

respectively, with success parameters $\mathrm{Vol}(\varepsilon)\int g(\bar{\theta}|\theta')f^\varepsilon(y|\bar{\theta})d\bar{\theta}$ and $\mathrm{Vol}(\varepsilon)\int g(\bar{\theta}|\theta)f^\varepsilon(y|\bar{\theta})d\bar{\theta}$. One can verify that detailed balance holds, i.e.

$$\pi^\varepsilon(\theta, x)\frac{g(\theta'|\theta)p^\varepsilon(z|y,\theta')}{\int g(\bar{\theta}|\theta)f^\varepsilon(y|\bar{\theta})d\bar{\theta}}\mathbb{E}\left[\min\left\{1, \frac{p(\theta')g(\theta|\theta')}{p(\theta)g(\theta'|\theta)} \times \frac{M_{r-1,\theta'}}{M_{r,\theta}-1}\right\}\right]$$
$$= \pi^\varepsilon(\theta', z)\frac{g(\theta|\theta')p^\varepsilon(x|y,\theta)}{\int g(\bar{\theta}|\theta')f^\varepsilon(y|\bar{\theta})d\bar{\theta}}\mathbb{E}\left[\min\left\{1, \frac{p(\theta)g(\theta'|\theta)}{p(\theta')g(\theta|\theta')} \times \frac{M_{r-1,\theta}}{M_{r,\theta'}-1}\right\}\right].$$

□

2.3 Computational Complexity

The robust, or adaptive, kernels proposed are different to the simple kernel in Algorithm 1 in that the number of simulations required depends on the values of θ and the proposed value θ'. As a result, an SMC sampler utilizing such kernels is similar to the PMC approach in that the cost of obtaining each step can be determined largely by the difficulty one has in hitting the data for various values of θ. In the examples that follow, we restrict our attention to the 1-hit kernel and the r-hit kernel with multiple θ proposals for $r = 2$ and compare their use within an SMC sampler with the use of the simple kernel and a PMC approach.

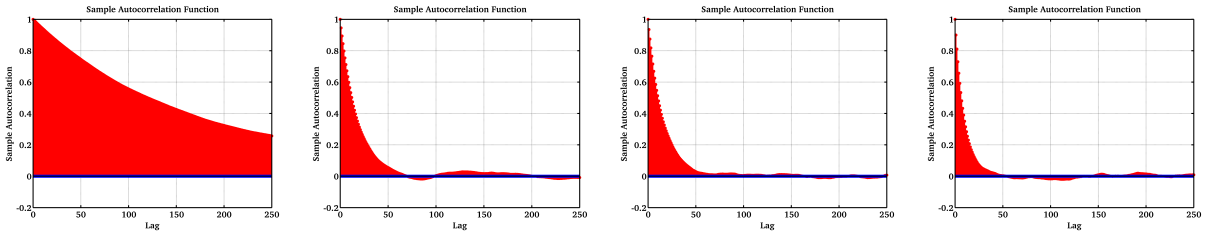
3 EXAMPLES

3.1 Univariate Normal Distribution

We consider a simple but relevant example in which we can compute $f(y|\theta)$ to show the differences between the kernels. This example may seem simplistic, but in fact the distribution of many summary statistics will be approximately normal, e.g. when dealing with a large number m of i.i.d. data and the summary statistic is the average of a summary statistic computed on each datum.

We look at the very simple case where the variance of the likelihood is known but the mean is not. In this case, θ is the mean and so $f(y|\theta) = \mathcal{N}(y; \theta, \sigma^2)$. The ABC likelihood is $f^\varepsilon(y|\theta) = \Phi(\frac{y+\varepsilon-\theta}{\sigma}) - \Phi(\frac{y-\varepsilon-\theta}{\sigma})$ for $\varepsilon > 0$. For our simulations, we let $y = 3$, $\sigma^2 = 1$ and use the prior $p(\theta) = \mathcal{N}(\theta; 0, 5)$, so that the exact posterior is $\mathcal{N}(\theta; \frac{5}{2}, \frac{5}{6})$. Figure 1 shows autocorrelation plots associated with each MCMC kernel, which all use a Gaussian random walk proposal with variance 0.25, for $\varepsilon = 0.1$. The exact kernel evaluates the likelihood f (i.e. $\varepsilon = 0$) explicitly for comparison. As one would expect, the simple kernel gives samples with a much larger autocorrelation than the adaptive kernels.

