Bayesian Analysis (2013) **8**, Number 2, pp. 411–438

An Adaptive Sequential Monte Carlo Sampler

Paul Fearnhead * and Benjamin M. Taylor †

Abstract. Sequential Monte Carlo (SMC) methods are not only a popular tool in the analysis of state–space models, but offer an alternative to Markov chain Monte Carlo (MCMC) in situations where Bayesian inference must proceed via simulation. This paper introduces a new SMC method that uses adaptive MCMC kernels for particle dynamics. The proposed algorithm features an online stochastic optimization procedure to select the best MCMC kernel and simultaneously learn optimal tuning parameters. Theoretical results are presented that justify the approach and give guidance on how it should be implemented. Empirical results, based on analysing data from mixture models, show that the new adaptive SMC algorithm (ASMC) can both choose the best MCMC kernel, and learn an appropriate scaling for it. ASMC with a choice between kernels outperformed the adaptive MCMC algorithm of Haario et al. (1998) in 5 out of the 6 cases considered.

Keywords: Adaptive MCMC, Adaptive Sequential Monte Carlo, Bayesian Mixture Analysis, Optimal Scaling, Stochastic Optimization

1 Introduction

Sequential Monte Carlo (SMC) is a class of algorithms that enable simulation from a target distribution of interest. These algorithms are based on defining a series of distributions, and generating samples from each distribution in turn. SMC was initially used in the analysis of state-space models. In this setting there is a time–evolving hidden state of interest, inference about which is based on a set of noisy observations (Gordon et al. 1993; Liu and Chen 1998; Doucet et al. 2001; Fearnhead 2002). The sequence of distributions is defined as the set of posterior distributions of the state at consecutive time-points given the observations up to those time points. More recent work has looked at developing SMC methods that can analyse state-space models which have unknown fixed parameters. Such methods introduce steps into the algorithm to allow the support of the sample of parameter values to change over time, for example by using ideas from kernel density estimation (Liu and West 2001), or Markov chain Monte Carlo (MCMC) moves (Gilks and Berzuini 1999; Storvik 2002; Fearnhead 2002).

Most recently, SMC methods have been applied as an alternative to MCMC for standard Bayesian inference problems (Neal 2001; Chopin 2002; Del Moral et al. 2006; Fearnhead 2008). In this paper the focus will be on methods for sampling from the posterior distribution of a set of parameters of interest. SMC methods for this class of targets introduce an artificial sequence of distributions running from the prior to the

*Department of Mathematics and Statistics, Lancaster University, UK. p.fearnhead@lancaster.ac.uk
†(Corresponding Author) Faculty of Health and Medicine, Lancaster University, UK. b.taylor1@lancaster.ac.uk

 DOI:10.1214/13-BA814

posterior, and sample recursively from these using a combination of Importance Sampling and MCMC moves. This approach to sampling has been demonstrated empirically to often be more effective than using a single MCMC chain (Jasra et al. 2007, 2008a). There are heuristic reasons for why this may true in general: the annealing of the target and spread of samples over the support means that SMC is less likely to be become trapped in posterior modes.

Simply invoking an untuned MCMC move within an SMC algorithm would likely lead to poor results because the move step would not be effective in combating sample depletion. The structure of SMC means that at the time of a move there is a sample from the target readily available, this can be used to compute posterior moments and inform the shape of the proposal kernel as in Jasra et al. (2008b); however, further refinements can lead to even better performance. Such refinements include the scaling of estimated target moments by an optimal factor, see Roberts and Rosenthal (2001) for example. For general targets and proposals no theoretical results for the choice of scaling exist, and this has led to the recent popularity of adaptive MCMC (Haario et al. 1998; Andrieu and Robert 2001; Roberts and Rosenthal 2009; Craiu et al. 2009; Andrieu and Thoms 2008). In this paper the idea of adapting the MCMC kernel within an SMC algorithm will be explored.

To date there has been little work at adapting SMC methods. Exceptions include the method of Jasra et al. (2008b), whose method assumes a likelihood tempered sequence of target densities (see Neal (2001)) and the adaptation procedure both chooses this sequence online and computes the variance of a random walk proposal kernel used for particle dynamics. Cornebise et al. (2008) also considers adapting the proposal distribution within SMC for state-space models. Assuming that the proposal density belongs to a parametric family with parameter θ, their method proceeds by simulating a number of realisations for each of a range of values of θ and selecting the value that minimises the empirical Shannon entropy of the importance weights; new samples are then re–proposed using this approximately optimal value. Further related work includes that of Douc et al. (2007) and Cappé et al. (2008) on respectively population Monte Carlo and adaptive importance sampling and also Schäfer and Chopin (2013).

The aims of this paper are to introduce a new adaptive SMC algorithm (ASMC) that automatically tunes MCMC move kernels and chooses between different proposal densities and to provide theoretical justification of the method. The algorithm is based on having a distribution of kernels and their tuning parameters at each iteration. Each current sample value, called a particle, is moved using an MCMC kernel drawn from this distribution. By observing the expected square jumping distance (Craiu et al. 2009; Sherlock and Roberts 2009) for each particle it is possible to learn in some sense which MCMC kernels are mixing better. The information thus obtained can then be used to update the distribution of kernels. The key assumption of the new approach is that the optimal MCMC kernel for moving particles does not change much over the iterations of the SMC algorithm; we note that this assumption is more intrinsic to SMC methods and not confined to our proposed algorithm. As will be discussed and shown empirically in Section 5, this can often be achieved by appropriate parameterisation of a family of MCMC kernels.

The structure of the paper is as follows. In the next section, the model of interest will be introduced and followed by a review of MCMC and SMC approaches. Then in Section 3, the new adaptive SMC will be presented. Guidelines on implementing the algorithm as well as some theory on the convergence will be presented in Section 4. In Section 5 the method will be evaluated using simulated data. The results show that the proposed method can successfully choose both an appropriate MCMC kernel and an appropriate scaling for the kernel. The paper ends with a discussion.

2 Model

The focus of this article will be on Bayesian inference for parameters, θ, from a model where independent identically distributed data is available. Note that the ideas behind the proposed adaptive SMC algorithm can also be applied to the non i.i.d. case, see Section 6. Let $\pi(\theta)$ denote the prior for θ and $\pi(y|\theta)$ the probability density for the observations. The aim will be to calculate the posterior density,

$$\pi(\theta|y_{1:n}) \propto \pi(\theta)\prod_{i=1}^{n}\pi(y_i|\theta), \tag{1}$$

where, here and throughout, π will be used to denote a probability density, and $y_{1:n}$ means $y_1, \ldots, y_n$.

In general, $\pi(\theta|y_{1:n})$ is analytically intractable and so to compute posterior functionals of interest, for example expectations, Monte Carlo simulation methods are often employed. Sections 2.1 and 2.2 provide a brief description of two such Monte Carlo approaches.

2.1 MCMC

An MCMC transition kernel, K_h, is a probability law governing the transition between states of a discrete Markov chain with some stationary distribution of interest, for example a posterior. K_h comprises a proposal kernel, here and throughout denoted q_h (the subscript h indicates dependence on a tuning parameter) and an acceptance ratio that depends on the target and, in general, the proposal densities (see Gilks et al. (1995); Gamerman and Lopes (2006) for reviews of MCMC methodology). The most generally applicable MCMC method is Metropolis–Hastings, described in Algorithm 1 (Metropolis et al. 1953; Hastings 1970).

Probably the simplest MH algorithm is random walk Metropolis (RWM). The proposal kernel for RWM is a symmetric density centred on the current state, the most common example being a multivariate normal, $q_h(\theta^{(i-1)}, \tilde{\theta}) = \mathcal{N}(\tilde{\theta}; \theta^{(i-1)}, h^2\hat{\Sigma}_\pi)$, where $\hat{\Sigma}_\pi$ is an estimate of the target covariance. Both the values of $\hat{\Sigma}_\pi$ and h are critical to the performance of the algorithm. If $\hat{\Sigma}_\pi$ does not accurately estimate the posterior covariance matrix, then the likely directions of the random walk moves will probably be inappropriate. On the other hand, a value of h that is too small will lead to high acceptance rates, but the samples will be highly correlated. If h is too large then the

Algorithm 1 Metropolis–Hastings Algorithm (Metropolis et al. 1953; Hastings 1970)

1: Start with an initial sample, $\theta^{(0)}$, drawn from any density, π_0.
2: **for** $j = 1, 2, \ldots$ **do**
3: Propose a move to a new location, $\tilde{\theta}$, by drawing a sample from $q_h(\theta^{(i-1)}, \tilde{\theta})$.
4: Accept the move (i.e., set $\theta^{(i)} = \tilde{\theta}$) with probability,

$$\min\left\{1, \frac{\pi(\tilde{\theta}|y_{1:n})}{\pi(\theta^{(i-1)}|y_{1:n})} \frac{q_h(\tilde{\theta}, \theta^{(i-1)})}{q_h(\theta^{(i-1)}, \tilde{\theta})}\right\}, \tag{2}$$

else set $\theta^{(i)} = \theta^{(i-1)}$.
5: **end for**

algorithm will rarely move, which in the worst case scenario could lead to a degenerate sample.

These observations on the rôle of h point to the idea of an *optimal scaling*, a h somewhere between the extremes that promotes the best mixing of the algorithm. In the case of elliptically symmetric unimodal targets, an optimal random walk scaling can sometimes be computed numerically; this class of targets includes the Multivariate Gaussian (Sherlock and Roberts 2009). Other theoretical results include optimal acceptance rates which are derived in the limit as the dimension of θ, $d \to \infty$ (see Roberts and Rosenthal (2001) for examples of targets and proposals). In general however, there are no such theoretical results.

One way of circumventing the need for analytical optimal scalings is to try to learn them online (Andrieu and Robert 2001; Atchadé and Rosenthal 2005), this can include both learning a good scaling, h, and estimating the target covariance, $\hat{\Sigma}_\pi$ (Haario et al. 1998). Recent research in adaptive MCMC has generated a number of new algorithms (see for example Andrieu and Thoms (2008); Roberts and Rosenthal (2009); Craiu et al. (2009)), though some care must be taken to ensure that the resulting chain has the correct ergodic distribution.

2.2 Sequential Monte Carlo

Sequential Monte Carlo (SMC) is a class of simulation–based methods for sampling from a target density of interest (see Doucet et al. (2001); Del Moral et al. (2006) for a review). The main idea behind SMC is to introduce a sequence of densities leading from the prior to the target density of interest and to iteratively update an approximation to these densities. For the application considered here, it is natural to define these densities as $\pi_t(\theta) = \pi(\theta|y_{1:t})$ for $t = 1, \ldots, n$; this 'data tempered' schedule will be used in the sequel. The approximations to each density are defined in terms of a collection of particles, $\theta_t^{(j)}$, together with their respective weights, $w_t^{(j)}$, for $j = 1, \ldots, M$, produced so that as $M \to \infty$, Monte Carlo sums converge almost surely to their 'correct'

expectations:

$$\frac{\sum_{j=1}^{M} w_t^{(j)} \zeta(\theta_t^{(j)})}{\sum_{i=1}^{M} w_t^{(i)}} \xrightarrow{\text{a.s.}} \mathbb{E}_{\pi_t(\theta_t)}[\zeta(\theta_t)],$$

for all π_t–integrable functions, ζ. Such a collection of particles and their respective weights will be denoted by $\{\theta_t^{(j)}, w_t^{(j)}\}_{j=1}^M$; notice that we do not assume the sum of the weights is equal to one. Furthermore, we will often write $\{\theta_t^{(j)}, w_t^{(j)}\}_{j=1}^M \sim \pi_t(\theta_t)$ or $\{\theta_t^{(j)}, w_t^{(j)}\}_{j=1}^M \sim \pi_t$ to make this explicit. When each particle, or sample, has weight $1/M$, we will sometimes write $\{\theta_t^{(j)}, 1/M\}_{j=1}^M$ and other times $\{\theta_t^{(j)}\}_{j=1}^M$.

