Towards Automatic Model Comparison
An Adaptive Sequential Monte Carlo Approach

Yan Zhou, Adam M. Johansen and John A. D. Aston*

March 13, 2013

Abstract

Model comparison for the purposes of selection, averaging and validation is a problem found throughout statistics and related disciplines. Within the Bayesian paradigm, these problems all require the calculation of the posterior probabilities of models within a particular class. Substantial progress has been made in recent years, but there are numerous difficulties in the practical implementation of existing schemes. This paper presents adaptive sequential Monte Carlo (SMC) sampling strategies to characterise the posterior distribution of a collection of models, as well as the parameters of those models. Both a simple product estimator and a combination of SMC and a path sampling estimator are considered and existing theoretical results are extended to include the path sampling variant. A novel approach to the automatic specification of distributions within SMC algorithms is presented and shown to outperform the state of the art in this area. The performance of the proposed strategies is demonstrated via an extensive simulation study making use of the Gaussian mixture model and two challenging realistic examples. Comparisons with state of the art algorithms show that the proposed algorithms are always competitive, and often substantially superior to alternative techniques, at equal computational cost and considerably less application-specific implementation effort.

Keywords: Adaptive Monte Carlo algorithms; Bayesian model comparison; Normalising constants; Path sampling; Thermodynamic integration

1 Introduction

Model comparison, selection and averaging lie at the core of Bayesian decision theory (Robert, 2007) and have attracted considerable attention in the past few decades. In most cases, approaches to the calculation of the required posterior model probabilities have revolved around simple asymptotic arguments or the post-processing of outputs from Markov chain Monte Carlo (MCMC) algorithms operating on the space of a single model or using specially designed MCMC techniques that provide direct estimates of these quantities (for example Reversible Jump MCMC, RJMCMC; Green (1995)). Within-model simulations are typically somewhat simpler, but generalisations of the harmonic mean estimator (Gelfand and Dey, 1994) which are typically used in this setting require careful design to ensure finite variances and, convergence assessment can be rather difficult. Simulations on the whole model spaces are often difficult to implement efficiently even though they can be conceptually appealing.

More robust and efficient Monte Carlo algorithms have been established in recent years. Many of them are population based, in that they deal explicitly with a collection of samples at each iteration, including sequential importance sampling and resampling (AIS, Neal (2001); SMC, (Del Moral et al., 2006b)) and population MCMC (PMCMC; Liang and Wong (2001); Jasra et al. (2007a)). However, most studies have focused on their abilities to

*AJ is partially supported by; EPSRC grant EP/I017984/1; JA by EPSRC grant EP/H016856/1 and the EPSRC/HEFCE CRiSM grant.

explore high dimensional and multimodal spaces. Results on the effectiveness of these algorithms when applied to Bayesian model comparison problems are less well studied. In the present work, we motivate and present a number of approaches based around the sequential Monte Carlo family of algorithms, and demonstrate the effectiveness of the proposed strategy empirically.

Sequential Monte Carlo (SMC) methods are a class of sampling algorithms which combine importance sampling and resampling. They have been primarily used as "particle filters" to solve optimal filtering problems; see, for example, Cappé et al. (2007); Doucet and Johansen (2011) for recent reviews. They are used here in a different manner, that proposed by Del Moral et al. (2006b) and developed by Del Moral et al. (2006a); Peters (2005). This framework involves the construction of a sequence of artificial distributions on spaces of increasing dimensions which admit the distributions of interest as particular marginals.

Although it is well known that SMC is well suited for the computation of normalising constants and that it is possible to develop relatively automatic SMC algorithms by employing a variety of "adaptive" strategies, their use for Bayesian model comparison has not yet received a great deal of attention. We highlight three strategies for computing posterior model probabilities using SMC, focusing on strategies which require minimal tuning and can be readily implemented requiring only the availability of *locally-mixing* MCMC proposals. These methods admit natural and scalable parallelisation and we demonstrate the potential of these algorithms with real implementations suitable for use on consumer-grade parallel computing hardware including GPUs, reinforcing the message of Lee et al. (2010). We also present a new approach to adaptation and guidelines on the near-automatic implementation of the proposed algorithms. These techniques are applicable to SMC algorithms in much greater generality. The proposed approach is compared with state of the art alternatives in extensive simulation studies which demonstrate its performance and robustness.

In the next section we provide a brief survey of the considerable literature on Monte Carlo methods for Bayesian model comparison. Section 3 presents three algorithms for performing model comparison using SMC techniques and Section 4 provides several illustrative applications, together with comparisons with other techniques. Section 5 extends the standard SMC central limit theorem to include the path sampling estimator. The paper concludes with some discussions in Section 6.

2 Background

Bayesian model comparison must be based upon the posterior distribution over models. It is only possible to obtain closed-form expressions for posterior model probabilities in very limited situations. Over the past five decades, this problem has attracted considerable attention. It is not feasible to exhaustively summarise this literature here. We aim only to describe the major contributions to the area and recent developments which are particularly relevant to the present paper.

2.1 Analytic Methods and MCMC

The Bayesian Information Criterion (BIC), developed by Schwarz (1978), is based upon a large sample approximation of the Bayes factor. It is defined as BIC $= -2\widehat{\ell} + k\log(n)$, where $\widehat{\ell}$ denotes the maximum of the log-likelihood for the observed data, k the number of model parameters and n the effective dimension of the data. An asymptotic argument concerning Bayes factors under appropriate regularity conditions justifies the choice of the model with the smallest value of BIC. Although appealing in its simplicity, such an approach can only be formally justified when a very large number of observations (compared to the number of parameters) is available.

In this era of fast computing, the difficulty of evaluating the integrals that must be computed in order to adapt a fully Bayesian approach is much smaller than it once was. The Bayesian approach to model comparison is, of course, to consider the posterior probabilities of the possible models (Bernardo and Smith, 1994, Chapter 6). Within this Bayesian framework the decision making process, which might include model comparison, model selection

or the choice of an action, depend upon the relative probabilities of several models (Robert, 2007, Chapter 7).

Given an (at most countable) collection of models $\{M_k\}_{k\in\mathcal{K}}$, with model M_k having parameter space Θ_k, Bayesian inference proceeds from a prior distribution over the collection of models, $\pi(M_k)$, a prior distribution for the parameters of each model, $\pi(\theta_k|M_k)$ and the likelihood under each model $p(\boldsymbol{y}|\theta_k, M_k)$. In order to perform model comparison, one requires the posterior model probability,

$$\pi(M_k|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|M_k)\pi(M_k)}{p(\boldsymbol{y})} \tag{1}$$

where $p(\boldsymbol{y}|M_k) = \int_{\theta_k} p(\boldsymbol{y}|\theta_k, M_k)\pi(\theta_k|M_k)\ \mathrm{d}\,\theta_k$ is termed the *evidence* for model M_k and the normalising constant $p(\boldsymbol{y}) = \sum_{k\in\mathcal{K}} p(\boldsymbol{y}|M_k)\pi(M_k)$ can be easily calculated if $|\mathcal{K}|$ is finite and the evidence for each model is available. The case where $|\mathcal{K}|$ is countable is discussed later. We first review some techniques for calculating the evidence for each model individually.

Several techniques have been proposed to approximate the evidence for a model using simulation techniques which approximate the posterior distribution of that model, including the harmonic mean estimator of Newton and Raftery (1994); Raftery et al. (2006) and the generalisations due to Gelfand and Dey (1994). These pseudo-harmonic mean methods are based around the insight that for any density g, such that g and the posterior density are mutually absolutely continuous, the fact that following identity holds,

$$\int \frac{g(\theta_k)}{p(\boldsymbol{y}, \theta_k|M_k)}\pi(\theta_k|\boldsymbol{y}, M_k)\ \mathrm{d}\,\theta_k = \int \frac{g(\theta_k)}{p(\boldsymbol{y}, \theta_k|M_k)}\frac{p(\boldsymbol{y}, \theta_k|M_k)}{p(\boldsymbol{y}|M_k)}\ \mathrm{d}\,\theta_k = \frac{1}{p(\boldsymbol{y}|M_k)} \tag{2}$$

and by approximating the leftmost integral using any numerical integration technique one can in principle obtain an estimate of the evidence. Unfortunately, considerable care is required in the implementation of such schemes in order to control the variance of the resulting estimator (and indeed, to ensure that this variance is finite; see for example Neal (1994)).