We ran an SMC sampler using the simple Metropolis-Hastings kernel, the 1-hit kernel and the 2-hit kernel with multiple θ proposals, as well as Algorithm 3 using adaptive proposal settings suggested in Beaumont, Cornuet, Marin, and Robert (2009). On this simple example, all the methods are able to recover the posterior reasonably accurately using $n = 500$ particles and $T = 100$ with $\varepsilon_t = 3 \times 0.97^t$. Figure 2



(a) Simple kernel (b) 1-hit kernel (c) 2-hit, multiple θ kernel (d) Exact kernel

Figure 1: Autocorrelation plots for the MCMC kernels.

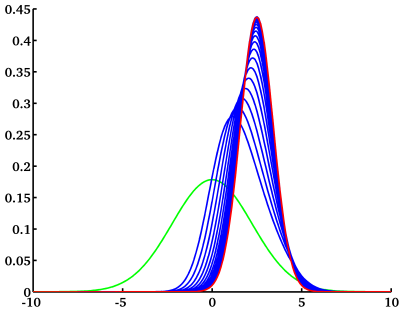


Figure 2: Sequence of ABC posterior densities (left to right) as ε decreases from 3 to 0.1 (blue). The prior ($\varepsilon = \infty$) and the exact posterior ($\varepsilon = 0$) are shown in green and red respectively.

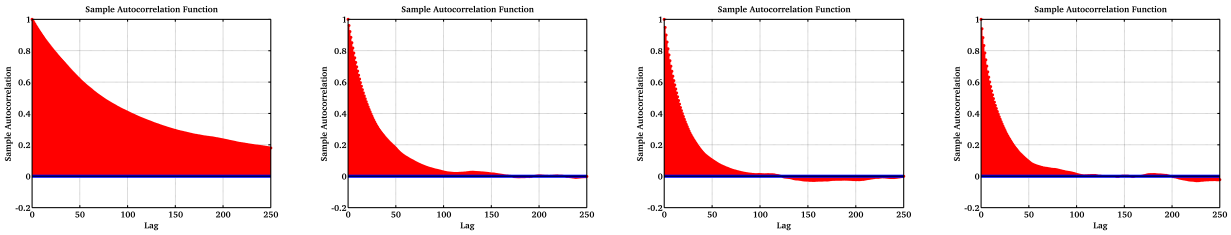
shows densities corresponding to ABC posteriors for different values of ε in addition to the prior and exact posterior. In addition, the posterior for θ is nearly indistinguishable from the exact posterior. The corresponding mean squared error of the mean, obtained over 100 runs of each filter was 0.0345, 0.0049, and 0.0048 for the simple, 1-hit and 2-hit kernels, and 0.0062 for Algorithm 3. In all cases the estimated bias was negligible except for the SMC sampler using the simple kernel.

3.2 Multivariate Normal Distribution

We now consider a more difficult example where the data consists of $m = 100$ i.i.d. 2-dimensional observations $\tilde{y}_{1:m}$ and where the summary statistics $y = S(\tilde{y}_{1:m})$ chosen to represent the data are the sample mean of each component and Pearson's sample correlation coefficient. The model for the data is a multivariate normal, for which the likelihood is analytically available, but we proceed to infer unknown parameters using ABC as a test of the methodology. In particular, the unknown parameters $\theta = (\mu_{1:2}, \sigma_{12})$ are the mean μ and $\sigma_{12} = \sigma_{21}$ where the covariance is $\Sigma = \{\sigma_{ij}\}$ and the values $\sigma_{11} = \sigma_{22} = 1$ are fixed. The prior for σ_{12} is uniform on $[-1,1]$ and the prior for each mean is a standard normal.