One step of an SMC algorithm can involve importance reweighting, resampling and moving the particles via an MCMC kernel (Gilks and Berzuini 1999; Chopin 2002). For concreteness, this paper will focus on the iterated batch importance sampling (IBIS) algorithm of Chopin (2002).

The simplest way to update the particle approximation in model (1) is to let $\theta_t^{(j)} = \theta_{t-1}^{(j)}$ and $w_t^{(j)} = w_{t-1}^{(j)} \pi(y_t|\theta_t^{(j)})$. However such an algorithm will degenerate for large t, as eventually only one particle will have non-negligible weight. With IBIS, resample–move steps (sometimes referred to here as simply 'move steps') are introduced to alleviate this. In a move step, the particles are first resampled so that the expected number of copies of particle $\theta_t^{(j)}$ is proportional to $w_t^{(j)}$. This process produces multiple copies of some particles. In order to create particle diversity, each resampled particle is moved by an MCMC kernel. The MCMC kernel is chosen to have stationary distribution π_t. The resulting particles are then assigned a weight of $1/M$.

The decision of whether to apply a resample-move step within IBIS is based on the effective sample size (ESS, see Kong et al. (1994); Liu and Chen (1998)). The ESS is a measure of variability of the particle weights; using this to decide whether to resample is justified by arguments within Liu and Chen (1995) and Liu et al. (1998). Full details of IBIS are given in Algorithm 2.

Whilst the focus of this article is on the IBIS algorithm, the ideas presented here can be applied to more general SMC algorithms (Del Moral et al. 2006). The use of an MCMC-Kernel move within IBIS is itself probably the most common approach for implementing SMC in practice (and corresponds to the implementation described in Section 3.3.2.3 of Del Moral et al. (2006)). Furthermore, the adaptive method developed in Section 3 immediately applies to algorithms that use a different sequence of target distributions, for example likelihood tempering (Neal 2001). The likelihood tempered target sequence is $\pi_t(\theta) = \pi(\theta)\pi(y_{1:n}|\theta)^{\xi_t}$, where $\{\xi_t\}$ is a sequence of real numbers starting at 0 (the prior) and ending on 1 (the posterior). The focus here is on data tempering, $\pi_t(\theta) = \pi(\theta|y_{1:t})$, because for the application considered in Section 5.2, calculating the likelihood has a cost which increases linearly with the number of observations.

The SMC algorithm of Del Moral et al. (2006) also implements moves at each iteration of the algorithm. However the approach in this article is closely related, and is

Algorithm 2 Chopin's IBIS algorithm

Initialise from the prior $\{\theta_0^{(j)}, w_0^{(j)}\}_{j=1}^M \sim \pi_0$.
for $t = 1, \ldots, n$ **do**
  Reweight $w_t^{(j)} = w_{t-1}^{(j)} \pi_t(\theta_{t-1}^{(j)}) / \pi_{t-1}(\theta_{t-1}^{(j)})$. Result: $\{\theta_{t-1}^{(j)}, w_t^{(j)}\}_{j=1}^M \sim \pi_t$.
  if particle weights not degenerate (see text) **then**
    $\{\theta_t^{(j)}, w_t^{(j)}\}_{j=1}^M \leftarrow \{\theta_{t-1}^{(j)}, w_{t-1}^{(j)}\}_{j=1}^M$
    $t \rightarrow t+1$.
  else
    Resample: let $\mathcal{K} = \{k_1, \ldots, k_M\} \subseteq \{1, \ldots, M\}$ be the resampling indices, then $\{\theta_{t-1}^{(k)}, 1/M\}_{k \in \mathcal{K}} \sim \pi_t$. Relabel: $k_j \leftarrow j$, the jth resampling index so that $\{\theta_{t-1}^{(j)}, 1/M\}_{j=1}^M \sim \pi_t$.
    Move via π_t–invariant MCMC kernel. Result: $\{\theta_t^{(j)}, 1/M\}_{j=1}^M \sim \pi_t$.
  end if
end for

equivalent to using a sequence of targets, $\tilde{\pi}_j(\theta) = \pi(\theta | y_{1:t_j})$, where t_j is the iteration where IBIS resamples for the jth time. IBIS is thus similar to SMC, with $\tilde{\pi}_j(\theta)$ as the sequence of targets. As such, IBIS itself can be viewed as choosing the sequence of targets in an adaptive way based on the closeness of successive targets, $\tilde{\pi}_{j-1}(\theta)$ and $\tilde{\pi}_j(\theta)$, as measured by the variability of the importance weights (Del Moral et al. 2010).

The efficiency of an algorithm such as IBIS depends on the mixing properties of the associated MCMC kernel. Within SMC there is the advantage of being able to use the current set of particles to help tune an MCMC kernel. For example, the weighted particles can give an estimate of the posterior covariance matrix, which can be used within a random walk proposal. However even in this case, the proposal variance still needs to be appropriately scaled (Roberts and Rosenthal 2001; Sherlock and Roberts 2009). In the next section the new adaptive SMC procedure will be introduced. The new algorithm can learn an appropriate tuning for the MCMC kernel, and can also be used to choose between a set of possible kernels.

3 The Adaptive SMC Sampler

First consider the case where the move step in the IBIS algorithm involves one type of MCMC kernel. Let π_t be an *arbitrary* continuous probability density (the target) and $K_{h,t}$ a π_t–invariant MCMC kernel with tuning parameter, h. The parameter h is to be chosen to maximise the following utility function,

$$\begin{aligned} g^{(t)}(h) &= \int \pi_t(\theta_{t-1}) K_{h,t}(\theta_{t-1}, \theta_t) \Lambda(\theta_{t-1}, \theta_t) \mathrm{d}\theta_{t-1} \mathrm{d}\theta_t, \\ &= \mathbb{E}\left[\Lambda(\theta_{t-1}, \theta_t)\right], \end{aligned} \tag{3}$$

where $\Lambda(\theta_{t-1}, \theta_t) > 0$ is a measure of mixing of the chain. Most MCMC adaptation criteria can be viewed in this way (Andrieu and Thoms 2008). Note that for simplicity

of presentation, Λ only depends on the current and subsequent state, though the idea readily extends to more complex cost functionals, for example involving multiple transitions of the MCMC chain. The function $g^{(t)}$ is the average performance of the chain with respect to Λ, which would normally be some measure of mixing. A computationally simple measure of mixing is the expected square jumping distance (ESJD). Maximising the ESJD is equivalent to minimising the lag-1 autocorrelation; this measure is often used within adaptive MCMC, see for example Sherlock and Roberts (2009) and Pasarica and Gelman (2010).

In the following it will be assumed that the proposal distribution can depend on quantities calculated from the current set of particles (for example estimates of the posterior variance), but this will be suppressed in the notation. The main idea of ASMC is to use the observed instances of $\Lambda(\theta_{t-1}, \theta_t)$ to help choose the best h. The tuning parameter will be treated as an auxiliary random variable. At time-step t the aim is to derive a density for the tunings, $\pi^{(t)}(h)$; note this should not be confused with the target densities on the parameters, $\pi_t(\theta)$. If a move step is invoked at this time, a sample of M realisations from $\pi^{(t)}(h)$, denoted $\{h_t^{(j)}\}_{j=1}^M$, will be drawn and 'allocated' to particles at random.

When moving the jth resampled particle, the tuning parameter $h_t^{(j)}$ will be used within the proposal distribution. Let $\theta_{t-1}^{(j)}$ be the jth resampled particle (see step 8 of Algorithm 2). In moving this particle, $\tilde{\theta}_t^{(j)}$ is drawn from $q_{h_t^{(j)}}(\theta_{t-1}^{(j)}, \cdot)$, and accepted with probability $\alpha_{h_t^{(j)}}(\theta_{t-1}^{(j)}, \tilde{\theta}_t^{(j)})$, given by (2). If the proposed particle is accepted then $\theta_t^{(j)} = \tilde{\theta}_t^{(j)}$ otherwise $\theta_t^{(j)} = \theta_{t-1}^{(j)}$.

We recommend that low-variance resampling methods such as residual, or stratified sampling be used for resampling both the particles as well as the tunings (Whitley 1994; Kitagawa 1996; Liu et al. 1998; Carpenter et al. 1999).

In practice, we use a Rao-Blackwellised, unbiased estimate of the utility function, $g(h_t^{(j)})$,

$$\tilde{\Lambda}(\theta_{t-1}^{(j)}, \tilde{\theta}_t^{(j)}) = \alpha_{h_t^{(j)}}(\theta_{t-1}^{(j)}, \tilde{\theta}_t^{(j)})\Lambda(\theta_{t-1}^{(j)}, \tilde{\theta}_t^{(j)}). \tag{4}$$

The approach in this paper is to use the observed $\tilde{\Lambda}(\theta_{t-1}^{(j)}, \tilde{\theta}_t^{(j)})$ to update the distribution $\pi^{(t)}(h)$ to a new distribution $\pi^{(t+1)}(h)$ in a way that moves towards values of h with a higher ESJD. In particular each $h_t^{(j)}$ will be assigned a weight, $f(\tilde{\Lambda}(\theta_{t-1}^{(j)}, \tilde{\theta}_t^{(j)}))$, for some function $f : \mathbb{R}^+ \to \mathbb{R}^+$. The new density of scalings will be defined,

$$\pi^{(t+1)}(h) \propto \sum_{j=1}^{M} f(\tilde{\Lambda}(\theta_{t-1}^{(j)}, \tilde{\theta}_t^{(j)}))R(h - h_t^{(j)}), \tag{5}$$

where $R(h - h_t^{(j)})$ is a density for h which is centred on $h_t^{(j)}$. Simulating from $\pi^{(t+1)}(h)$ is achieved by first resampling the $h_t^{(j)}$s with probabilities proportional to their weight and then adding noise to each resampled value; the distribution of this noise is given

by $R(\cdot)$. If there is no resampling at step t then $\pi^{(t+1)}(h) = \pi^{(t)}(h)$. In practice, the scheme can be initiated with an arbitrary distribution $\pi(h)$.

In computing a new density of the tunings, a function f will be used to weight the $h_t^{(j)}$s. The function f should be increasing with $\tilde{\Lambda}$, so that more weight is placed on tunings that produce bigger moves. The specific choice of f considered in this paper is a simple linear weighting scheme,

$$f(\tilde{\Lambda}) = a + \tilde{\Lambda}, \qquad a \geq 0. \tag{6}$$

Theoretical justification for this choice is given in the next section. This approach is similar in spirit to that of genetic algorithms (Jennison and Sheehan 1995).

The motivation for adding noise to the resampled h–values is to avoid the distributions $\pi^{(t)}(h)$ degenerating too quickly to a point-mass on a single value. It can be viewed as a form of kernel density estimation, and as such it is natural to allow the variance of the noise to depend on the variance of $\pi^{(t)}(h)$ and the number of particles, M. Asymptotic results for kernel density estimation suggest that this variance should decrease to 0 as M increases. Similar ideas are used in dynamic SMC methods for dealing with fixed parameters, for example West (1993); Liu and West (2001).

An initial collection of tuning parameters is drawn from a density, $\pi^{(0)}(h)$. In the examples considered here, this density was taken to be a uniform density on an appropriate support. For adaptive random walk kernels, an appropriate support may be constructed quickly by using ideas similar to those in Section 5.1 to compute $g(h)$ (defined in (3)) for a Gaussian of the appropriate dimension, and then choosing a wide interval around the mode.