In the particular case of the Gibbs sampler, Chib (1995) provides an alternative approach to the approximation of the evidence from within-model simulations based on that the identity,

$$p(\boldsymbol{y}|M_k) = \frac{p(\boldsymbol{y}|\theta_k, M_k)\pi(\theta_k|M_k)}{\pi(\theta_k|\boldsymbol{y}, M_k)}, \tag{3}$$

holds for any value of θ_k. Therefore, an estimator of the marginal likelihood can be obtained by replacing θ_k with a particular value, say $\theta_k^\star$, which is usually chosen from the high probability region of the posterior distribution and approximating the denominator $\pi(\theta_k^\star|\boldsymbol{y}, M_k)$ using the output from a Gibbs sampler. Though this method does not suffer the instability associated with generalised harmonic mean estimators, it requires that all full conditional densities are known including their normalising constants. This approach was generalised to other Metropolis-Hastings algorithms by Chib and Jeliazkov (2001), where only the proposal distributions are required to be known including their normalising constants.

The first MCMC method which operated simultaneously over the full collection of models of interest providing direct estimates of posterior model probabilities was probably the approach of Grenander and Miller (1994). However, the general Reversible Jump MCMC (RJMCMC) strategy first proposed by Green (1995) is undoubtedly the most widespread of these techniques. RJMCMC adapts the Metropolis-Hastings algorithm to construct a Markov chain on an extended state-space which admits the posterior distribution over both model and parameters as its invariant distribution. The algorithm operates on the space $\bigcup_{k\in\mathcal{K}}(\{M_k\}\times\Theta_k)$. A countable set of types of moves are considered, say $m \in \mathcal{M}$, and each move type m is capable of moving between two models, say M_k and $M_{k'}$ (where $k = k'$ in the case of within-model moves). At state θ_k, a move type m together with a new state $\theta_{k'}$ are proposed according to $q_m(\theta_k, \theta_{k'})r_m(\theta_k)$, where $r_m(\theta_k)$ is the probability of choosing type m move when at state θ_k and $q_m(\theta_k, \theta_{k'})$ is the proposal kernel for the new state when move type m is made. The move is accepted with probability

$$\alpha_m(\theta_k, \theta_{k'}) = \min\left\{1, \frac{\pi(M_{k'})\pi(\theta_{k'}|M_{k'})p(\boldsymbol{y}|\theta_{k'}, M_{k'})}{\pi(M_k)\pi(\theta_k|M_k)p(\boldsymbol{y}|\theta_k, M_k)}\frac{q_m(\theta_{k'}, \theta_k)r_m(\theta_k')}{q_m(\theta_k, \theta_{k'})r_m(\theta_k)}\right\}. \tag{4}$$

In practice, sampling of the proposed new state $\theta_{k'}$ is often achieved by drawing a vector of continuous random variables, say u, which are independent of θ_k and applying a deterministic mapping of vector (θ_k, u) to $\theta_{k'}$. The inverse of the move, from $\theta_{k'}$ back to θ_k, then uses the inverse of this transformation. Through a simple change of variable, the conditional density $q_m(\theta_k, \theta_{k'})$ can be expressed in terms of the density of vector u, $q(u)$, and the acceptance probability becomes

$$\alpha_m(\theta_k, \theta_{k'}) = \min\left\{1, \frac{\pi(M_{k'})\pi(\theta_{k'}|M_{k'})p(\boldsymbol{y}|\theta_{k'}, M_{k'})}{\pi(M_k)\pi(\theta_k|M_k)p(\boldsymbol{y}|\theta_k, M_k)} \frac{r_m(\theta_{k'})}{r_m(\theta_k)} \frac{1}{q(u)} \left|\frac{\partial \theta_{k'}}{\partial(\theta_k, u)}\right|\right\}, \quad (5)$$

where the last term is the determinant of the Jacobian of the transformation. The design of efficient between-model moves is often difficult, and the mixing of these moves largely determines the performance of the algorithm. For example, in multimodal models, where RJMCMC has attracted substantial attention, information available in the posterior distribution of a model of any given dimension does not characterise modes that exist only in models of higher dimension, and thus successful moves between those models become unlikely and difficult to construct (Jasra et al., 2007b). In addition, RJMCMC will not characterise models of low posterior probability well, as those models will be visited by the chain only rarely. In some cases it will be difficult to determine whether the low acceptance rates of between-model moves result from actual characteristics of the posterior or from a poorly-adapted proposal kernel.

The related continuous time birth and death algorithm of Stephens (2000) was shown by Cappé et al. (2003) to have no qualitative advantage over the simpler discrete time formulation. A post-processing approach to improve the computation of normalising constants from RJMCMC output using a bridge-sampling approach was advocated by Bartolucci et al. (2006). Sophisticated variants of these algorithms, such as those developed in Peters et al. (2010), have been considered in recent years but depend upon essentially the same construction and ultimately require adequate mixing of the underlying Markov process.

Carlin and Chib (1995) presented an alternative method for simulating the model probability directly through a Gibbs sampler on the space $\{M_k\}_{k\in\mathcal{K}} \times \prod_{k\in\mathcal{K}} \Theta_k$. The joint parameter is thus (M, θ) where θ is the vector $(\theta_k)_{k\in\mathcal{K}}$ and conditional on model M_k the data $\boldsymbol{y}$ only depends on a subset, θ_k, of the parameters. To form the Gibbs sampler, a so called pseudoprior $\pi(\theta_k|M \neq M_k)$ in addition to the usual prior $\pi(\theta_k|M_k)$ is selected, such that given the model indicator M, the parameters associated with different models are conditionally mutually independent. In this way, a Gibbs sampler can be constructed provided that all the full conditional distributions $\pi(\theta_k|\boldsymbol{y}, \theta_{k'\neq k}, M)$ and $\pi(M = M_k|\boldsymbol{y}, \theta)$ for $k \in \mathcal{K}$ are available. The major drawback of this approach is that the performance and validity of the sampler is very sensitive to the selected pseudopriors, and as usual for all Gibbs samplers, the full conditional distribution needs to be readily sampled from. This approach was later generalised by Godsill (2001) who also explored the connection with RJMCMC.

Overall, the methods reviewed above either demand some knowledge of the target distributions that is often missing in reality, or require substantial tuning in order for the algorithms to perform well.

2.2 Recent Developments on Population-Based Methods

In the recent computational statistics literature, there has been a tendency to consider the use of population-based sampling methods. There are two popular approaches among many others. One is based on sequential importance sampling and resampling, such as the SMC sampler (Del Moral et al., 2006b) and earlier development of annealed importance sampling (AIS; Neal (2001)) which can be viewed as a special case of SMC. Another approach is population MCMC (PMCMC; Marinari and Parisi (1992); Geyer (1991); Liang and Wong (2001)) also known as parallel tempering, which uses a collection of parallel MCMC chains to approximate a target distribution. PMCMC operates by constructing a sequence of distributions $\{\pi_t\}_{t=0}^T$ with π_0 corresponding to the target distribution and successive elements of this sequence consisting of distributions from which it is increasingly easy to sample. A population of samples is maintained, with the i^{th} element of the population being approximately distributed according to π_i; the algorithm proceeds by simulating an ensemble of parallel MCMC chains each targeting one of these distributions. The chains interact with

one another via exchange moves, in which the state of two adjacent chains is swapped, and this mechanism allows for information to be propagated between the chains and hopefully for the fast mixing of π_T to be partially transferred to the chain associated with π_0. The outputs are samples that approximate the product $\prod_{t=0}^{T} \pi_t$ which admits the target distribution as its first coordinate marginal.

There is evidence in the literature of substantial interest in the potential of using population based methods to explore high dimensional and multimodal parameter spaces which were previously difficult for conventional MCMC algorithms. Jasra et al. (2007a) compared the performance of the two approaches in this context. There are also increasing interest of using these methods for Bayesian model comparison. The PMCMC outputs can be post-processed in the same way as conventional MCMC to obtain estimates of evidence for each model (for example using a generalised harmonic mean estimator). However, this approach inherits many of the disadvantages of this estimator. Jasra et al. (2007b) combined PMCMC with RJMCMC and thus provide a direct estimate of the posterior model probability. Another approach is to use the outputs from all the chains to approximate the path sampling estimator (Gelman and Meng, 1998), see Calderhead and Girolami (2009). However, the mixing speed of PMCMC is sensitive to the number and placement of the distributions $\{\pi_t\}_{t=0}^{T}$ (see Atchadé et al. (2010) for the optimal placement of distributions in terms of a particular mixing criterion for a restricted class of models). As seen in Calderhead and Girolami (2009), the placement of distributions can play a crucial role in the performance of the estimator, a topic we will revisit in Section 4.