With simulated data obtained with parameters $(\mu_1, \mu_2, \sigma_{12}) = (-0.308, 2.26, 0.5)$, Figure 3 shows autocorrelation plots associated with each MCMC kernel for $\varepsilon = 0.1$. The kernels all use Gaussian random walk proposals with variance 0.01 for each mean component and 0.0625 for σ_{12}. The kernel actually used is a cycle of a mixture of kernels where only one component is updated at a time. As before, the autocorrelation of samples from the simple kernel is much higher than that of the adaptive kernels.

Figure 4 shows kernel density estimates obtained for one run of an SMC algorithm with 500 and 5000 particles using the kernels as well as Algorithm 3 (PMC). All of the samplers give reasonable estimates with respect to the ABC posterior using 5000 particles, but it is worth noting that the use of $\varepsilon_T = 0.1$ means that the exact posterior has slightly more mass concentrated near the true values, particularly of



(a) Simple kernel (b) 1-hit kernel (c) 2-hit, multiple θ kernel (d) Exact kernel

Figure 3: Autocorrelation plots for the MCMC kernels.

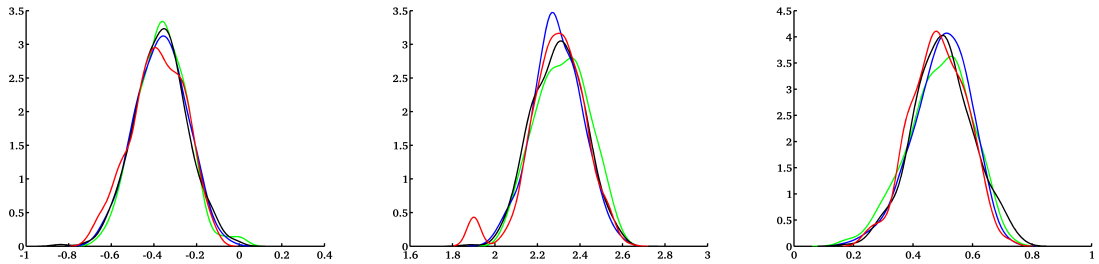
σ_{12}, than the ABC posterior. Of some interest here is the effective number of samples provided by the sampler, which is defined for weighted samples $(\theta_T^{(1:n)}, w_T^{(1:n)})$ as $\text{ESS} = \left(\sum_{j=1}^n w_T^{(j)}\right)^2 / \left(\sum_{j=1}^n \left(w_T^{(j)}\right)^2\right)$. For the output of the PMC sampler this can be calculated from the final time weights alone, whilst for the SMC samplers with MCMC kernels one can first group the particles with identical θ values to give a more accurate measure of particle diversity. For 500 particles the simple, 1-hit, 2-hit with multiple θ and PMC samplers provide effective sample sizes of 306, 409, 428 and 282 respectively whilst for 5000 particles they provide 3078, 4283, 4382 and 3698 respectively. While the results may not be indicative of general behavior the lower effective sample size values here for PMC are due to the variability of the weights from the difference between the proposal distribution at time T and the prior. In addition, the cost of the weighting step became more significant with 5000 particles for PMC, although we avoid discussing timing comparisons of the approaches here due to the reliance of run times on implementation, computer architecture and, most importantly, model-specific considerations.