One assumption of the proposed approach is that a good choice of h at one time-step will be a good choice at nearby time-steps. Note that this is based on an implicit assumption within SMC that successive targets are similar (see Chopin (2002); Del Moral et al. (2006) for example). Furthermore, using estimates of posterior variances within the proposal distribution can also help ensure that good values of h at one time-step will be a good choice at nearby time-steps. Some theoretical results concerning this matter will be presented in Section 4. We note that for improved performance, but at an additional computational cost, an MCMC move step could be applied at any iteration of the algorithm.

To choose between different types of MCMC kernel is now a relatively straightforward extension of the above. Assume there are n_K different MCMC kernels, each defined by a proposal distribution $q_{h,i}$, where $i \in \{1, \ldots, n_K\}$. Instead of just resampling the tuning parameters after the particle resampling step, now both the kernels and their associated parameters are resampled. The algorithm learns a set of distributions, $\pi^{(t)}(h, i)$, for the pair of kernel type and associated tuning parameter. Each particle is assigned a random kernel type and tuning drawn from this distribution, with the pair, $(h_{t-1}^{(j)}, i_{t-1}^{(j)})$, associated with $\theta_{t-1}^{(j)}$. The algorithm proceeds by weighting this pair based

on the observed $\tilde{\Lambda}(\theta_{t-1}^{(j)}, \tilde{\theta}_t^{(j)})$ values as before, and updating the distribution,

$$\pi^{(t)}(h,i) \propto \sum_{j=1}^{M} f(\tilde{\Lambda}(\theta_{t-1}^{(j)}, \tilde{\theta}_t^{(j)}))R(h - h_{t-1}^{(j)})\delta_{i_{t-1}^{(j)}}(i) \tag{7}$$

where $\delta_{i_{t-1}^{(j)}}(i)$ is a point mass on $i = i_{t-1}^{(j)}$.

The method is described in detail below, see Algorithm 3. Within the specific implementation described, the pairs, $\{(h_t^{(j)}, i_t^{(j)})\}_{j=1}^M$, sampled from $\pi^{(t)}(h,i)$ are allocated to particles randomly immediately after the resample–move step at iteration t. These pairs are then kept until the next time a resample–move step is called.

Algorithm 3 The Adaptive SMC algorithm. Here, $\pi_0(\cdot), \ldots, \pi_n(\cdot)$ are an arbitrary sequence of targets; an MCMC kernel is assumed for particle dynamics.

1: Initialise from the prior $\{\theta_0^{(j)}, w_0^{(j)}\}_{j=1}^M \sim \pi_0$.
2: Draw a selection of pairs of MCMC kernels with associated tuning parameters, $\{(h_0^{(j)}, K_{h,0}^{(j)})\}_{j=1}^M \equiv \{(h_0^{(j)}, i_0^{(j)})\}_{j=1}^M \sim \pi(h,i)$, and attach one to each particle arbitrarily.
3: **for** $t = 1, \ldots, n$ **do**
4: Reweight $w_t^{(j)} = w_{t-1}^{(j)} \pi_t(\theta_{t-1}^{(j)}) / \pi_{t-1}(\theta_{t-1}^{(j)})$. Result: $\{\theta_{t-1}^{(j)}, w_t^{(j)}\}_{j=1}^M \sim \pi_t$.
5: **if** particle weights not degenerate (see text) **then**
6: $\{\theta_t^{(j)}, w_t^{(j)}\}_{j=1}^M \leftarrow \{\theta_{t-1}^{(j)}, w_{t-1}^{(j)}\}_{j=1}^M$
7: $\{(h_t^{(j)}, K_{h,t}^{(j)})\}_{j=1}^M \leftarrow \{(h_{t-1}^{(j)}, K_{h,t-1}^{(j)})\}_{j=1}^M$
8: $t \to t+1$.
9: **else**
10: Resample: let $\mathcal{K} = \{k_1, \ldots, k_M\} \subseteq \{1, \ldots, M\}$ be the resampling indices, then $\{\theta_{t-1}^{(k)}, 1/M\}_{k \in \mathcal{K}} \sim \pi_t$. Relabel: $k_j \leftarrow j$, the jth resampling index so that $\{\theta_{t-1}^{(j)}, 1/M\}_{j=1}^M \sim \pi_t$. DO NOT resample kernels or tuning parameters at this stage.
11: Move $\theta_{t-1}^{(j)}$ via the π_t–invariant MCMC kernel, $K_{h,t}^{(j)}$, and tuning parameter $h_{t-1}^{(j)}$, denote the proposed new particle as $\tilde{\theta}_t^{(j)}$ and accepted/rejected particle as $\theta_t^{(j)}$. Result: $\{\theta_t^{(j)}, 1/M\}_{j=1}^M \sim \pi_t$.
12: To obtain $\{(h_t^{(j)}, K_{h,t}^{(j)})\}_{k=1}^M \equiv \{(h_t^{(j)}, i_t^{(j)})\}$, sample M times from (7). Allocate the new selection to particles at random.
13: **end if**
14: **end for**

The new method treats the tuning parameters as auxiliary random variables, but this is not the only way to choose good tuning parameters. One option would be to directly optimise $g^{(t)}(h)$ at each iteration, however this function is typically intractable (except for a small subset of targets and MCMC kernels). Another option would be to use stochastic optimisation at each move step in the style of Andrieu and Thoms (2008).

4 Theoretical Results

In this section the proposed algorithm will be justified by a series of theoretical results; guidance as to how it should best be implemented will also be given. The results presented here apply in the limit as the number of particles, $M \to \infty$.

We assume a multivariate tuning parameter, $\boldsymbol{h}$, and that there are finitely many types of MCMC kernel to choose from. In this section, we further assume that the variance of the kernel $R(\cdot)$ in (5) is 0. Lastly, we assume that the choice of MCMC kernels can be represented as a random variable I, taking values in a finite set, $i \in \mathfrak{I} := \{i_1, \ldots, i_{n_K}\}$. The extension to a countable choice of kernels is trivial, but irrelevant from a practical point of view. For notational simplicity we assume $\boldsymbol{h}$ can take values in the same set $\mathcal{H}$ regardless of the value i, though generalising this is trivial.

For a slight notational simplification, the criterion Λ will be used, rather than $\tilde{\Lambda}$ (as suggested in algorithm 3); this does not affect the validity of any of the arguments, which also hold for $\tilde{\Lambda}$. The section is split into two parts.

Firstly, in section 4.1, it is of interest to examine what happens to the distribution of $(\boldsymbol{h}, i)$s after one step of reweighting and resampling; this result will lead to a criterion for the choice of weight function that guarantees MCMC mixing improvement with respect to Λ. In section 4.2, the sequential improvement of $(\boldsymbol{h}, i)$s will be considered over many steps of the ASMC algorithm and with a changing target. General conditions for convergence of ASMC to the optimal kernel and tuning parameter will be provided.

4.1 One Step Improvement and Weighting Function

In this section and in the relevant proofs, it is appropriate to temporarily drop the t superscript, e.g., $g^{(t)} \equiv g$, $\boldsymbol{h}_t^{(j)} \equiv \boldsymbol{h}^{(j)}$, $\theta_{t-1} \equiv \theta$, $\theta_t \equiv \theta'$, and $\pi_{t-1}(\theta) \equiv \pi(\theta)$. For a given number of particles M we will have that the θ are drawn from a density $\pi^{[M]}(\theta)$ which is the SMC approximation to $\pi(\theta)$.

To study the effect of reweighting and resampling on the distribution of the $\boldsymbol{h}$s and is, suppose that currently $\{(\boldsymbol{h}^{(j)}, i^{(j)})\}_{j=1}^{M} \overset{\text{iid}}{\sim} \pi(\boldsymbol{H}|I)\pi(I)$, the joint pdf of a random variable, $(\boldsymbol{H}, I)$. The weight attached to any pair $(\boldsymbol{h}, i)$ is random. Therefore, since these pairs are assigned to particles independently of the value of the particle, the weight has mean,

$$w^{[M]}(\boldsymbol{h}, i) = \int \pi^{[M]}(\theta) K_{\boldsymbol{h}}^{(i)}(\theta, \theta') f(\Lambda(\theta, \theta')) \mathrm{d}\theta \mathrm{d}\theta'.$$

Now if we define a weight based on the conditional expection of $f(\Lambda)$ when $\theta \sim \pi(\theta)$:

$$w(\boldsymbol{h}, i) = \mathbb{E}_{\Theta, \Theta'|\boldsymbol{H}, I}[f(\Lambda)|\boldsymbol{H} = \boldsymbol{h}, I = i] = \int \pi(\theta) K_{\boldsymbol{h}}^{(i)}(\theta, \theta') f(\Lambda(\theta, \theta')) \mathrm{d}\theta \mathrm{d}\theta' \quad (8)$$

then standard SMC results give that under regularity conditions, we will have that as $M \to \infty$ that $w^{[M]}(\boldsymbol{h}, i) \to w(\boldsymbol{h}, i)$ in probability, see Crisan (2001) and Del Moral (2004).

The following proposition, which is used repeatedly in subsequent results, shows how reweighting and resampling affects $\pi(\boldsymbol{h}, i)$.

Proposition 1. *Suppose* $\theta \sim \pi^{[M]}(\theta)$ *and* $\{(\boldsymbol{h}^{(j)}, i^{(j)})\}_{j=1}^{M} \overset{iid}{\sim} \pi(\boldsymbol{H}|I)\pi(I)$*, the joint pdf of a random variable,* $(\boldsymbol{H}, I)$*, independent of* θ*. Let* $w(\boldsymbol{h}, i)$ *be the weighting function defined as in (8). Assume as* $M \to \infty$ *we have* $w^{[M]}(\boldsymbol{h}, i) \to w(\boldsymbol{h}, i)$ *in probability; also suppose that* $\sum_{i\in\mathfrak{I}} \int_{\mathcal{H}} w(\boldsymbol{h}, i)\pi(\boldsymbol{h}, i)\mathrm{d}\boldsymbol{h} > 0$ *and is finite.*

Then in the limit as $M \to \infty$*, the distribution of the reweighted and subsequently resampled* $(\boldsymbol{h}, i)$*s is,*

$$\pi^{\star}(\boldsymbol{h}, i) = \frac{w(\boldsymbol{h}, i)\pi(\boldsymbol{h}, i)}{\sum_{i\in\mathfrak{I}} \int_{\mathcal{H}} w(\boldsymbol{h}, i)\pi(\boldsymbol{h}, i)\mathrm{d}\boldsymbol{h}}. \tag{9}$$

Proof: See Appendix 1. □

Since ASMC uses a selection of kernels each with a selection of $\boldsymbol{h}$s, it is appropriate as a starting point to look for conditions under which their *distribution* is improved. It would be desirable if, over $\pi^{\star}(\boldsymbol{h}, i)$, the objective function would on average take a higher value, for then the new distribution would on average perform better with respect to Λ than the old. This criterion can be stated in mathematical form: conditions on f are sought for which,

$$\sum_{i\in\mathfrak{I}} \int_{\mathcal{H}} \pi^{\star}(\boldsymbol{h}, i)g(\boldsymbol{h}, i)\mathrm{d}\boldsymbol{h} \geq \sum_{i\in\mathfrak{I}} \int_{\mathcal{H}} \pi(\boldsymbol{h}, i)g(\boldsymbol{h}, i)\mathrm{d}\boldsymbol{h}$$

where

$$g(\boldsymbol{h}, i) = \int \pi(\theta)K_{\boldsymbol{h}}^{(i)}(\theta, \theta')\Lambda(\theta, \theta')\mathrm{d}\theta\mathrm{d}\theta'.$$