The use of AIS for computing normalising constants directly and via path sampling dates back at least to Neal (2001); see Vyshemirsky and Girolami (2008) for a recent example of its use in the computation of model evidences. In the literature it has generally been suggested that more general SMC strategies provide no advantage over AIS when the normalizing constant is the object of inference. Later we will demonstrate that this is not generally true, adding improved robustness of normalizing constant estimates to the advantages afforded by resampling within SMC. We will also discuss more details on the use of SMC and path sampling for Bayesian model section in the next section. The use of PMCMC coupled with path sampling was discussed in Vyshemirsky and Girolami (2008).

Jasra et al. (2008) developed a method using a system of interacting SMC samplers for trans-dimensional simulation. The targeting distribution π and its space S are the same as in RJMCMC. As usual in SMC, a sequence of distributions $\{\widetilde{\pi}_t\}_{t=0}^{T}$ with increasing dimensions are constructed such that $\widetilde{\pi}_T$ admits π as a marginal. The algorithm starts with a set of SMC samplers with equal number of particles; each of them targets $\widetilde{\pi}_{i,t}(x) \propto \widetilde{\pi}_t(x)\mathbb{I}(x \in S_{i,t})$ up to a predefined time index $t^\star$, such that $\{S_{i,0}\}$ is a partition of S and $S_{i,t^\star} = S$. At time $t^\star$ particles from all samplers are allowed to coalesce, and from this time on, all of them are iterated with the same Markov kernel until the single sampler reaches the target π. Each individual sampler explores only a portion of the parameter space and by using the information which each sampler gains about that region of the parameter space, with a properly chosen $t^\star$, the resulting sampler will be able to explore the whole space efficiently. One of the three algorithms detailed in the next section coincides, essentially, with the final stage of the approach of Jasra et al. (2008); the other algorithms which are developed rely on a quite different strategy.

A proof of concept study in which several SMC approaches to the problem were outlined was provided by Zhou et al. (2012) and these approaches are developed below. These strategies based around various combinations of path sampling (Gelman and Meng, 1998) and SMC (as used by Johansen et al. (2006) in a rare events context and by Rousset and Stoltz (2006) in the context of the estimation of free energy differences) or the unbiased estimation of the normalizing constant via standard SMC techniques (Del Moral, 1996; Del Moral et al., 2006b).

A number of other recent developments should be mentioned for completeness, but are not directly relevant to the problems considered here. A strategy for SMC-based variable selection was developed by Schäfer and Chopin (2013); this approach depends upon the precise structure of this particular problem and does not involve the explicit computation of normalizing constants.

In recent years, several approaches to the problem of Bayesian model comparison in settings in which the likelihood cannot be evaluated have also been proposed: Grelaud

et al. (2009); Didelot et al. (2011); Robert et al. (2011). This class of problems falls outside the scope of the current paper. We will assume throughout that the likelihood of all models can be evaluated point wise.

2.3 Challenges for Model Comparison Techniques

We conclude this background section by noting that there are a number of desirable features in algorithms which seek to address any model comparison problem and that these desiderata can find themselves in competition with one another. One always requires accurate evaluation of Bayes factors or model proportions and to obtain these one requires estimates of either normalizing constants or posterior model probabilities with small error making the efficiency of any Monte Carlo algorithm employed in their estimation critical. If one is interested in characterising behaviour conditional upon a given model or even calculating posterior-predictive quantities, it is likely to be necessary to explore the full parameter space of each model; this can be difficult if one employs between-model strategies which spend little time in models of low probability. In many settings end-users seek to interpret the findings of model selection experiments and in such cases, accurate characterisation of all models including those of relatively small probability may be important.

3 Methodology

SMC samplers allow us to obtain, iteratively, collections of weighted samples from a sequence of distributions $\{\pi_t\}_{t=0}^T$ over essentially any random variables on some measurable spaces $(E_t, \mathcal{E}_t)$, by constructing a sequence of auxiliary distributions $\{\widetilde{\pi}_t\}_{t=0}^T$ on spaces of increasing dimensions,

$$\widetilde{\pi}_t(x_{0:t}) = \pi_t(x_t) \prod_{s=0}^{t-1} L_s(x_{s+1}, x_s), \tag{6}$$

where the sequence of Markov kernels $\{L_s\}_{s=0}^{t-1}$, termed backward kernels, is formally arbitrary but critically influences the estimator variance. See Del Moral et al. (2006b) for further details and guidance on the selection of these kernels.

Standard sequential importance resampling algorithms can then be applied to the sequence of synthetic distributions, $\{\widetilde{\pi}_t\}_{t=0}^T$. At time $t = n-1$, assume that a set of weighted particles $\{W_{n-1}^{(i)}, X_{0:n-1}^{(i)}\}_{i=1}^N$ approximating $\widetilde{\pi}_{n-1}$ is available, then at time $t = n$, the path of each particle is extended with a Markov kernel say, $K_n(x_{n-1}, x_n)$ and the set of particles $\{X_{0:n}^{(i)}\}_{i=1}^N$ reach the distribution $\eta_n(x_{0:n}) = \eta_0(x_0) \prod_{t=1}^n K_t(x_{t-1}, x_t)$, where η_0 is the initial distribution of the particles. To correct the discrepancy between η_n and $\widetilde{\pi}_n$, importance sampling is then applied, that is the weights are corrected by

$$W_n(x_{0:n}) \propto \frac{\widetilde{\pi}_n(x_{0:n})}{\eta_n(x_{0:n})} = \frac{\pi_n(x_n) \prod_{s=0}^{n-1} L_s(x_{s+1}, x_s)}{\eta_0(x_0) \prod_{t=1}^n K_t(x_{t-1}, x_t)} \propto W_{n-1}(x_{0:n-1}) \widetilde{w}_n(x_{n-1}, x_n) \tag{7}$$

where $\widetilde{w}_n$, termed the *incremental weights*, are calculated as,

$$\widetilde{w}_n(x_{n-1}, x_n) = \frac{\pi_n(x_n) L_{n-1}(x_n, x_{n-1})}{\pi_{n-1}(x_{n-1}) K_n(x_{n-1}, x_n)} \tag{8}$$

If π_n is only known up to a normalizing constant, say $\pi_n(x_n) = \gamma_n(x_n)/Z_n$, then we can use the *unnormalised* incremental weights

$$w_n(x_{n-1}, x_n) = \frac{\gamma_n(x_n) L_{n-1}(x_n, x_{n-1})}{\gamma_{n-1}(x_{n-1}) K_n(x_{n-1}, x_n)} \tag{9}$$

for importance sampling. Further, with the previously *normalised* weights $\{W_{n-1}^{(i)}\}_{i=1}^N$, we can estimate the ratio of normalizing constant Z_n/Z_{n-1} by

$$\frac{\widehat{Z}_n}{Z_{n-1}} = \sum_{i=1}^N W_{n-1}^{(i)} w_n(X_{n-1:n}^{(i)}), \tag{10}$$

	PHM	RJMCMC	PMCMC	**SMC1**	**SMC2**	**SMC3**
Can deal with a countable set of models		✓		✓		
Can exploit inter-model relationships		✓		✓		✓
Characterises improbable models	✓		✓		✓	✓
Doesn't require reversible-pairs of moves	✓		✓	✓	✓	✓
Doesn't require inter-model mixing	✓		✓		✓	
Admits straightforward parallelisation			✓/×	✓	✓	✓
Doesn't rely upon ergodicity arguments				✓	✓	✓

Table 1: Strengths of computational strategies for model choice. PMCMC admits parallelisation up to the number of chains used, but is not a natural candidate for implementation on massively-parallel architectures.

and

$$\frac{\widehat{Z}_n}{Z_1} = \prod_{p=2}^{n} \frac{\widehat{Z}_p}{Z_{p-1}} = \prod_{p=2}^{n} \sum_{i=1}^{N} W_{p-1}^{(i)} w_p(X_{p-1:p}^{(i)}), \tag{11}$$

provides an unbiased (Del Moral, 2004, Proposition 7.4.1) estimate of Z_n/Z_1. Sequentially, the normalizing constant between initial distribution π_0 and target π_T can be estimated. See Del Moral et al. (2006b) for details on calculating the incremental weights in general; in practice, when K_n is π_n-invariant, $\pi_n \ll \pi_{n-1}$, and

$$L_{n-1}(x_n, x_{n-1}) = \frac{\pi_n(x_{n-1})K_n(x_{n-1}, x_n)}{\pi_n(x_n)} \tag{12}$$

is used as the backward kernel, the unnormalised incremental weights become

$$w_n(x_{n-1}, x_n) = \frac{\gamma_n(x_{n-1})}{\gamma_{n-1}(x_{n-1})}. \tag{13}$$

This will be the situation throughout the remainder of this paper.