4 DISCUSSION

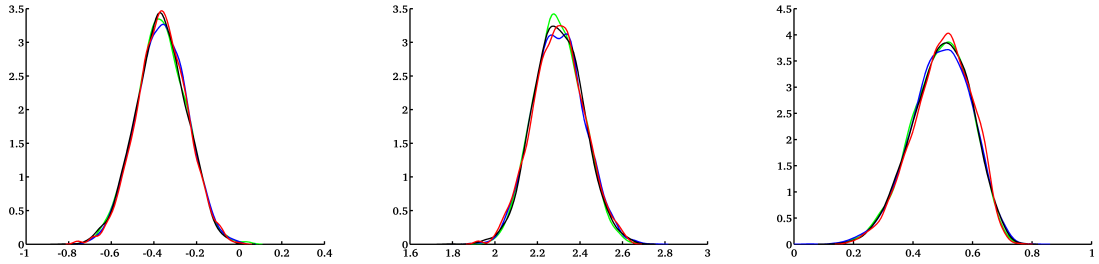
The adaptive kernels provide reasonable performance within an SMC sampler for the examples considered here. Naturally, it is difficult to ascertain how indicative these will be of their performance in general, and we do not attempt to interpret them as such. Nevertheless, the results suggest that future work could attempt to identify which SMC algorithms should be selected in particular situations. In practice, it is not expected that any one method will dominate any other in all situations. For a particular model, it would also be interesting to analyze how the methods scale with the dimension of θ.

The posterior for θ in the examples are not multi-modal, for which SMC approaches are known to be effective in comparison to other approaches. One might also expect the SMC sampler with the simple MCMC kernel to be less competitive when the desired ε of interest is smaller. Indeed, in the examples considered it is not clear that the additional computational effort required by the adaptive kernels is justified by the modest gains in accuracy. A possible criticism of the adaptive kernel is that one could instead specify deterministically to run longer cycles of standard kernels as ε decreases. However, the adaptive kernels both sidestep the need to tune this parameter, and also spend more computational effort only for pairs (θ, θ') that are in "difficult" regions of the space rather than for every possible pair equally. Another interesting question is if and how one can overcome the variation of the weights due to the discrepancy between the proposal and the prior in the PMC approach, which is responsible for lower effective sample sizes.

Direct comparisons of computation time have been avoided here, although for the algorithms implemented the run times of almost all the methods were comparable. The only exception was the SMC sampler using the simple MCMC kernel, which was faster. The reason for avoiding such comparisons is that different models will have different computational complexities associated with the simulation of data and so it is difficult to quantify the impact of the $O(n^2)$ cost of the weighting step in the PMC approach, which could be negligible for many reasonable values of n in practice. The adaptive kernels differ from the simple kernel



(a) μ_1 (b) μ_2 (c) σ_{12}



(d) μ_1 (e) μ_2 (f) σ_{12}

Figure 4: Kernel density estimates from the samplers using (a)-(c) 500 particles and (d)-(f) 5000 particles. The overlaid plots correspond to SMC samplers with simple MCMC kernels (green), 1-hit kernels (blue), 2-hit multiple θ kernels (black) and the PMC algorithm (red).

in that the computational complexity of the SMC sampler at each time is variable, and further work could seek to determine how PMC and SMC samplers compare in terms of computation time in this respect on a variety of realistic models. Another computational consideration is the potential ability to perform many simulations in parallel and whether or not this is affected by the use of a common value of θ, as might be the case when executing algorithms on a single instruction, multiple data architecture like a graphics processing unit (see, e.g., Lee et al. (2010)). Such considerations could lead one to prefer, e.g., the 1-hit kernel over the 2-hit kernel with multiple θ proposals or PMC. Given the large number of simulations required by ABC methods in general, any opportunity to parallelize computation is likely to be beneficial.

While not pursued here in order to simplify the presentation, the SMC samplers approach does not preclude the use of additional adaptation, such as adaptive schedule selection and the use of adaptive proposals within the MCMC kernels, as in Del Moral, Doucet, and Jasra (2012). One could also use the same adaptation in both the PMC and SMC samplers approaches. Combined with compelling recent advances also in the automatic selection of the summary statistics and the adjustment of the data to be consistent with the data-generating process (Fearnhead and Prangle 2012), such adaptive Monte Carlo methods could be used to provide nearly fully automated procedures for inference in a variety of ABC problems.