Lemma 1. *Assuming* g *is* $\pi(\boldsymbol{h}, i)$*–integrable, in the limit as* $M \to \infty$*,*

$$\begin{aligned} \mathbb{E}_{\pi^{\star}(\boldsymbol{h},i)}[g(\boldsymbol{h}, i)] \geq \mathbb{E}_{\pi(\boldsymbol{h},i)}[g(\boldsymbol{h}, i)] \\ \iff \mathrm{cov}_{\pi(\boldsymbol{h},i)}[g(\boldsymbol{h}, i), w(\boldsymbol{h}, i)] \geq 0. \end{aligned} \tag{10}$$

That is, provided there is positive correlation between the objective function $g(\boldsymbol{h}, i)$ *and the weighting function,* $w(\boldsymbol{h}, i)$*, the new distribution of* $(\boldsymbol{h}, i)$*s will on average perform better (on* $g(\boldsymbol{h}, i)$*) with respect to* Λ *than the old.*

Proof: The result is obtained by expanding definitions in (10):

$$\begin{aligned} \mathbb{E}_{\pi^{\star}(\boldsymbol{h},i)}[g(\boldsymbol{h}, i)] &\geq \mathbb{E}_{\pi(\boldsymbol{h},i)}[g(\boldsymbol{h}, i)], \\ \iff \mathbb{E}_{\pi(\boldsymbol{h},i)}[w(\boldsymbol{h}, i)g(\boldsymbol{h}, i)] &\geq \mathbb{E}_{\pi(\boldsymbol{h},i)}[w(\boldsymbol{h}, i)]\mathbb{E}_{\pi(\boldsymbol{h},i)}[g(\boldsymbol{h}, i)], \\ \iff \mathrm{cov}_{\pi(\boldsymbol{h},i)}[g(\boldsymbol{h}, i), w(\boldsymbol{h}, i)] &\geq 0. \end{aligned}$$

□

Although this result does not directly yield a general form for f, it does give a simple criterion that must be fulfilled by any candidate function. An immediate corollary gives more concrete guidance:

Corollary 1. *A linear weighting scheme,* $f(\Lambda) = a + \Lambda$*, where* $a \geq 0$*, satisfies (10).*

Proof: This is trivially verified using the linearity property of the covariance. □

A consequence of this lemma is that the ASMC algorithm with linear weights will lead to sequential improvement with respect to Λ under very weak assumptions on the target and initial density for $(\boldsymbol{h}, i)$. A linear weighting scheme may at first glance seem sub–optimal, and that it should be possible to learn $(\boldsymbol{h}, i)$ more quickly using a function $f(\Lambda)$ that increases at a super–linear rate. It is conjectured that such functions will not always guarantee an improvement in the distribution of $(\boldsymbol{h}, i)$. For example consider $f(\Lambda) = \Lambda^2$, where the weighting function takes the form, $w(\boldsymbol{h}, i) = g(\boldsymbol{h}, i)^2 + \mathbb{V}[\Lambda|\boldsymbol{H} = \boldsymbol{h}, I = i]$. Because of the $\mathbb{V}[\Lambda|\boldsymbol{H} = \boldsymbol{h}, I = i]$ term, which may be large for values of $(\boldsymbol{h}, i)$ where $g(\boldsymbol{h}, i)$ is small, it is no longer true that $\mathrm{cov}_{\pi(\boldsymbol{h},i)}[g(\boldsymbol{h}, i), w(\boldsymbol{h}, i)] \geq 0$ in general.

4.2 Convergence Over a Number of Iterations

The goal of this section is to provide a theoretical result concerning the ability of ASMC to update the distribution of $(\boldsymbol{h}, i)$s with respect to a sequence of targets, $\pi_1(\theta_1), \ldots, \pi_n(\theta_n)$. To simplify notation, it will be assumed that a move occurs at each iteration of the algorithm. The result can be extended to the case where moves occur intermittently, providing they incur infinitely often in the limit as the number of data points goes to infinity.

Define a set of functions, $\{g^{(t)}(\boldsymbol{h}, i) : \mathcal{A} \to \mathbb{R}_{\geq 0}\}_{t=1}^{n}$,

$$g^{(t)}(\boldsymbol{h}, i) = \int \pi_t(\theta_{t-1}) K_{\boldsymbol{h},t}^{(i)}(\theta_{t-1}, \theta_t) \Lambda(\theta_{t-1}, \theta_t) \mathrm{d}\theta_{t-1} \mathrm{d}\theta_t,$$

where $\mathcal{A} = \mathcal{H} \times \mathfrak{I}$; and for each t, $K_{\boldsymbol{h},t}$ is a π_t–invariant MCMC kernel. We assume for each $(\boldsymbol{h}, i) \in \mathcal{A}$ that $g^{(t)}(\boldsymbol{h}, i)$ is integrable for all t.

For a linear weighting scheme,

$$\pi^{(t)}(\boldsymbol{h}, i) \propto \pi(\boldsymbol{h}, i) \prod_{s=1}^{t} (a + g^{(s)}(\boldsymbol{h}, i)).$$

Below it will be shown that as $t \to \infty$ if the sequence of functions, $\{g^{(t)}(\boldsymbol{h}, i)\}$, converges quickly enough to a fixed function, $g(\boldsymbol{h}, i)$, and if g has a unique global maximum, $(\boldsymbol{h}_{\mathrm{opt}}, i_{\mathrm{opt}})$, then $\pi^{(t)}(h)$ will converge to a point mass on $(\boldsymbol{h}_{\mathrm{opt}}, i_{\mathrm{opt}})$.

The main assumption of this theorem regards the convergence of the sequence of functions $\{g^{(t)}(\boldsymbol{h}, i)\}$, condition (11) below. The existence of a limiting $g(\boldsymbol{h}, i)$ informs the choice of parameterisation for the MCMC kernel. Though this assumption may seem restrictive, the key to understanding the utility of the theorem in practice is that the proposal should be adapted to suit the changing target in some way independent of the tuning parameter. For example, standard Bayesian asymptotics (Ghosal 1999) suggest that if θ_0 is the true parameter value and Σ_t is the posterior variance after t observations, then the posterior for $\Sigma_t^{-1/2}(\theta_t - \theta_0)$ will converge to a standard Gaussian distribution.

In a random walk kernel, for example, the variance should therefore be parameterised as $h^2\hat{\Sigma}_t$, where $\hat{\Sigma}_t$ is an estimate of the posterior variance given t observations. This choice of parametrisation should mean that $g^{(t)}(h)$ converges to the the expected square jump distance for a standard Gaussian target, given RWM with proposal variance $h\mathbb{I}_d$, where $\mathbb{I}_d$ is the $d \times d$ identity matrix. The assumption is also linked to the idea that a good value of $(\boldsymbol{h}, i)$ for the target at time t is required to be a good value at times later on. As mentioned above, the motivation behind SMC is that successive targets should be similar. These issues will be explored empirically in the next section.

Theorem 1. *Let $\pi(\boldsymbol{h}, i) = \pi(\boldsymbol{h}|i)\pi(i)$ be the initial density for the tuning parameter with support $\mathcal{A} = H \times \mathfrak{I}$ and $a > 0$. Define, as above,*

$$\pi^{(t)}(\boldsymbol{h}, i) \propto \pi(\boldsymbol{h}, i) \prod_{s=1}^{t} (a + g^{(s)}(\boldsymbol{h}, i)).$$

Suppose there exists a function (random variable) $g : \mathcal{A} \to \mathbb{R}_{\geq 0}$ such that

$$\sup_{\mathcal{A}} |g^{(t)} - g| \leq k_g t^{-\alpha}, \qquad \alpha \in (0, 1),\ k_g \in \mathbb{R}_{>0}. \tag{11}$$

Furthermore, suppose g has a unique global maximum, $(\boldsymbol{h}_{opt}, i_{opt}) \in \mathcal{A}$, with the property that for i_{opt} there exists an open set around $\boldsymbol{h}_{opt}$ in which g is continuous.

Then as $t \to \infty$, $\pi^{(t)}(\boldsymbol{h}, i)$ tends to a Dirac mass centred on the optimal pair of kernel and associated scaling, $(\boldsymbol{h}_{opt}, i_{opt})$.

Proof: See Appendix 2. □

5 Results

This section is organised as follows. In Section 5.1, the convergence of h to an optimal scaling will be demonstrated empirically using a linear Gaussian model. Then in Section 5.2 the problem of Bayesian mixture analysis will be introduced. In Sections 5.3 and 5.4 the proposed method will be evaluated in simulation studies using the example of Bayesian mixture posteriors as defining the sequence of targets of interest.

As per Sherlock and Roberts (2009), the expected (Mahalanobis) square jumping distance will be considered as an MCMC performance criterion:

$$\Lambda(\theta_{t-1}, \theta_t) = (\theta_{t-1} - \theta_t)^T \hat{\Sigma}_{\pi_t}^{-1} (\theta_{t-1} - \theta_t),$$

where θ_{t-1} and θ_t are two points in the parameter space and $\hat{\Sigma}_{\pi_t}$ is an empirical estimate of the target covariance obtained from the current set of particles.

Two different MCMC kernels will be considered; these are defined by the following two proposals:

$$\begin{aligned} q_{\mathrm{rw}}(\theta_{t-1}, \tilde{\theta}_t) &= \mathcal{N}(\theta_{t-1}, h^2\hat{\Sigma}_{\pi_t}), \\ q_{\mathrm{lw}}(\theta_{t-1}, \tilde{\theta}_t) &= \mathcal{N}(\alpha\theta_{t-1} + (1-\alpha)\bar{\theta}_t, h^2\hat{\Sigma}_{\pi_t}), \end{aligned}$$

where $\bar{\theta}_t$ and $\hat{\Sigma}_{\pi_t}$ are respectively estimates of the target and covariance and in the latter, $h \in (0, 1]$ and $\alpha = \sqrt{1 - h^2}$. The first of these is a *random–walk* proposal. The second is based upon a method for updating parameter values in Liu and West (2001), here named the '*Liu/West*' proposal. The Liu/West proposal has mean shrunk towards the mean of the target and the imposed choice of $\alpha = \sqrt{1 - h^2}$ sets the mean and variance of proposed particles to be the same as that of the current particles. Note that if the target is Gaussian, then this proposal can be shown to be equivalent to a Langevin proposal (Roberts and Tweedie 1996).

5.1 Convergence of h

It is of interest to examine some examples of $g(h)$ and demonstrate convergence of one of the proposed algorithms to the optimal scaling and kernel in the context of a choice between two candidate proposal densities. In this section, we take as an example the $g(h)$ arising from (1) a Gaussian proposal and (2) a t proposal, both exploring a Gaussian target and with mixing criterion Λ being the squared jumping distance.

The results in this section are based on 100 observations simulated from a 5–dimensional standard Gaussian density, $y_{1:100} \overset{\text{iid}}{\sim} \mathcal{N}(0, \mathbb{I}_5)$. The observation variance was assumed to be known and therefore the probability model, or likelihood, was specified as,

$$\pi(y|\theta) = \mathcal{N}(y; \theta, \mathbb{I}_5).$$

The prior on the unknown parameter, θ, the vector of means, was set to $\mathcal{N}(0, 5\mathbb{I}_5)$. ASMC with a random walk proposal was used to generate $M = 2000$ particles from the posterior. We allowed the algorithm to choose between two potential random walk kernels: a multivariate Gaussian or a multivariate t. We fixed the multivariate t random walk to have 3 degrees of freedom, giving the algorithm the option to choose a heavy-tailed proposal (which as will be seen in this case, is sub-optimal). Resampling was invoked when the ESS dropped below $M/2$ and no noise was added to the hs after resampling. The initial distribution for h was chosen to be uniform on $(0, 10)$ for both proposal kernels. We note that for the target in consideration here, the sequence of functions $\{g^{(t)}\}$ for either of the proposal kernels does not change much since each intermediate target is exactly Gaussian and each proposal is scaled by the approximate variance of the target. The optimum scaling for the Gaussian proposal kernel, h_{opt}, was computed using 1-dimensional numerical integration and Theorem 1 of Sherlock and Roberts (2009).