3.1 Sequential Monte Carlo for Model Comparison

The problem of interest is characterising the posterior distribution over $\{M_k\}_{k\in\mathcal{K}}$, a set of possible models, with model M_k having parameter vector $\theta_k \in \Theta_k$ which must also usually be inferred. Given prior distributions $\pi(M_k)$ and $\pi(\theta_k|M_k)$ and likelihood $p(\boldsymbol{y}|\theta_k, M_k)$ we seek the posterior distributions $\pi(M_k|\boldsymbol{y}) \propto p(\boldsymbol{y}|M_k)$. There are three fundamentally different approaches to the computations:

1. Calculate posterior model probabilities directly.
2. Calculate the evidence, $p(\boldsymbol{y}|M_k)$, of each model.
3. Calculate pairwise evidence ratios.

Each approach admits a natural SMC strategy. The relative strengths of these approaches, which are introduced in the following sections, and alternative methods are identified in Table 1.

3.1.1 SMC1: An All-in-One Approach

One could consider obtaining samples from the same distribution employed in the RJMCMC approach to model comparison, namely:

$$\pi^{(1)}(M_k, \theta_k) \propto \pi(M_k)\pi(\theta_k|M_k)p(\boldsymbol{y}|\theta_k, M_k) \tag{14}$$

which is defined on the disjoint union space $\bigcup_{k\in\mathcal{K}}(\{M_k\}\times\Theta_k)$.

One obvious SMC approach is to define a sequence of distributions $\{\pi_t^{(1)}\}_{t=0}^T$ such that $\pi_0^{(1)}$ is easy to sample from, $\pi_T^{(1)}=\pi^{(1)}$ and the intermediate distributions move smoothly between them. In the remainder of this section, we use the notation (M_t,θ_t) to denote a random sample on the space $\bigcup_{k\in\mathcal{K}}(\{M_k\}\times\Theta_k)$ at time t. One simple approach, which might be expected to work well, is the use of an annealing scheme such that:

$$\pi_t^{(1)}(M_t,\theta_t)\propto\pi(M_t)\pi(\theta_t|M_t)p(\boldsymbol{y}|\theta_t,M_t)^{\alpha(t/T)}, \tag{15}$$

for some monotonically increasing $\alpha:[0,1]\rightarrow[0,1]$ such that $\alpha(0)=0$ and $\alpha(1)=1$. Other approaches are possible and might prove more efficient for some problems (such as the "data tempering" approach which Chopin (2002) proposed for parameter estimation which could easily be incorporated in our framework), but this strategy provides a convenient generic approach. These choices lead to Algorithm 1.

This approach might outperform RJMCMC when it is difficult to design fast-mixing Markov kernels. There are many examples of such an annealed SMC strategy outperforming MCMC at a given computational cost — see, for example, Fan et al. (2008); Johansen et al. (2008); Fearnhead and Taylor (2010). Such trans-dimensional SMC has been proposed in several contexts (Peters, 2005) and an extension proposed and analysed by Jasra et al. (2008).

Algorithm 1 SMC1: An All-in-One Approach to Model Comparison.

Initialisation: Set $t\leftarrow 0$.
Sample $X_0^{(i)}=(M_0^{(i)},\theta_0^{(i)})\sim\nu$ for some proposal distribution ν (usually the joint prior).
Weight $W_0^{(i)}\propto w_0(X_0^{(i)})=\pi(M_0^{(i)})\pi(\theta_0^{(i)}|M_0^{(i)})/\nu(M_0^{(i)},\theta_0^{(i)})$.
Apply resampling if necessary (e.g., if ESS (Kong et al., 1994) less than some threshold).
Iteration: Set $t\leftarrow t+1$.
Weight $W_t^{(i)}\propto W_{t-1}^{(i)}p(\boldsymbol{y}|\theta_{t-1}^{(i)},M_{t-1}^{(i)})^{\alpha(t/T)-\alpha([t-1]/T)}$.
Apply resampling if necessary.
Sample $X_t^{(i)}\sim K_t(\cdot|X_{t-1}^{(i)})$, a $\pi_t^{(1)}$-invariant kernel.
Repeat the *Iteration* step *until* $t=T$.

We include this approach for completeness and study it empirically later. However, the more direct approaches described in the following sections lead more naturally to easy-to-implement strategies with good performance.

3.1.2 SMC2: A Direct-Evidence-Calculation Approach

An alternative approach would be to estimate explicitly the evidence associated with each model. We propose to do this by sampling from a sequence of distributions for each model: starting from the parameter prior and sweeping through a sequence of distributions to the posterior.

Numerous strategies are possible to construct such a sequence of distributions, but one option is to use for each model M_k, $k\in\mathcal{K}$, the sequence $\{\pi_t^{(2,k)}\}_{t=0}^{T_k}$, defined by

$$\pi_t^{(2,k)}(\theta_t)\propto\pi(\theta_t|M_k)p(\boldsymbol{y}|\theta_t,M_k)^{\alpha_k(t/T_k)}. \tag{16}$$

where the number of distribution T_k, and the annealing schedule, $\alpha_k:[0,1]\rightarrow[0,1]$ may be different for each model. This leads to Algorithm 2.

The estimator of the posterior model probabilities depends upon the approach taken to estimate the normalizing constant. Direct estimation of the evidence can be performed using the output of this SMC algorithm and the standard unbiased estimator, termed SMC2-DS below:

$$\sum_{i=1}^N\frac{\pi(\theta_0^{(k,i)}|M_k)}{\nu(\theta_0^{(k,i)})}\times\prod_{t=2}^T\sum_{i=1}^N W_{t-1}^{(k,i)}p(\boldsymbol{y}|\theta_{t-1}^{(k,i)}M_k)^{\alpha_k(t/T_k)-\alpha_k([t-1]/T_k)} \tag{17}$$

where $W_{t-1}^{(k,i)}$ is the importance weight of sample i, $\theta_{t-1}^{(k,i)}$, during iteration $t-1$ for model M_k. An alternative approach to computing the evidence is also worthy of consideration. As has been suggested, and shown to perform well empirically previously (Johansen et al., 2006, see, for example), it is possible to use all of the samples from every generation of an SMC sampler to approximate the path sampling estimator and hence to obtain an estimate of the ratio of normalizing constants. Section 3.2 provides details.

The posterior distribution of the parameters conditional upon a particular model can also be approximated with:

$$\widehat{\pi}_{T_k}^{(2,k)}(\mathrm{d}\,\theta) = \sum_{i=1}^{N} W_{T_k}^{(k,i)} \delta_{\theta_{T_k}^{(k,i)}}(\mathrm{d}\,\theta).$$

where $\delta_{\theta_{T_k}^{(k,i)}}$ is the Dirac measure.

This approach is appealing for several reasons. One is that it is designed to estimate directly the quantity of interest: the evidence, producing a sample from that distribution at the same time. Another advantage of this approach over SMC1 and the RJMCMC approach is that it provides as good a characterisation of each model as is required: it is possible to obtain a good estimate of the parameters of every model, even those for which the posterior probability is small. Perhaps most significant is the fact that this approach does not require the design of proposal distributions or Markov kernels which move from one model to another: each model is dealt with in isolation. Whilst this may not be desirable in every situation, there are circumstances in which efficient moves between models are almost impossible to devise.

This approach also has some disadvantages. In particular, it is necessary to run a separate simulation for each model — rendering it impossible to deal with countable collections of models (although this is not such a substantial problem in many interesting cases). The ease of implementation may often offset this limitation.

Algorithm 2 SMC2: A Direct-Evidence-Calculation Approach.

For each model $k \in \mathcal{K}$ perform the following algorithm.

Initialisation: Set $t \leftarrow 0$.

Sample $\theta_0^{(k,i)} \sim \nu_k$ for some proposal distribution ν_k (usually the parameter prior).

Weight $W_0^{(k,i)} \propto w_0(\theta_0^{(k,i)}) = \pi(\theta_0^{(k,i)}|M_k)/\nu_k(\theta_0^{(k,i)})$.