References

Beaumont, M., J. Cornuet, J. Marin, and C. Robert. 2009. "Adaptive approximate Bayesian computation". *Biometrika* 96 (4): 983–990.

Cappé, O., A. Guillin, J. Marin, and C. Robert. 2004. "Population monte carlo". *Journal of Computational and Graphical Statistics* 13 (4): 907–929.

Del Moral, P., A. Doucet, and A. Jasra. 2006. "Sequential Monte Carlo Samplers". *Journal of the Royal Statistical Society B* 68 (3): 411–436.

Del Moral, P., A. Doucet, and A. Jasra. 2012. "An adaptive sequential Monte Carlo method for approximate Bayesian computation". *Statistics and Computing* 22 (5): 1009–1020.

Douc, R., O. Cappé, and E. Moulines. 2005. "Comparison of resampling schemes for particle filtering". In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, 64–69. IEEE.

Fearnhead, P., and D. Prangle. 2012. "Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation". *Journal of the Royal Statistical Society B* 74 (3): 419–474.

Hastings, W. 1970. "Monte Carlo sampling methods using Markov chains and their applications". *Biometrika* 57 (1): 97–109.

Jasra, A., and A. Doucet. 2008. "Stability of sequential Monte Carlo samplers via the Foster–Lyapunov condition". *Statistics & Probability Letters* 78 (17): 3062–3069.

Lee, A., C. Andrieu, and A. Doucet. 2012a. "Active particles and locally adaptive Markov chain Monte Carlo". In preparation.

Lee, A., C. Andrieu, and A. Doucet. 2012b. "Discussion of paper by P. Fearnhead and D. Prangle". *Journal of the Royal Statistical Society B* 74 (3): 419–474.

Lee, A., C. Yau, M. Giles, A. Doucet, and C. Holmes. 2010. "On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods". *Journal of Computational and Graphical Statistics* 19 (4): 769–789.

Marin, J., P. Pudlo, C. Robert, and R. Ryder. 2012. "Approximate Bayesian computational methods". *Statistics and Computing*. To appear.

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

Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. 1953. "Equation of state calculations by fast computing machines". *Journal of Chemical Physics* 21 (6): 1087–1092.

Pritchard, J., M. Seielstad, A. Perez-Lezaun, and M. Feldman. 1999. "Population growth of human Y chromosomes: a study of Y chromosome microsatellites.". *Molecular Biology and Evolution* 16 (12): 1791–1798.

Schweizer, N. 2012. "Non-asymptotic error bounds for sequential MCMC methods in multimodal settings". ArXiv:1205.6733.

Sisson, S., Y. Fan, and M. Tanaka. 2007. "Sequential Monte Carlo without likelihoods". *Proceedings of the National Academy of Sciences* 104 (6): 1760–1765.

Tavaré, S., D. J. Balding, R. C. Griffiths, and P. Donnelly. 1997. "Inferring coalescence times from DNA sequence data". *Genetics* 145 (2): 505–518.

Toni, T., D. Welch, N. Strelkowa, A. Ipsen, and M. Stumpf. 2009. "Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems". *Journal of the Royal Society Interface* 6 (31): 187–202.

Whiteley, N. 2012. "Sequential Monte Carlo samplers: error bounds and insensitivity to initial conditions". *Stochastic Analysis and Applications* 30 (5): 774–798.

AUTHOR BIOGRAPHIES

ANTHONY LEE is a Research Fellow in the Centre for Research in Statistical Methodology in the Department of Statistics at the University of Warwick. He received a D. Phil. in Statistics from the University of Oxford in 2011. His research interests are in computational statistics and Bayesian inference, with an emphasis on Monte Carlo methodology. His email address is anthony.lee@warwick.ac.uk and his web page is http://www.warwick.ac.uk/go/alee.