The left plot in Figure 1 shows $g(h)$ for this target explored by the Gaussian proposal kernel (black) and the multivariate t proposal (green). This plot was produced by simulation using standard Metropolis Hastings MCMC for a range of potential values for h; the expected square jumping distance was computed empirically from 10000 samples. The plot shows that of the two proposals, we expect the random walk to be chosen as it has the higher $g(h)$, we further note that the optimal value of h for the Gaussian proposal is slightly larger than the optimal value of h for the t proposal.

The right plot illustrates several features of the adaptive algorithm: the resampling

frequency, when the multivariate t kernel is rejected as sub-optimal, that the algorithm does indeed converge to the true optimal scaling for the Gaussian random walk and the approximate rate of this convergence. Note also that just before rejecting the t proposal, the method had approximately converged to the best scaling in that case as well – slightly lower than the value for the Gaussian proposal, as expected.

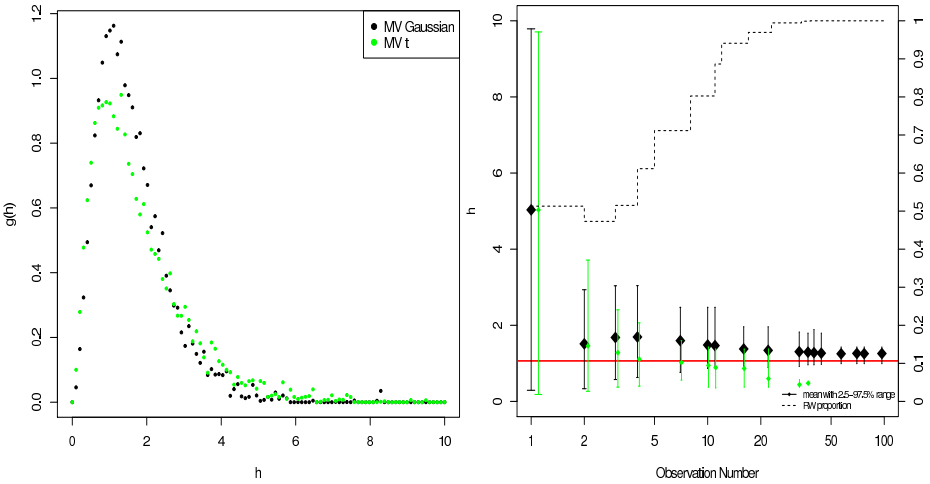

Figure 1: Left plot: $g(h)$ for a 5–dimensional Gaussian target, explored with a random walk Metropolis algorithm (Gaussian proposal in black and t proposal in green) and with ESJD as the optimization criterion. Right plot: convergence of h for the same density based on 100 simulated observations; the horizontal line is the approximately optimal scaling, 1.06; the dashed line indicates at each iteration the proportion of proposal kernels that were Gaussian random walks.

5.2 Bayesian Mixture Analysis

The ability of the ASMC algorithm to learn MCMC tuning parameters in more complicated scenarios is now evaluated using simulated data from a range of mixture likelihoods (for a complete review of this topic, see Frühwirth-Schnatter (2006)). Let $p_1, \ldots, p_r > 0$ be such that $\sum_{i=1}^{r} p_i = 1$. Let $\mathcal{N}(\cdot\,; \mu, v)$ denote the normal density function with mean μ and variance v. Let $\theta = \{p_{1:r-1}, v_{1:r}, \mu_{1:r}\}$.

The likelihood function for a single observation, y_i, is

$$\pi(y_i|\theta) = \sum_{j=1}^{r} p_j \mathcal{N}(y_i; \mu_j, v_j). \tag{12}$$

The prior θ was multivariate normal, on a transformed space using the generalised logit scale for the weights, log scale for variances, and leaving the means untransformed. The components of θ were assumed independent *a priori*; the priors were $\log(p_j/p_r) \sim \mathcal{N}(0, 1^2)$, $\log(v_j) \sim \mathcal{N}(-1.5, 1.3^2)$ and $\mu_j \sim \mathcal{N}(0, 0.75^2)$, where $j = 1, \ldots, r-1$ in

RWfixed	Random walk ordered by means, with h chosen based on the theoretical results for Gaussian targets (Roberts and Rosenthal 2001; Sherlock and Roberts 2009).
RWadaptive	Adaptive random walk MCMC ordered on means with uniform prior on h.
LWmean	Adaptive Liu/West type proposal ordered by means.
LWvariance	Adaptive Liu/West proposal ordered by variances.
Kmix	Adaptive choice between random walk ordered by means, Liu/West ordered using means and a Liu/West ordered on variances.

Table 1: Details of algorithms compared in the simulation study.

the case of the weights and $j = 1, \ldots, r$ for the means and variances. The MCMC moves within the SMC algorithm were performed in the transformed space, with the appropriate inverse transformed values to compute the likelihood in (12).

An issue with mixture models is that for the above choice of prior, the likelihood and posterior are invariant to permutation of the component labels (Stephens 2000). As a result the posterior distribution has a multiple of $r!$ modes, corresponding to each possible permutation. One way of overcoming this problem is by introducing a constraint on the parameters, such as labelling the components so that $\mu_1 < \mu_2 < \cdots < \mu_r$, or so that $v_1 < v_2 < \cdots < v_r$. In the MCMC literature, constraints such as these are often imposed post–processing, see for example Celeux et al. (2000). This choice will affect the empirical moments of the resulting posterior and hence the proposal distribution of the MCMC kernel – both the random walk and Liu/West proposals depend on the posterior covariance, the latter also depending on the mean. In particular if there is a choice of ordering whereby the posterior is closer to Gaussian, then this is likely to lead to better mixing of the MCMC kernels. This phenomenon motivates the idea that it is also possible to choose between orderings on the parameter vector, which will be investigated in the sequel.

5.3 Details of Implementation of ASMC

In analysing the simulated data, a number of SMC and ASMC algorithms were compared. These correspond to using the MCMC kernels shown in Table 1. In each case the reference to ordering relates to how the component labels were defined, and thus affect the estimate of the posterior mean and covariance used.

The above methods were also compared with the adaptive MCMC algorithm of Haario et al. (1998), denoted **AMCMC**. The specific implementation is as follows. The prior densities were identical to those for ASMC, the parameter vector was ordered by means and the random walk tuning was computed using the approximately optimal Gaussian scaling given by $h = 2.4/\sqrt{3r-1}$. AMCMC was run for 12000 iterations for the 5 dimensional datasets (datasets 1–4 in Section 5.4) and for 30000 iterations for the

8 dimensional datasets (datasets 5 and 6 in Section 5.4): these values were chosen so as to approximately match the number of likelihood computations involved between the ASMC and AMCMC methods. The burn–in period was set to half of the number of iterations and the method was initialised by a draw from the prior. There was an initial non–adaptive phase, lasting 1000 iterations, where the proposal kernel was scaled by the prior covariance and after which scaling was via estimates of the posterior covariance computed from the chain to–date, this was updated every 100 iterations.

For the ASMC algorithms, the initial distribution of hs was chosen to be uniform on $(0, 2]$ for the random walk and on $(0, 1]$ for the Liu/West proposal. In the case of the random walk, this range of hs can be justified by considering the optimal scaling for a random walk Metropolis on a multivariate Gaussian target in 5 dimensions namely $2.38/\sqrt{5} = 1.06$ (and decays with increasing dimension as $O(d^{-1/2})$). For the Liu/West, h must be in $(0, 1]$.

In each case we chose R (see Equation 5) to be a Gaussian kernel with variance 0.015^2. A sensitivity analysis showed that changing the variance of the noise slightly did not affect the conclusions of this research. The parameter for the linear h-weighting scheme was $a = 0$. If any h was perturbed below zero, a small value, 1×10^{-6}, was imputed and similarly for the Liu/West approach, any h perturbed above 1 was replaced by 1.

The number of particles was set to $M = 2000$ for the 2–mixture datasets and $M = 5000$ for the 3–mixture datasets. Each algorithm was run 100 times on each dataset with the order of observations randomised each time. For the MCMC based methods an ESS tolerance of $M/2$ was used, as in Jasra et al. (2007). Resampling of the particles was via residual sampling (Whitley 1994; Liu et al. 1998), but multinomial sampling was used in selecting hs. For ease of computing posterior quantities of interest, each of the above algorithms was forced to resample and move on the last iteration.

To compare the performance of different methods, a measure of the accuracy of the estimated predictive density was used. This is advantageous because it is invariant to re–labelling of the mixture components – the alternative of comparing the accuracy of posterior marginals is compromised by the so-called label switching problem. The chosen accuracy measure was the variability of the predictive density (VPD) and was calculated as follows. Each run of the algorithm produces a weighted particle set, from which an estimate of $\mathbb{E}[\pi(y^{(i)}|y_{1:n})]$ can be obtained at 100 points, $\{y^{(i)} : i = 1, \ldots, 100\}$, equi-spaced between -2.5 and 2.5. For each i, the 100 simulation runs produce 100 realisations of $\mathbb{E}[\pi(y^{(i)}|y_{1:n})]$; let $\hat{y}^{(i,j)}$ be the estimate of $y^{(i)}$ obtained from run j. The VPD measure used in this paper is

$$\text{mean}_i[\text{var}_j(\hat{y}^{(i,j)})],$$

where mean_i is the mean over the is and var_j is the variance of the estimates of $y^{(i)}$ obtained from the 100 simulations. The VPD gives an indication of the global variability of the predictive density across the simulations. In the tables, the relative VPD is used, which gives a scale–free comparison between methods. The SMC/ASMC algorithm with a relative VPD of 1 is the reference algorithm and has the smallest VPD of the

Dataset 1:	$0.5\mathcal{N}(y; -0.25, 0.5^2) + 0.5\mathcal{N}(y; 0.25, 0.5^2)$
Dataset 2:	$0.5\mathcal{N}(y; 0, 1^2) + 0.5\mathcal{N}(y; 0, 0.1^2)$
Dataset 3:	$0.3\mathcal{N}(y; -1, 0.5^2) + 0.7\mathcal{N}(y; 1, 0.5^2)$
Dataset 4:	$0.5\mathcal{N}(y; -0.75, 0.1^2) + 0.5\mathcal{N}(y; 0.75, 0.1^2)$
Dataset 5:	$0.35\mathcal{N}(y; -0.1, 0.1^2) + 0.3\mathcal{N}(y; 0, 0.5^2) + 0.35\mathcal{N}(y; 0.1, 1^2)$
Dataset 6:	$0.25\mathcal{N}(y; -0.5, 0.1^2) + 0.5\mathcal{N}(y; 0, 0.2^2) + 0.25\mathcal{N}(y; 0.5, 0.1^2)$

Table 2: Details of likelihoods, $\pi(y|\boldsymbol{\theta})$, in the simulation study.

SMC/ASMC methods; larger values indicate higher VPDs. For the AMCMC methods, the predictive densities were computed using all available samples, i.e., with 6000 for the 2–mixture datasets and 15000 for the 3–mixture datasets. For the SMC/ASMC methods a Rao–Blackwellised version of the predictive density was computed using all current and proposed particles available from the last iteration (that is, using 4000/10000 sample points respectively for the 2/3–mixture datasets).

5.4 Results

100 realisations were simulated from the following likelihoods in Table 2. This choice of datasets in combination with the selection of MCMC kernels allows several hypotheses to be tested empirically. Firstly, by comparing the performance of RWfixed with RWadaptive in these cases, it is possible to see whether anything is lost or gained by adapting the proposal kernel. Secondly, the impact of the different kernel orderings on MCMC mixing will become apparent by considering the performance of LWmean and LWvariance in these settings. Datasets 3, 4 and 6 have well 'separated' means and similar variances, so one might expect algorithms ordering by means to perform better; whereas datasets 2 and 5 have well separated variances and similar means, so perhaps the algorithms ordering by variances might do well here. Thirdly, the Kmix algorithm should be able to choose the best ordering and it is of interest to compare the results from this algorithm with an adaptive version of the individual kernels.