Apply resampling if necessary.

Iteration: Set $t \leftarrow t+1$.

Weight $W_t^{(k,i)} \propto W_{t-1}^{(k,i)} p(\boldsymbol{y}|\theta_{t-1}^{(k,i)}, M_k)^{\alpha(t/T_k)-\alpha([t-1]/T_k)}$.

Apply resampling if necessary.

Sample $\theta_t^{(k,i)} \sim K_t(\cdot|\theta_{t-1}^{(k,i)})$, a $\pi_t^{(k,2)}$-invariant kernel.

Repeat the *Iteration* step *until* $t = T_k$.

3.1.3 SMC3: A Relative-Evidence-Calculation Approach

A final approach can be thought of as *sequential model comparison*. Rather than estimating the evidence associated with any particular model, we could estimate pairwise evidence ratios directly. The SMC sampler starts with a initial distribution being the posterior of one model (which could comes from a separate SMC sampler starting from its prior) and moves towards the posterior of another related model. Then the sampler can continue towards another related model.

Given a finite collection of models $\{M_k\}$, $k \in \mathcal{K}$, suppose the models are ordered in a sensible way (e.g., M_{k-1} is nested within M_k or θ_k is of higher dimension than θ_{k-1}). For each $k \in \mathcal{K}$, we consider a sequence of distributions $\{\pi_t^{(3,k)}\}_{t=0}^{T_k}$, such that $\pi_0^{(3,k)}(M,\theta) = \pi(\theta|\boldsymbol{y}, M_k)\mathbb{I}_{\{M_k\}}(M)$ and $\pi_{T_k}^{(3,k)}(M,\theta) = \pi(\theta|\boldsymbol{y}, M_{k+1})\mathbb{I}_{\{M_{k+1}\}}(M) = \pi_0^{(3,k+1)}(M,\theta)$. When it is possible to construct a SMC sampler that iterates over this sequence of distributions, the estimate of the ratio of normalizing constants is the Bayes factor estimate of model M_{k+1} in favour of model M_k.

This approach is conceptually appealing, but requires the construction of a smooth path between the posterior distributions of interest. The geometric annealing strategy which has been advocated as a good generic strategy in the previous sections is only appropriate when the support of successive distributions is non-increasing. This is unlikely to be the case in interesting model comparison problems.

In this paper we consider a sequence of distributions on the disjoint union $\{M_k, \Theta_k\} \cup \{M_{k+1}, \Theta_{k+1}\}$, with the sequence of distributions $\{\pi_t^{(3,k)}\}_{t=0}^{T_k}$ defined as the full posterior,

$$\pi_t^{(3,k)}(M_t, \theta_t) \propto \pi_t(M_t)\pi(\theta_t|M_t)p(\boldsymbol{y}|\theta_t, M_t) \tag{18}$$

where $M_t \in \{M_k, M_{k+1}\}$ and the prior of models at time t, $\pi_t(M_t)$ is defined by

$$\pi_t(M_{k+1}) = \alpha(t/T_k) \tag{19}$$

for some monotonically increasing $\alpha : [0,1] \to [0,1]$ such that $\alpha(0) = 0$ and $\alpha(1) = 1$. It is clear that the MCMC moves between iterations need to be similar to those in the RJMCMC or SMC1 algorithms. The difference is that instead of efficient exploration of the whole model space, only moves between two models are required and the sequence of distributions employed helps to ensure exploration of both model spaces. The algorithm for this particular sequence of distribution is outlined in Algorithm 3. It can be extended to other possible sequence of distributions between models.

An advantage of this approach is that it provides direct estimate of the Bayes factor which is of interest for model comparison purpose while not requiring exploration of as complicated a space as that employed within RJMCMC or SMC1. The estimating of normalizing constant in SMC3 can follows in exactly the same manner as in the SMC2 case. In SMC3, the same estimator provides a direct estimate of the Bayes factor.

Algorithm 3 SMC3: A Relative-Evidence-Calculation Approach to Model Comparison.

Initialisation: Set $k \leftarrow 1$.

Use Algorithm 2 to obtain weighted samples for $\pi_{T_1}^{(3,1)}$, the parameter posterior for model M_1

Relative Evidence Calculation

Set $k \leftarrow k+1$, $t \leftarrow 0$.

Denote current weighted samples as $\{W_0^{(k,i)}, X_0^{(k,i)}\}_{i=1}^N$ where $X_0^{(k,i)} = (M_0^{(k,i)}, \theta_0^{(k,i)})$

Apply resampling if necessary.

Iteration: Set $t \leftarrow t+1$.

Weight $W_t^{(k,i)} \propto W_{t-1}^{(k,i)} \pi_t(M_{t-1}^{(k,i)}) / \pi_{t-1}(M_{t-1}^{(k,i)})$.

Apply resampling if necessary.

Sample $(M_t^{(k,i)}, \theta_t^{(ki)}) \sim K_t(\cdot|M_{t-1}^{(k,i)}\theta_{t-1}^{(k,i)})$, a $\pi_t^{(3,k)}$-invariant kernel.

Repeat the Iteration *step up to* $t = T_k$.

Repeat the *Relative Evidence Calculation* step *until sequentially all relative evidences are calculated.*

3.2 Path Sampling via SMC2/SMC3

The estimation of the normalizing constant associated with our sequences of distributions can be achieved by a Monte Carlo approximation to the *path sampling* formulation given by Gelman and Meng (1998) (sometimes known as thermodynamic integration or Ogata's method). This approach is very closely related to the use of AIS for the same purpose Neal (2001) but as will be demonstrated below the incorporation of some other elements of the more general SMC algorithm family can improve performance at negligible cost. Given a parameter α which defines a family of distributions, $\{p_\alpha = q_\alpha/Z_\alpha\}_{\alpha\in[0,1]}$ which move smoothly from $p_0 = q_0/Z_0$ to $p_1 = q_1/Z_1$ as α increases from zero to one, one can estimate the logarithm of the ratio of their normalizing constants via a simple integral relationship

which holds under very mild regularity conditions:

$$\log\left(\frac{Z_1}{Z_0}\right) = \int_0^1 \mathbb{E}_\alpha\left[\frac{\mathrm{d}\log q_\alpha(\cdot)}{\mathrm{d}\,\alpha}\right]\mathrm{d}\,\alpha, \tag{20}$$

where $\mathbb{E}_\alpha$ denotes expectation under p_α; see Gelman and Meng (1998). Note that the sequence of distributions in the SMC2 and SMC3 algorithms above, can both be interpreted as belonging to such a family of distributions, with $\alpha_t = \alpha(t/T_k)$, where the mapping $\alpha : [0,1] \to [0,1]$ is again monotonic with $\alpha(0) = 0$ and $\alpha(1) = 1$.

The SMC sampler provides us with a set of weighted samples obtained from a sequence of distributions suitable for approximating this integral. At each t we can obtain an estimate of the expectation within the integral for $\alpha(t/T)$ via the usual importance sampling estimator, and this integral can then be approximated via numerical integration. Whenever the sequence of distributions employed by SMC3 has appropriate differentiability it is also possible to employ path sampling to estimate, directly, the evidence ratio via this approach applied to the samples generated by that algorithm. In general, given an increasing sequence $\{\alpha_t\}_{t=0}^T$ where $\alpha_0 = 0$ and $\alpha_T = 1$, a family of distributions $\{p_\alpha\}_{\alpha\in[0,1]}$ as before, and a SMC sampler that iterates over the sequence of distribution $\{\pi_t = p_{\alpha_t} = q_{\alpha_t}/Z_{\alpha_t}\}_{t=0}^T$, then with the weighted samples $\{W_t^{(j)}, X_t^{(j)}\}_{j=1}^N$, and $t = 0, \ldots, T$, a path sampling estimator of the ratio of normalizing constants $\Xi_T = \log(Z_1/Z_0)$ can be approximated (using an elementary trapezoidal scheme) by

$$\widehat{\Xi}_T^N = \sum_{t=1}^T \frac{1}{2}(\alpha_t - \alpha_{t-1})(U_t^N + U_{t-1}^N) \tag{21}$$

where

$$U_t^N = \sum_{j=1}^N W_t^{(j)} \frac{\mathrm{d}\log q_\alpha(X_t^{(j)})}{\mathrm{d}\,\alpha}\Big|_{\alpha=\alpha_t} \tag{22}$$

We term these estimators SMC2-PS and SMC3-PS in the followings. The combination of SMC and path sampling is somewhat natural and has been proposed before, e.g., Johansen et al. (2006) although not there in a Bayesian context. Despite the good performance observed in the setting of rare event simulation, the estimation of normalizing constants by this approach seems to have received little attention in the literature. We suspect that this is because of widespread acceptance of the suggestion of Del Moral et al. (2006b), that SMC doesn't outperform AIS when normalizing constants are the object of inference or that of Calderhead and Girolami (2009) that all simulation-based estimators based around path sampling can be expected to behave similarly. We will demonstrate below that these observations, whilst true in certain contexts, do not hold in full generality.