The simulation results from these datasets are presented in Table 3. These give both the relative VPD for each method, but also an estimated mean ESJD for each method.

The mean number of likelihood evaluations for the SMC algorithms in datasets 1–6 were respectively: 5.65×10^5, 9.17×10^5, 9.18×10^5, 9.12×10^5, 2.65×10^6, 2.40×10^6; there was little variability between the individual algorithms. In comparison, AMCMC used 1.2×10^5 likelihood evaluations for datasets 1–4 and 3×10^6 likelihood evaluations for the others.

As would be hoped, a very strong correlation between lower VPD and higher ESJD is evident for the SMC/ASMC algorithms. This empirically supports the use of ESJD as the chosen criterion for adapting the MCMC kernels.

There is relatively little difference across scenarios between the fixed and adaptive random walk methods. Furthermore, the adaptive random walk settles on a similar

Table 3: Rel. VPD is relative VPD, JD is the mean square jumping distance, Acc is the mean final acceptance probability, h is the mean final scaling by kernel and Propn is the mean final kernel proportions. The kernels 'LWm' and 'LWv' indicate respectively a Liu/West proposal ordering on means or variances.

Method	Rel. VPD	JD	Acc.	h	Propn
Dataset 1					
LWvariance	1	1.869	0.3	0.941	
LWmean	1.189	1.818	0.32	0.956	
Kmix	1.258	1.845	0.317	LWm 0.963 LWv 0.958	LWm 0.785 LWv 0.215
RWadaptive	2.391	0.708	0.21	0.946	
AMCMC	2.396	0.575	0.13	1.073	
RWfixed	3.414	0.641	0.18	1.064	
Dataset 2					
LWvariance	1	9.139	0.873	0.978	
Kmix	2.843	9.023	0.854	LWm 0.984 LWv 0.978	LWm 0.005 LWv 0.995
AMCMC	28.333	0.197	0.019	1.073	
LWmean	112.23	1.869	0.129	0.969	
RWadaptive	188.094	0.77	0.134	0.584	
RWfixed	219.907	0.596	0.041	1.064	
Dataset 3					
LWmean	1	6.38	0.792	0.98	
Kmix	1.54	6.378	0.806	LWm 0.979	LWm 1
AMCMC	7.465	0.847	0.146	1.073	
RWfixed	40.538	1.124	0.277	1.064	
RWadaptive	45.739	1.057	0.369	1.045	
LWvariance	148.827	0.737	0.064	0.966	
Dataset 4					
LWmean	1	7.132	0.875	0.98	
Kmix	1.099	7.127	0.877	LWm 0.979	LWm 1
AMCMC	24.024	0.462	0.057	1.073	
RWadaptive	48.606	1.143	0.274	1.086	
RWfixed	51.919	1.167	0.298	1.064	
LWvariance	1096.167	0.632	0.027	0.961	
Dataset 5					
AMCMC	0.883	0.356	0.04	0.849	
Kmix	1	2.258	0.234	LWm 0.964 LWv 0.971	LWm 0.044 LWv 0.956
LWvariance	1.151	2.284	0.183	0.971	
LWmean	2.792	1.007	0.092	0.961	
RWadaptive	4.923	0.847	0.205	0.435	
RWfixed	5.187	0.56	0.055	0.84	
Dataset 6					
LWmean	1	4.099	0.277	0.972	
Kmix	1.018	3.994	0.363	LWm 0.973	LWm 1
AMCMC	1.556	0.211	0.04	0.849	
RWfixed	3.244	0.996	0.429	0.84	
RWadaptive	3.259	0.93	0.192	0.693	
LWvariance	3.951	1.951	0.13	0.944	

scaling as the fixed scaled version in datasets 3 and 4, whereas in datasets 1, 2, 5 and 6, RWadaptive settles to values below RWfixed. In datasets 1, 2, 4 and 5, the adaptive RW outperformed the fixed equivalent (though the difference was negligible in datasets 4 and 5); this is likely due to the fact that the covariance was not a good estimate and the adaptive version of the algorithm was able to rescale to compensate for this. In datasets 3 and 6, the fixed random walk marginally outperformed the adaptive.

The 'correctly ordered' sequential Liu/West algorithms considerably outperform those using RW kernels in all six datasets and the incorrectly ordered versions perform worse or as poorly as the RW. For the Liu/West proposals, the h selected in each dataset was very close to 1: this special value corresponds to an independence kernel in the form of a moment–matched Gaussian approximation of the target. This is of interest as, in combination with the high acceptance rates of between 80–87% in datasets 2–4, it suggests that the 'correct' ordering makes the target, ostensibly a very *complex* density function, approximately Gaussian in these cases.

The Kmix algorithm is able to choose between orderings; the advantages of this are clearly evidenced in the results, as it selects the best ordering in each case, with the exception of dataset 1 (where the means and variances are both similar). The Kmix sampler settles almost unanimously on one ordering above the others. These results show empirically that there is not much difference in using a single (correctly chosen) kernel compared with using a selection of kernels.

The performance of AMCMC was surpassed in all cases by the Kmix algorithm excepting dataset 5, where AMCMC was the best performing algorithm. In this latter case and in dataset 6, neither AMCMC nor the SMC/ASMC algorithms performed well. AMCMC outperformed RWadaptive in each case apart from dataset 1, where the difference was small. However, the results show the average jumping distance of the kernel used in the ASMC algorithm was greater than that of AMCMC in all cases, suggesting ASMC is able to adapt better to well-mixing kernels. To make this comparison more clear, two MCMC algorithms were run on each data-set, one using the final kernel found by AMCMC and one using a kernel based on the ASMC run, with the final estimated covariance matrix and the final mean value of the tuning parameter. The resulting MCMC algorithms performed very similarly in 3 cases (VPD of the two MCMC algorithms within 10% of each other) and the kernel found by ASMC performed better in the other 3 (VPD reduced by 30%, 40% and 80%).

The `R` and `C` functions for the mixture example are available from the corresponding author.

6 Discussion

This paper introduces a new method for automatically tuning and choosing between different MCMC kernels. Where MCMC based SMC code already exists, adapting the hs would be a relatively straightforward means of enhancing performance, the main effort being in calculating the ratio of the proposed particles in the accept/reject step.

Probably the most important conclusion from the simulation studies presented is that there is not much lost in terms of performance in the adaption process – the Kmix algorithm performed comparably to the respective best performing individual component and the adaptive random walk Metropolis performed similarly to the fixed, approximately optimally scaled version.

Although the method as presented has assumed that i.i.d. observations are available from the likelihood, the ASMC algorithm readily extends to the case of a dependent sequence. In this case the tth target density is given by,

$$\pi_t \propto \pi(\theta)\pi(y_1|\theta)\prod_{i=2}^{t}\pi(y_i|y_{1:i-1},\theta), \tag{13}$$

the extension to general sequences of target densities, $\{\pi_i\}_{i=1}^n$, including (13), being immediate and implied by the choice of notation in Algorithm 3.

The main assumption of ASMC is that a good h at time t is likely also to perform well at time $t+1$. One piece of evidence that supports this assumption is that the resampling frequency decreases with an increasing number of observations (Chopin 2002). This implies that, although π_1 and π_2 may be quite different, π_{1001} and π_{1002} are likely to be less so, provided that the data provides sufficient information on the parameters. As mentioned earlier in the text, the assumption of similar successive target densities is also required for the efficiency of the non–adaptive version (Chopin 2002; Del Moral et al. 2006).

ASMC can be easily extended by considering other proposal densities. For example it is possible to formulate a t–distributed version of the Liu/West proposal, allowing for heavier tailed proposals, the heaviness of which can be selected automatically by adaptively choosing the number of degrees of freedom; this t–based proposal includes the Liu/West as a special case. Other interesting algorithms can be formulated using DE proposals (Ter Braak 2006) (which generalises the snooker algorithm of Gilks et al. (1994)) or regional MCMC proposals (Roberts and Rosenthal 2009; Craiu et al. 2009) – both of which appeal strongly to the particle structure of the new method.

References

Andrieu, C. and Robert, C. (2001). "Controlled MCMC for Optimal Sampling." Technical report, Université Paris–Dauphine. 412, 414

Andrieu, C. and Thoms, J. (2008). "A tutorial on adaptive MCMC." *Statistics and Computing*, 18(4): 343–373. 412, 414, 416, 419

Atchadé, Y. and Rosenthal, J. (2005). "On adaptive Markov chain Monte Carlo algorithms." *Bernoulli*, 11(5): 815–828. 414

Cappé, O., Douc, R., Guillin, A., Marin, J.-M., and Robert, C. P. (2008). "Adaptive importance sampling in general mixture classes." *Statistics and Computing*, 18(4): 447–459. 412

Carpenter, J., Clifford, P., and Fearnhead, P. (1999). "Improved particle filter for nonlinear problems." *Radar, Sonar and Navigation, IEEE Proceedings*, 146(1): 2 –7. 417

Celeux, G., Hurn, M., and Robert, C. P. (2000). "Computational and Inferential Difficulties with Mixture Posterior Distributions." *Journal of the American Statistical Association*, 95(451): 957–970. 426

Chopin, N. (2002). "A sequential particle filter method for static models." *Biometrika*, 89(3): 539–552. 411, 415, 418, 431

Cornebise, J., Moulines, E., and Olsson, J. (2008). "Adaptive methods for sequential importance sampling with application to state space models." *Statistics and Computing*, 18(4): 461–480. 412

Craiu, R. V., Rosenthal, J., and Yang, C. (2009). "Learn From Thy Neighbor: Parallel-Chain and Regional Adaptive MCMC." *Journal of the American Statistical Association*, 104(488): 1454–1466. 412, 414, 431

Crisan, D. (2001). "Particle Filters – A Theoretical perspective." In *Sequential Monte Carlo methods in practice*, chapter 2, 17–42. Springer. 420

Del Moral, P. (2004). *Feynman-Kac Formulae. Genealogical and interacting particle systems with applications*. Springer. 420

Del Moral, P., Doucet, A., and Jasra, A. (2006). "Sequential Monte Carlo samplers." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3): 411–436. 411, 414, 415, 418, 431

— (2010). "An Adaptive Sequential Monte Carlo Method for Approximate Bayesian Computation."
URL http://www.cs.ubc.ca/%7Earnaud/delmoral_doucet_jasra_smcabc.pdf
416

Douc, R., Guillin, A., Marin, J.-M., and Robert, C. P. (2007). "Minimum variance importance sampling via Population Monte Carlo." *ESAIM: Probability and Statistics*, 11: 427–447. 412

Doucet, A., de Freitas, N., and Gordon, N. (eds.) (2001). *Sequential Monte Carlo Methods in Practice*. Springer–Verlag New York. 411, 414

Fearnhead, P. (2002). "MCMC, sufficient statistics and particle filters." *Journal of Computational and Graphical Statistics*, 11: 848–862. 411

— (2008). "Computational Methods for Complex Stochastic Systems: A Review of Some Alternatives to MCMC." *Statistics and Computing*, 18: 151–171. 411

Frühwirth-Schnatter, S. (2006). *Finite Mixture and Markov Switching Models*. Springer. 425

Gamerman, D. and Lopes, H. F. (2006). *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference (2nd ed.)*. Chpman & Hall/CRC. 413

Ghosal, S. (1999). "Asymptotic Normality of Posterior Distributions in High Dimensional Linear Models." *Bernoulli*, 5(2): 315–331. 422