3.3 Extensions and Refinements

3.3.1 Improved Univariate Numerical Integration

As seen in the last section, the path sampling estimator requires evaluation of the expectation,

$$\mathbb{E}_\alpha\left[\frac{\mathrm{d}\log q_\alpha(\cdot)}{\mathrm{d}\,\alpha}\right]$$

for $\alpha \in [0,1]$, which can be approximated by importance sampling using samples generated by a SMC sampler operating on the sequence of distributions $\{\pi_t = p_{\alpha_t} = q_{\alpha_t}/Z_t\}_{t=0}^T$ directly for $\alpha \in \{\alpha_t\}_{t=0}^T$. For arbitrary $\alpha \in [0,1]$, finding t such that $\alpha \in (\alpha_{t-1}, \alpha_t)$, the expectation can be easily approximated using existing SMC samples — the quantities required in the importance weights to obtain such an estimate have already been calculated during the running of the SMC algorithm and such computations have little computational cost.

As noted by Friel et al. (2012) we can use more sophisticated numerical integration strategies to reduce the path sampling estimator bias. For example, higher order Newton-Cotes rules rather than the Trapezoidal rule can be implemented straightforwardly. In the case of SMC it is especially straightforward to estimate the required expectations at arbitrary

α and so we cheaply use higher order integration schemes and we can also use numerical integrations which make use of a finer mesh $\{\alpha'_t\}_{t=0}^{T'}$ than $\{\alpha_t\}_{t=0}^{T}$. Since higher order numerical integrations based on approximations of derivatives obtained from Monte Carlo methods may potentially be unstable in some situations, the second approach can be more appealing in some applications. A demonstration of the bias reduction effect is provided in Section 4.3.

3.3.2 Adaptive Specification of Distributions

In settings in which the importance weights at time t depend only upon the sample at time $t-1$, such as that considered here, it is relatively straightforward to consider sample-dependent, adapative specification of the sequence of distributions (typically by choosing the value of a tempering parameter, such as α_t based upon the current sample). Jasra et al. (2010) proposed such a method of adaptive placing the distributions in SMC algorithms based on controlling the rate at which the effective sample size (ESS; Kong et al. (1994)) falls. With very little computation cost, this provides an automatic method of specifying a tempering schedule in such a way that the ESS decays in a regular fashion. Schäfer and Chopin (2013, Algorithm 2) used a similar technique but by moving the particle system only when it resamples they are in a setting which would be equivalent to resampling at every timestep (with longer time steps, followed by multiple applications of the MCMC kernel) in our formulation. We advocate resampling only adaptively when ESS is smaller than certain preset threshold, and here we propose a more general adaptive scheme for the selection of the sequence of distributions which has significantly better properties when adaptive resampling is employed.

The ESS was designed to assess the loss of efficiency arising from the use a simple weighted sample (rather than a simple random sample from the distribution of interest) in the computation of expectations. It's obtained by considering a sample approximation of a low order Taylor expansion of the variance of the importance sampling estimator of an arbitrary test function to that of the simple Monte Carlo estimator; the test function itself vanishes from the expression as a consequence of this low order expansion.

In our context, allowing $W_{t-1}^{(i)}$ to denote the *normalized weights* of particle i at the end of time $t-1$, and $w_t^{(i)}$ to denote the *unnormalized* incremental weights of particle i during iteration t the ESS calculated using the current weight of each particle is simply:

$$\text{ESS}_t = \left[\sum_{j=1}^{N}\left(\frac{W_{t-1}^{(j)}w_t^{(j)}}{\sum_{k=1}^{N}W_{t-1}^{(k)}w_t^{(k)}}\right)^2\right]^{-1} = \frac{\left(\sum_{j=1}^{N}W_{t-1}^{(j)}w_t^{(j)}\right)^2}{\sum_{k=1}^{N}\left(W_{t-1}^{(k)}\right)^2\left(w_t^{(k)}\right)^2} \tag{23}$$

It's clearly appropriate to use this quantity (which corresponds to the coefficient of variation of the current normalized importance weights) to assess weight degeneracy and to make decisions about appropriate resampling times (cf. Del Moral et al. (2012)) but it is rather less apparent that it's the correct quantity to consider when adaptively specifying a sequence of distributions in an SMC sampler.

The ESS of the current sample weights tells us about the accumulated mismatch between proposal and target distributions (on an extended space including the full trajectory of the sample paths) since the last resampling time. Fixing either the relative or absolute reduction in ESS between successive distributions does *not* lead to a common discrepancy between successive distributions unless resampling is conducted after every iteration as will be demonstrated below.

When specifying a sequence of distributions it is natural to aim for a similar discrepancy between each pair of successive distributions. In the context of effective sample size, the natural question to ask is consequently, how large can we make $\alpha_t - \alpha_{t-1}$ whilst ensuring that π_t remains sufficiently similar to π_{t-1}. One way to measure the discrepancy would be to consider how good an importance sampling proposal π_{t-1} would be for the estimation of expectations under π_t and a natural way to measure this is via the sample approximation of a Taylor expansion of the relative variance of such an estimator exactly as in the ESS.

Such a procedure leads us to a quantity which we have termed the *conditional* ESS

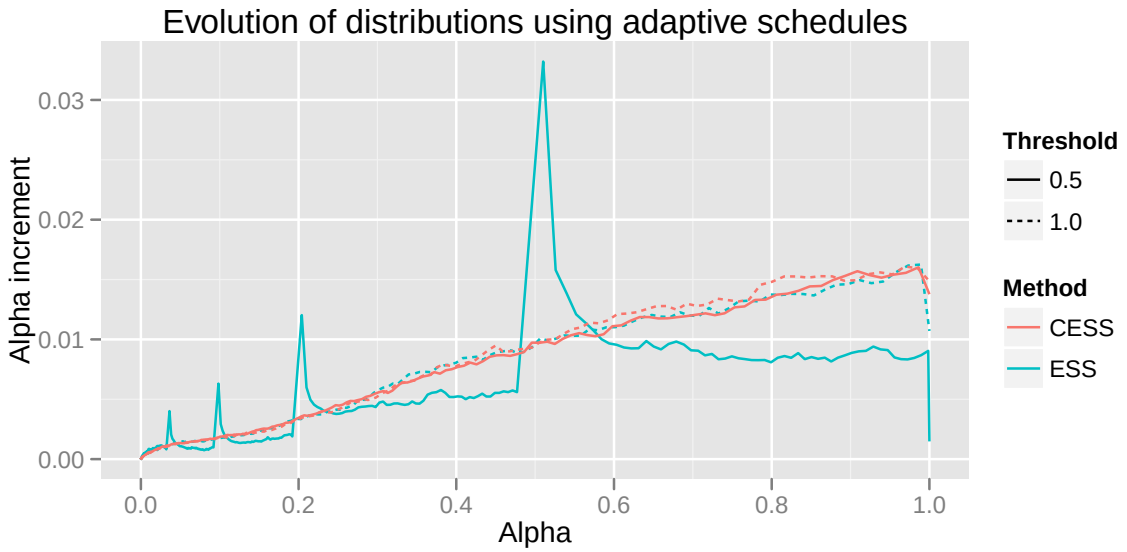

Figure 1: A typical plot of $\alpha_t - \alpha_{t-1}$ against α_t (for the Gaussian mixture model example of Section 4.1 using the SMC2 algorithm). The specifications of the adaptive parameter (ESS or CESS) are adjusted such that all four samplers use roughly the same number of distributions.