Gilks, W. and Berzuini, C. (1999). "Following a moving target – Monte Carlo inference for dynamic Bayesian models." *Journal of the Royal Statistical Society, Series B*, 63(1): 127–146. 411, 415

Gilks, W., Richardson, S., and Spiegelhalter, D. (eds.) (1995). *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC. 413

Gilks, W. R., Roberts, G. O., and George, E. I. (1994). "Adaptive Direction Sampling." *Journal of the Royal Statistical Society. Series D (The Statistician)*, 43(1): 179–189. 431

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). "Novel approach to nonlinear/non-Gaussian Bayesian state estimation." *Radar and Signal Processing, IEEE Proceedings F*, 140(2): 107–113. 411

Haario, H., Saksman, E., and Tamminen, J. (1998). "An Adaptive Metropolis algorithm." *Bernoulli*, 7: 223–242. 411, 412, 414, 426

Hastings, W. K. (1970). "Monte Carlo sampling methods using Markov chains and their applications." *Biometrika*, 57(1): 97–109. 413, 414

Jasra, A., Doucet, A., Stephens, D. A., and Holmes, C. C. (2008a). "Interacting sequential Monte Carlo samplers for trans-dimensional simulation." *Computational Statistics & Data Analysis*, 52(4): 1765–1791. 412

Jasra, A., Stephens, D. A., Doucet, A., and Tsagaris, T. (2008b). "Inference for Levy driven Stochastic Volatility Models via Adaptive SMC." http://www.theodorostsagaris.com/svvg-DAS.pdf. 412

Jasra, A., Stephens, D. A., and Holmes, C. C. (2007). "On population-based simulation for static inference." *Statistics and Computing*, 17(3): 263–279. 412, 427

Jennison, C. and Sheehan, N. (1995). "Theoretical and Empirical Properties of the Genetic Algorithm as a Numerical Optimizer." *Journal of Computational and Graphical Statistics*, 4(4): 296–318. 418

Kitagawa, G. (1996). "Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models." *Journal of Computational and Graphical Statistics*, 5(1): 1–25. 417

Kong, A., Liu, J. S., and Wong, W. H. (1994). "Sequential Imputations and Bayesian Missing Data Problems." *Journal of the American Statistical Association*, 89(425): 278–288. 415

Liu, J. and West, M. (2001). *Sequential Monte Carlo Methods in Practice*, chapter 10: Combined Parameter and State Estimation in Simulation-Based Filtering. Springer–Verlag New York. 411, 418, 424

Liu, J. S. and Chen, R. (1995). "Blind Deconvolution Via Sequential Imputations." *Journal of the American Statistical Association*, 90: 567–576. 415

— (1998). "Sequential Monte Carlo Methods for Dynamic Systems." *Journal of the American Statistical Association*, 93(443): 1032–1044. 411, 415

Liu, J. S., Chen, R., and Wong, W. H. (1998). "Rejection Control and Sequential Importance Sampling." *Journal of the American Statistical Association*, 93(443): 1022–1031. 415, 417, 427

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). "Equation of State Calculations by Fast Computing Machines." *The Journal of Chemical Physics*, 21(6): 1087–1092. 413, 414

Neal, R. (2001). "Annealed Importance Sampling." *Statistics and Computing*, 11(2): 125–139. 411, 412, 415

Pasarica, C. and Gelman, A. (2010). "Adaptively scaling the Metropolis algorithm using expected squared jumped distance." *Statistica Sinica*, 20: 343–364. 417

Roberts, G. and Rosenthal, J. (2001). "Optimal Scaling for Various Metropolis-Hastings Algorithms." *Statistical Science*, 16(4): 351–367. 412, 414, 416, 426

Roberts, G. O. and Rosenthal, J. S. (2009). "Examples of adaptive MCMC." *Journal of Computational and Graphical Statistics*, 18(2): 349–367. 412, 414, 431

Roberts, G. O. and Tweedie, R. L. (1996). "Exponential Convergence of Langevin Distributions and Their Discrete Approximations." *Bernoulli*, 2(4): 341–363. 424

Schäfer, C. and Chopin, N. (2013). "Sequential Monte Carlo on large binary sampling spaces." *Statistics and Computing*, 23: 163–184. 412

Sherlock, C. and Roberts, G. (2009). "Optimal scaling of the random walk Metropolis on elliptically symmetric unimodal targets." *Bernoulli*, 15(3): 774–798. 412, 414, 416, 417, 423, 424, 426

Stephens, M. (2000). "Dealing with label switching in mixture models." *Journal of the Royal Statistical Society, Series B*, 62(4): 795–809. 426

Storvik, G. (2002). "Particle filters for state-space models with the presence of unknown static parameters." *IEEE Transactions on Signal Processing*, 50: 281–289. 411

Ter Braak, C. J. F. (2006). "A Markov Chain Monte Carlo version of the genetic algorithm Differential Evolution: easy Bayesian computing for real parameter spaces." *Statistics and Computing*, 16(3): 239–249. 431

West, M. (1993). "Mixture models, Monte Carlo, Bayesian updating and dynamic models." *Computing Science and Statistics*, 24: 325–333. 418

Whitley, D. (1994). "A genetic algorithm tutorial." *Statistics and Computing*, 4: 65–85. 417, 427

Appendix 1: Proof of Proposition 1

Recall that we use π to denote a probability density. Let $\Lambda^{(j)} = \Lambda(\theta^{(j)}, \theta'^{(j)})$, i.e., the observed Λ for the jth particle and $\mathbb{I}$ denote the indicator function. The collection $\{(\boldsymbol{h}^{(j)}, i^{(j)}), 1/M\}_{j=1}^{M}$ is an iid sample from $\pi(\boldsymbol{H}, I)$. Let

$$W_j = \frac{f(\Lambda^{(j)})}{\sum_{i=1}^{M} f(\Lambda^{(i)})}$$

be a set of weights and define a discrete random variable $(\boldsymbol{H}^\star, I^\star)$, which takes value $(\boldsymbol{h}^{(j)}, i^{(j)})$ with probability W_j. For any $\mathcal{B} = \mathcal{H}_\mathcal{B} \times \mathfrak{I}_\mathcal{B} \subseteq \mathcal{H} \times \mathfrak{I}$,

$$\begin{aligned} \mathbb{P}\mathrm{r}[(\boldsymbol{H}^\star, I^\star) \in \mathcal{B}] &= \sum_{j=1}^{M} W_j \mathbb{I}[(\boldsymbol{h}^{(j)}, i^{(j)}) \in \mathcal{B}], \\ &= \frac{\frac{1}{M}\sum_{j=1}^{M} f(\Lambda^{(j)}) \mathbb{I}[(\boldsymbol{h}^{(j)}, i^{(j)}) \in \mathcal{B}]}{\frac{1}{M}\sum_{i=1}^{M} f(\Lambda^{(i)})}. \end{aligned}$$

Now we wish to use the strong law of large numbers for the numerator and denominator. For this we need to know the expectations of these, which can be calculated using the properties of conditional expectation in terms of $w^{[M]}(\boldsymbol{h}^{(j)}, i^{(j)})$, with for example,

$$\mathbb{E}_{\pi^{[M]}(\theta)K(\theta,\theta')}\{f(\Lambda)\mathbb{I}[(\boldsymbol{H}^\star, I^\star) \in \mathcal{B}]\} = \mathbb{E}_{\boldsymbol{H}^\star, I^\star}\{w^{[M]}(\boldsymbol{H}, I)\mathbb{I}[(\boldsymbol{H}, I) \in \mathcal{B}]\}.$$

Thus in the limit as $M \to \infty$, using also that $w^{[M]}(\boldsymbol{h}, i) \to w(\boldsymbol{h}, i)$ we get

$$\begin{aligned} &\left| \frac{\frac{1}{M}\sum_{j=1}^{M} f(\Lambda^{(j)}) \mathbb{I}[(\boldsymbol{h}^{(j)}, i^{(j)}) \in \mathcal{B}]}{\frac{1}{M}\sum_{i=1}^{M} f(\Lambda^{(i)})} - \frac{\sum_{k\in\mathfrak{I}_\mathcal{B}} \int_{\boldsymbol{s}\in\mathcal{H}_\mathcal{B}} w(\boldsymbol{s}, k)\pi(\boldsymbol{s}, k)\mathrm{d}\boldsymbol{s}}{\sum_{i\in\mathfrak{I}} \int_{\boldsymbol{h}\in\mathcal{H}} w(\boldsymbol{h}, i)\pi(\boldsymbol{h}, i)\mathrm{d}\boldsymbol{h}} \right| \\ \leq &\left| \frac{\frac{1}{M}\sum_{j=1}^{M} f(\Lambda^{(j)}) \mathbb{I}[(\boldsymbol{h}^{(j)}, i^{(j)}) \in \mathcal{B}]}{\frac{1}{M}\sum_{i=1}^{M} f(\Lambda^{(i)})} - \frac{\sum_{k\in\mathfrak{I}_\mathcal{B}} \int_{\boldsymbol{s}\in\mathcal{H}_\mathcal{B}} w^{[M]}(\boldsymbol{s}, k)\pi(\boldsymbol{s}, k)\mathrm{d}\boldsymbol{s}}{\sum_{i\in\mathfrak{I}} \int_{\boldsymbol{h}\in\mathcal{H}} w^{[M]}(\boldsymbol{h}, i)\pi(\boldsymbol{h}, i)\mathrm{d}\boldsymbol{h}} \right| \\ + &\left| \frac{\sum_{k\in\mathfrak{I}_\mathcal{B}} \int_{\boldsymbol{s}\in\mathcal{H}_\mathcal{B}} w^{[M]}(\boldsymbol{s}, k)\pi(\boldsymbol{s}, k)\mathrm{d}\boldsymbol{s}}{\sum_{i\in\mathfrak{I}} \int_{\boldsymbol{h}\in\mathcal{H}} w^{[M]}(\boldsymbol{h}, i)\pi(\boldsymbol{h}, i)\mathrm{d}\boldsymbol{h}} - \frac{\sum_{k\in\mathfrak{I}_\mathcal{B}} \int_{\boldsymbol{s}\in\mathcal{H}_\mathcal{B}} w(\boldsymbol{s}, k)\pi(\boldsymbol{s}, k)\mathrm{d}\boldsymbol{s}}{\sum_{i\in\mathfrak{I}} \int_{\boldsymbol{h}\in\mathcal{H}} w(\boldsymbol{h}, i)\pi(\boldsymbol{h}, i)\mathrm{d}\boldsymbol{h}} \right| \\ \to &\ 0, \end{aligned}$$

as required. □

Appendix 2: Proof of Theorem 1

The of this theorem proceeds in two parts. We start by observing that $\pi^{(n)}(\boldsymbol{h}, i) = \pi(\boldsymbol{h}, i)\exp\{nf_n\}$ where,

$$f_n(\boldsymbol{h}, i) = \frac{1}{n}\sum_{t=1}^{n}\log(a + g^{(t)}(\boldsymbol{h}, i)).$$

In the first part, the following results will be proved:

- There exists a function, $f : \mathcal{A} \to \mathbb{R}_{\geq 0}$, such that $\sup_{(\boldsymbol{h},i)\in\mathcal{A}} |f_n - f| \leq k_f n^{-\alpha}$.
- $(\boldsymbol{h}_{\text{opt}}, i_{\text{opt}})$ is the unique global maximum of f.
- There exists an open set of $\boldsymbol{h}$ around $\boldsymbol{h}_{\text{opt}}$ in which $f(\boldsymbol{h}_{\text{opt}}, i_{\text{opt}})$ is continuous.

In the second part of the proof, these results will be used to show that as $n \to \infty$, $\pi^{(n)}(\boldsymbol{h}, i)$ approaches a Dirac mass centred on $(\boldsymbol{h}_{\text{opt}}, i_{\text{opt}})$.

Part 1

Claim that $f(\boldsymbol{h}, i) = \log(a + g(\boldsymbol{h}, i))$. It is easy to show that as $g(\boldsymbol{h}, i) \geq 0$,

$$\sup_{(\boldsymbol{h},i)\in\mathcal{A}} |(a + g^{(t)})/(a + g) - 1| \leq k_l t^{-\alpha},$$

where $k_l = k_g/a$.

Using $\log(x) \leq x - 1$ we have, that for any $(\boldsymbol{h}, i) \in \mathcal{A}$

$$\log\left\{\frac{a + g^{(t)}}{a + g}\right\} \leq \frac{a + g^{(t)}}{a + g} - 1 \leq k_l t^{-\alpha}.$$

Put $k_m = 2k_l$ and $c = -(1/2)\log(1/2)$, noting that $c \in (0, 1)$. Since the function $\gamma(x) = 1 - x - \exp\{-2x\}$ is increasing on $[0, c]$, we have $\gamma(x) \geq 0$ on this interval, as $\gamma(0) = 0$. Hence by re-arranging and taking logs in the inequality $\gamma(x) \geq 0$, we have $\log(1 - x) \geq -2x$ for any $x \in [0, c]$ and so provided that $t > (k_l/c)^{1/\alpha}$, for all $(\boldsymbol{h}, i) \in \mathcal{A}$,

$$\log\left\{\frac{a + g^{(t)}}{a + g}\right\} \geq \log(1 - k_l t^{-\alpha}) \geq -2k_l t^{-\alpha} = -k_m t^{-\alpha}.$$

The preceding arguments show that for all $t > (k_l/c)^{1/\alpha}$,

$$\sup_{(\boldsymbol{h},i)\in\mathcal{A}}\left|\log\left\{\frac{a + g^{(t)}}{a + g}\right\}\right| = \sup_{(\boldsymbol{h},i)\in\mathcal{A}} |\log(a + g^{(t)}) - \log(a + g)| \leq k_m t^{-\alpha}.$$

Put $t^\star = \lceil (k_l/c)^{1/\alpha} \rceil$ and $c_{t^\star} = \sum_{t=1}^{t^\star - 1} \left| \log(a + g^{(t)}) - \log(a+g) \right| < \infty$ then for all $(\boldsymbol{h}, i) \in \mathcal{A}$,

$$\begin{aligned}
|f_n - \log(a+g)| &\leq \frac{1}{n} \sum_{t=1}^{n} \left| \log(a + g^{(t)}) - \log(a+g) \right|, \\
&\leq \frac{c_{t^\star}}{n} + \frac{k_m}{n} \sum_{t=t^\star}^{n} t^{-\alpha}, \\
&\leq \frac{c_{t^\star}}{n} + \frac{k_m}{n} \int_0^n t^{-\alpha} \mathrm{d}t, \\
&= \frac{c_{t^\star}}{n} + \frac{k_m}{1-\alpha} n^{-\alpha}, \\
&= c_{t^\star} n^{-\alpha} + \frac{k_m}{1-\alpha} n^{-\alpha}, \qquad \text{since } 0 < \alpha < 1, \\
&< k_f n^{-\alpha},
\end{aligned}$$

where $k_f = c_{t^\star} + k_m/(1-\alpha)$ as required.

The continuity and strict monotonicity of the logarithm and the assumptions on g imply that $(\boldsymbol{h}_{\text{opt}}, i_{\text{opt}})$ is the unique global maximum of f and also that for i_{opt} there exists an open set around $\boldsymbol{h}_{\text{opt}}$ in which f is continuous.

Part 2

In this part, the properties of f will be used to show that for any set containing $(\boldsymbol{h}_{\text{opt}}, i_{\text{opt}})$ as $n \to \infty$, the probability that $(\boldsymbol{H}, I)$ belongs to that set tends to 1.

By the uniqueness of $(\boldsymbol{h}_{\text{opt}}, i_{\text{opt}})$ and local continuity of f about this point, there exists an $\epsilon > 0$ such that for all $i \neq i_{\text{opt}}$

$$f(\boldsymbol{h}_{\text{opt}}, i_{\text{opt}}) - \max_{\boldsymbol{h}} f(\boldsymbol{h}, i) > \epsilon.$$

Let $\bar{\mathcal{X}}$ denote the complement of $\mathcal{X}$ in $\mathcal{A}$. Let $\mathcal{H}_0 \subset \mathcal{H}$ be any set containing $\boldsymbol{h}_{\text{opt}}$. By virtue of the global uniqueness of $(\boldsymbol{h}_{\text{opt}}, i_{\text{opt}})$ and continuity of $f(\,\cdot\,, i_{\text{opt}})$ around $\boldsymbol{h}_{\text{opt}}$, there exists an open set $\mathcal{H}_1 \subset \mathcal{H}_0$ also containing $\boldsymbol{h}_{\text{opt}}$ on which $g(\,\cdot\,, i_{\text{opt}})$ is concave and with the property, $\inf_{\boldsymbol{h} \in \mathcal{H}_1} f(\boldsymbol{h}, i_{\text{opt}}) \geq \sup_{(\boldsymbol{h}) \in \bar{\mathcal{H}}_1} f(\boldsymbol{h}, i_{\text{opt}})$. Such a set exists precisely because $g(\,\cdot\,, i_{\text{opt}})$ is locally continuous about the global maximum.

For any $\mathcal{H}_1$ we can define an $\mathcal{H}_2 \subset \mathcal{H}_1$ such that there exists $\epsilon_1 < \epsilon_2$ with

$$\begin{aligned}
\sup_{\boldsymbol{h} \in \mathcal{H}_2} \{ f(\boldsymbol{h}_{\text{opt}}, i_{\text{opt}}) - f(\boldsymbol{h}, i_{\text{opt}}) \} &= \epsilon_1, \\
\inf_{\boldsymbol{h} \in \bar{\mathcal{H}}_1} \{ f(\boldsymbol{h}_{\text{opt}}, i_{\text{opt}}) - f(\boldsymbol{h}, i_{\text{opt}}) \} &= \epsilon_2.
\end{aligned}$$

Furthermore for any $\mathcal{H}_0$ we can choose $\mathcal{H}_1$ and $\mathcal{H}_2$ such that $\epsilon > \epsilon_2$.

Consider the probability of $\boldsymbol{H} \in \mathcal{H}_0$ and $I = i_{\text{opt}}$ after n updates,

$$\begin{aligned}
\mathbb{P}\text{r}[\boldsymbol{H} \in \mathcal{H}_0 \cap I = i_{\text{opt}}] &> \mathbb{P}\text{r}[\boldsymbol{H} \in \mathcal{H}_1 \cap I = i_{\text{opt}}] \\
&= \frac{\int_{\mathcal{H}_1} \pi^{(n)}(\boldsymbol{h}, i_{\text{opt}}) \mathrm{d}\boldsymbol{h}}{\sum_{i \in \mathfrak{I}} \int_{\mathcal{H}} \pi^{(n)}(\boldsymbol{h}, i) \mathrm{d}\boldsymbol{h}}, \\
&= \frac{\int_{\mathcal{H}_1} \pi(\boldsymbol{h}, i_{\text{opt}}) \exp\{n f_n(\boldsymbol{h}, i_{\text{opt}})\} \mathrm{d}\boldsymbol{h}}{\sum_{i \in \mathfrak{I}} \int_{\mathcal{H}} \pi(\boldsymbol{h}, i) \exp\{n f_n(\boldsymbol{h}, i)\} \mathrm{d}\boldsymbol{h}}.
\end{aligned}$$

Now we can obtain a lower bound for the numerator

$$\begin{aligned}
\int_{\mathcal{H}_1} \pi(\boldsymbol{h}, i_{\text{opt}}) \exp\{n f_n(\boldsymbol{h}, i_{\text{opt}})\} \mathrm{d}\boldsymbol{h} &\geq \int_{\mathcal{H}_2} \pi(\boldsymbol{h}, i_{\text{opt}}) \exp\{n f_n(\boldsymbol{h}, i_{\text{opt}})\} \mathrm{d}\boldsymbol{h} \\
&\geq \int_{\mathcal{H}_2} \pi(\boldsymbol{h}, i_{\text{opt}}) \exp\{n f(\boldsymbol{h}, i_{\text{opt}}) - k_f n^{1-\alpha}\} \mathrm{d}\boldsymbol{h} \\
&\geq \int_{\mathcal{H}_2} \pi(\boldsymbol{h}, i_{\text{opt}}) \exp\{n f(\boldsymbol{h}_{\text{opt}}, i_{\text{opt}}) \\
&\qquad - n\epsilon_1 - k_f n^{1-\alpha}\} \mathrm{d}\boldsymbol{h}.
\end{aligned}$$

A similar argument gives an upper bound for the difference between the denominator and the numerator

$$\begin{aligned}
&\sum_{i \in \mathfrak{I}} \left(\int_{\mathcal{H}} \pi(\boldsymbol{h}, i) \exp\{n f_n(\boldsymbol{h}, i)\} \mathrm{d}\boldsymbol{h} \right) - \int_{\mathcal{H}_1} \pi(\boldsymbol{h}, i_{\text{opt}}) \exp\{n f_n(\boldsymbol{h}, i_{\text{opt}})\} \mathrm{d}\boldsymbol{h} \\
&\quad \leq \sum_{i \in \mathfrak{I}} \int_{\mathcal{H}} \pi(\boldsymbol{h}, i) \exp\{n f(\boldsymbol{h}_{\text{opt}}, i_{\text{opt}}) - n\epsilon_2 + k_f n^{1-\alpha}\} \mathrm{d}\boldsymbol{h}.
\end{aligned}$$

Thus we have

$$\begin{aligned}
&\frac{\int_{\mathcal{H}_1} \pi(\boldsymbol{h}, i_{\text{opt}}) \exp\{n f_n(\boldsymbol{h}, i_{\text{opt}})\} \mathrm{d}\boldsymbol{h}}{\sum_{i \in \mathfrak{I}} \left(\int_{\mathcal{H}} \pi(\boldsymbol{h}, i) \exp\{n f_n(\boldsymbol{h}, i)\} \mathrm{d}\boldsymbol{h} \right) - \int_{\mathcal{H}_1} \pi(\boldsymbol{h}, i_{\text{opt}}) \exp\{n f_n(\boldsymbol{h}, i_{\text{opt}})\} \mathrm{d}\boldsymbol{h}} \\
&\qquad \geq \exp\{n(\epsilon_2 - \epsilon_1) + 2k_f n^{1-\alpha}\} \int_{\mathcal{H}_2} \pi(\boldsymbol{h}, i_{\text{opt}}) \mathrm{d}\boldsymbol{h}.
\end{aligned}$$

This tends to infinity as $n \to \infty$ since $\epsilon_2 - \epsilon_1 > 0$. Therefore $\mathbb{P}\text{r}[(\boldsymbol{H}, I) \in \mathcal{H}_0 \times \{i_{\text{opt}}\}] \to 1$ as $n \to \infty$. Since the choice of $\mathcal{H}_0 \ni \boldsymbol{h}_{\text{opt}}$ was arbitrary, it may be made infinitesimally small and still, after enough iterations of the sampler $\mathbb{P}\text{r}[(\boldsymbol{H}, I) \in \mathcal{H}_0 \times \{i_{\text{opt}}\}] \to 1$. This implies that $\pi^{(n)}(\boldsymbol{h}, i)$ tends in distribution to a Dirac mass centred on $(\boldsymbol{h}_{\text{opt}}, i_{\text{opt}})$ and establishes the claim. □

Acknowledgments

The authors wish to thank the anonymous reviewers of this article, whose feedback has helped to improve it.