(CESS):

$$\text{CESS}_t = \left[\sum_{j=1}^{N} N W_{t-1}^{(j)} \left(\frac{w_t^{(j)}}{\sum_{k=1}^{N} N W_{t-1}^{(k)} w_t^{(k)}}\right)^2\right]^{-1} = \frac{\left(\sum_{j=1}^{N} W_{t-1}^{(j)} w_t^{(j)}\right)^2}{\sum_{k=1}^{N} \frac{1}{N} W_{t-1}^{(k)} \left(w_t^{(k)}\right)^2} \tag{24}$$

which is equal to the ESS only when resampling is conducted during every iteration. The factor of $1/N$ in the denominator arises from the fact that $\{W_{t-1}^{(i)}\}$ is normalized to sum to unity rather than to have expectation unity: the bracketed term coincides with a sample approximation (using the actual sample which is properly weighted to target π_{t-1}) of the expected sum of the unnormalized weights squared divided by the square of a sample approximation of the expected sum of unnormalized weights when considering sampling from π_{t-1} and targeting π_t by simple importance sampling.

Figure 1 shows the variation of $\alpha_t - \alpha_{t-1}$ with α_t when fixed reductions in ESS and CESS are used to specify the sequence of distributions both when resampling is conducted during every iteration (or equivalently, when the ESS falls below a threshold of 1.0) and when resampling is conducted only when the ESS falls below a threshold of 0.5. As is demonstrated in Section 4 the CESS-based scheme leads to a reduction in estimator variance of around 20% relative to a manually tuned (quadratic; see Section 4.1.1) schedule while the ESS-based strategy provides little improvement over the linear case unless resampling is conducted during every iteration.

In addition to providing a significantly better performance at essentially no cost, the use of the CESS emphasizes the purpose of the adaptive specification of the sequence of distributions: to produce a sequence in which the difference between each successive pair is the same (when using the CESS one is seeking to ensure that the variance of the importance weights one would arrive at if using π_{t-1} as a proposal for π_t is constant).

3.3.3 Adaptive Specification of Proposals

The SMC sampler is remarkably robust to the mixing speed of MCMC kernels employed as can be seen in the empirical study below. However, as with any sampling algorithms, faster mixing doesn't harm performance and in some cases will considerably improve it. In the particular case of Metropolis-Hastings kernels, the mixing speed relies on adequate proposal scales.

We adopt a simpler approach based on Jasra et al. (2010). They applied an idea used within adaptive MCMC methods (Andrieu and Moulines, 2006) to SMC samplers by using variance of parameters estimated from its particle system approximation as the proposal scale for the next iteration, suitably scaled with reference to the dimension of the parameters to be proposed. Although, in practice we found that such an automatic approach does not always lead to optimal acceptance rates it generally produces satisfactory results and is simple to implement. In difficult problems alternative approaches to adaptation could be employed; one approach demonstrated in Jasra et al. (2010) is to simply employ a pair of acceptance rate thresholds and to alter the proposal scale from the simply estimated value whenever the acceptance rate falls outside those threshold values.

More sohisticated proposal strategies could undoubtedly improve performance further and warrant further investigation. One possible approach is using the Metropolis adjusted Langevin algorithm (MALA; see Roberts and Tweedie (1996)). In summary, MALA derives a Metropolis-Hastings proposal kernel for a target π which satisfies suitable differentiability and positivity conditions, from the Langevin diffusion,

$$\mathrm{d}\, L_t = \frac{1}{2}\nabla \log \pi(L_t)\,\mathrm{d}\, t + \mathrm{d}\, B_t$$

where B_t is the standard Brownian motion. Given a state X_{n-1}, a new state is proposed by discrete approximation to the above diffusion. That is, for a fixed $h > 0$,

$$X_n \sim \mathcal{N}\Big(X_{n-1} + \frac{1}{2}\nabla \log \pi(X_{n-1}), hI_d\Big) \tag{25}$$

where I_d is the identify matrix and d is the dimension of the state space. The new proposed state is accepted or rejected through the usual Metropolis-Hastings algorithm. Compared to a "vanilla" random walk, which often has very robust theoretical properties, MALA is attractive when it is possible and its convergence conditions (Roberts and Tweedie, 1996) can be met, because only one discrete approximation parameter h needs to be tuned for optimal performance. In addition, results from Roberts and Rosenthal (2001) suggested that MALA can be more efficient than a random walk when using optimal scalings. We could also use the particle approximation at time index $t = n-1$ to estimate the covariance matrix of π_n and thus tune the scale h on-line. As these algorithms are known to be somewhat sensitive to scaling, and we seek approaches robust enough to employ with little user intervention, we have not investigated this strategy here.

3.4 An Automatic, Generic Algorithm

With the above refinements, we are ready to implement the SMC2 algorithm with minimal tuning and application-specific effort while providing robust and accurate estimates of the model evidence $p(\boldsymbol{y}|M_k)$. First the geometric annealing scheme that connects the prior $\pi(\theta_k|M_k)$ and the posterior $\pi(\theta_k|\boldsymbol{y}, M_k)$, provides a smooth path for a wide range of problems.

Second, the actual annealing schedule under this scheme can be determined through the adaptive schedule as described above. The advantage of the adaptive schedule will be shown empirically later.

Third, we can adaptively specify the Metropolis random walk (or MALA) scales through the estimation of their scaling parameters as the sampler iterates. In contrast to the MCMC setting, where such adaptive algorithms will usually require a burn-in period, which will not be used for further estimation, in SMC, the variance and covariance estimates come at almost no cost, as all the samples will later be used for marginal likelihood estimation. Additionally, adaptation within SMC does not require separate theoretical justification – something which can significantly complicate the development of effective, theoretically justified schemes in the MCMC setting. Alternatively, we can also specify the proposal scales in a deterministic, but sensible way. Since SMC algorithms are relatively robust to the change of scales, such deterministic scales will not require the same degree of tuning as is required to obtain good performance in MCMC algorithms.

The adaptive strategy can be applied to both algorithms directly. The applicability to the SMC3 algorithm depends on the nature of the sequence of distributions. We outline the strategy in Algorithm 4.

Algorithm 4 An Automatic, Generic Algorithm for Bayesian Model Comparison

Accuracy control

Set constant $\text{CESS}^\star \in (0, 1)$, using a small pilot simulation if necessary.

Initialization: Set $t \leftarrow 0$.

Perform the *Initialization* step as in Algorithm 1 or 2

Iteration: Set $t \leftarrow t + 1$

Step size selection

Use a binary search to find $\alpha^\star$ such that $\text{CESS}_{\alpha^\star} = \text{CESS}^\star$

Set $\alpha_t \leftarrow \alpha^\star$ if $\alpha^\star \leq 1$, otherwise set $\alpha_t \leftarrow 1$

Proposal scale calibration

Computing the importance sampling estimates of first two moments of parameters.

Set the proposal scale of the Markov proposal K_t with the estimated parameter variances.

Perform the *Iteration* step as in Algorithm 1 or 2 with the found α_t and proposal scales.

Repeat the *Iteration* step *until* $\alpha_t = 1$ then set $T = t$.

As laid out above, the algorithms require minimal tuning. Its robustness, accuracy and efficiency will be shown empirically in Section 4. We shall also note that SMC1 is less straightforward as the between model moves still require effort to design and implement. In SMC3, the specification of the sequences between posterior distributions are less generic compared to the geometric annealing scheme in SMC2. However, the adaptive schedule and automatic tuning of MCMC proposal scales, both can be applied in these two algorithms in principal.

Although further enhancements and refinements are clearly possible, we focus in the remainder of this article on this simple, generic algorithm which can be easily implemented in any application and has proved sufficiently powerful to provide good estimation in the examples we have encountered thus far.

4 Illustrative Applications

In this section, we will use three examples to illustrate the algorithms. The Gaussian mixture model is discussed first, with implementations for all three SMC algorithms with comparison to RJMCMC and PMCMC. It will be shown that all five algorithms agree on the results while the performance in terms of Monte Carlo variance varies considerably. It will also be demonstrated how the adaptive refinements of the algorithms behaves in practice. We will reach the conclusion that considering ease of implementation, performance and generality, the SMC2 algorithm is most promising among all three strategies.

Then two more realistic examples, a nonlinear ODE model and a Positron Emission Tomography compartmental model are used to study the performance and robustness of algorithm SMC2 compared to AIS and PMCMC. Various configurations of the algorithms are considered including both sequential and parallelized implementations.

The `C++` implementations of all examples can be found at `https://github.com/zhouyan/vSMC`.

4.1 Gaussian Mixture Model

Since Richardson and Green (1997), the Gaussian mixture model (GMM) has provided a canonical example of a model-order-determination problem. We use the model formulation of Del Moral et al. (2006b) to illustrate the efficiency and robustness of the methods proposed in this paper compared to other approaches. The model is as follows; data $\boldsymbol{y} = (y_1, \ldots, y_n)$ are independently and identically distributed as

$$y_i|\theta_r \sim \sum_{j=1}^{r} \omega_j \mathcal{N}(\mu_j, \lambda_j^{-1})$$

where $\mathcal{N}(\mu_j, \lambda_j^{-1})$ denotes the Normal distribution with mean μ_j and precision λ_j; $\theta_r = (\mu_{1:r}, \lambda_{1:r}, \omega_{1:r})$ and r is the number of components in each model. The parameter space is thus $\mathbb{R}^r \times (\mathbb{R}^+)^r \times \Delta_r$ where $\Delta_r = \{\omega_{1:r} : 0 \leq \omega_j \leq 1; \sum_{j=1}^r \omega_j = 1\}$ is the standard r-simplex. The priors which are the same for each component are taken to be $\mu_j \sim \mathcal{N}(\xi, \kappa^{-1})$, $\lambda_j \sim \mathcal{G}(\nu, \chi)$ and $\omega_{1:r} \sim \mathcal{D}(\rho)$ where $\mathcal{D}(\rho)$ is the symmetric Dirichlet distribution with parameter ρ and $\mathcal{G}(\nu, \chi)$ is the Gamma distribution with shape ν and scale χ. The prior parameters are set in the same manner as in Richardson and Green (1997). Specifically, let $y_{\min}$ and $y_{\max}$ be the minimum and maximum of data $\boldsymbol{y}$, the prior parameters are set such that

$$\xi = (y_{\max} + y_{\min})/2, \quad \kappa = (y_{\max} - y_{\min})^{-2}, \quad \nu = 2, \quad \chi = 50\kappa, \quad \rho = 1$$

The data is simulated from a four components model with $\mu_{1:4} = (-3, 0, 3, 6)$, and $\lambda_j = 2$, $\omega_j = 0.25$, $j = 1, \ldots, 4$.

We consider several algorithms. First the RJMCMC algorithm as in Richardson and Green (1997), and second an implementation of the SMC1 algorithm. Next AIS, PMCMC and SMC2 are used for within-model simulations. The last is an implementation of the SMC3 algorithm. In all the algorithms, the local move which does not change the dimension of the model is constructed as a composition of Metropolis-Hastings random walk kernels:

- Update $\mu_{1:r}$ using a multivariate Normal random walk proposal.
- Update $\lambda_{1:r}$ using a multivariate Normal random walk on logarithmic scale, i.e., on $\log \lambda_j$, $j = 1, \ldots, r$.
- Update $\omega_{1:r}$ using a multivariate Normal random walk on logit scale, i.e., on ω_j/ω_r, $j = 1, \ldots, r-1$.

The RJMCMC, SMC1 and SMC3 algorithms use two additional reversible jump moves. The first is a combine and split move; the second is a birth and death move. Both are constructed in the same manner as in Richardson and Green (1997). Also in these implementations, an adjacency condition was imposed on the means $\mu_{1:r}$, such that $\mu_1 < \mu_2 < \cdots < \mu_r$. No such restriction was used for other algorithms.

In SMC1, SMC2 and PMCMC implementations, the distributions are chosen with a geometric schedule, i.e., as in Equation (15) for SMC1 and Equation (16) for the other two. This annealing scheme has been used in Del Moral et al. (2006b); Jasra et al. (2007a) and many other works. The geometric scheme can also be seen in Calderhead and Girolami (2009) for PMCMC tempering. A schedule $\alpha(t/T) = (t/T)^p$, with $p = 2$ was used. The rationale behind this particular schedule can be seen in Calderhead and Girolami (2009) and other values of p were also tried while $p \approx 2$ performs best in this particular example. The adaptive schedule was also implemented for SMC2 and AIS algorithms.

The proposal scales for each block of the random walks are specified dynamically according to values of $\alpha(t/T)$ for the SMC2 and AIS algorithms and also manually tuned for other algorithms such that the acceptance rates fall in $[0.2, 0.5]$. Later for the SMC2 and AIS algorithms, we also consider adaptive schedule of the distribution specification parameter $\alpha(t/T)$ and the proposal scales of the random walks.

For SMC2, SMC3 and AIS we consider both the direct estimator and the path sampling estimator. For PMCMC we consider the path sampling estimator.

4.1.1 Results

The SMC1 implementation uses 10^4 particles and 500 distributions. The RJMCMC implementation uses 5×10^6 iterations in addition to 10^6 iterations of burn-in period. The resulting estimates of model probabilities are shown in Table 2.

The SMC2, SMC3 and AIS implementations use $1,000$ particles and 500 iterations. The PMCMC implementation uses 50 chains and 10^4 iterations in addition to 10^4 iterations of burn-in period — these implementations have approximately equal computational costs. From the results obtained under the SMC1 and RJMCMC algorithms it is clear that, in this particular example, simulations for models with fewer than ten components are adequate to characterize the model space. Therefore, under this configuration, the cost is roughly the same in terms of computational resources as that of the SMC1 and RJMCMC algorithms. From the results of RJMCMC and SMC1, we consider four and five components models (i.e., the true model and the most competitive amongst the others). The estimates are shown

Algorithm	Quantity	Number of components						
		≤ 2	3	4	5	6	7	≥ 8
SMC1	$\mathbb{P}(M = M_k)$	0	0.00257	0.886	0.103	0.00715	0.00128	0
	$\log B_{4,k}$	∞	5.84	0	2.15	4.82	6.54	∞
RJMCMC	$\mathbb{P}(M = M_k)$	0	0.000526	0.887	0.103	0.00623	0.00324	0
	$\log B_{4,k}$	∞	6.56	0	2.15	4.96	5.61	∞

Table 2: Gaussian mixture model estimates obtained via SMC1 and RJMCMC

Quantity	Algorithms						
	SMC2-DS	SMC2-PS	SMC3-DS	SMC3-PS	AIS-DS	AIS-PS	PMCMC
$\log B_{4,5}$	2.15	2.15	2.16	2.21	2.16	2.17	2.63
SD	0.25	**0.22**	0.61	0.62	1.12	1.10	0.41

Table 3: Gaussian mixture model estimates obtained via SMC2, SMC3, AIS and PMCMC

in Table 3 which, like all of the other tables in this section, summarises the Monte Carlo variability of 100 replicate runs of each algorithm..

From Tables 2 and 3, it can be seen that the unbiased estimators (RJMCMC, SMC1, SMC2-DS, SMC3-DS and AIS-DS) agree with each other. Among the path sampling estimators, SMC2-PS and AIS-PS have little bias. SMC3-PS shows a little more bias. The PMCMC algorithm has a considerable larger bias as the number of distributions is relatively small (as noted previously, a larger number will negatively affect the mixing speed).

In terms of Monte Carlo variance, in Table 3, SMC2 clearly has an advantage compared to its no-resampling variant, AIS. The differences of Monte Carlo SD between SMC2, SMC3 and PMCMC, although they do not affect model selection in this particular example, are considerable.

Effects of resampling It is clear from these results that resampling (when required) can substantially improve the estimation of normalising constants within an SMC framework. This doesn't contradict the statement in Del Moral et al. (2006b) which suggests that resampling may not much help when the normalising constant is the object of interest: the theoretical argument which supports this relies upon the assumption that the Markov kernel used to mutate the particles mixes extremely rapidly and the result is obtained under the assumption that resampling is performed after every iteration. When the Markov kernel is not so rapidly mixing, the additional stability provided by the resampling operation can outweight the attendant increase in Monte Carlo variance and that is what we observed here (and in the case of the other examples considered below; results not shown).

Effects of adaptive schedules To assess the evolution of distributions with an adaptive schedule, we consider the relation between $\alpha_t - \alpha_{t-1}$ and α_t. As stated before, one of the motivations of using CESS for adaptive placement of distribution is to ensure that α_t evolves the same path regardless the resampling strategies. Figure 1 shows the evolution of α_t when fixing ESS or CESS and resampling every iteration or only when ESS $< N/2$. As shown in the plot, when fixing CESS, the evolution of the distributions is not affected by the resampling strategy. In contrast, fixing ESS yields a sequence of distributions which depends strongly upon the resampling strategy.

In terms of the actual performance when using the CESS adaptive strategy in the SMC2 and AIS algorithms, a reduction of standard deviation of 20% was observed comparing to $\alpha(t/T) = (t/T)^2$, the one shown in Table 3. When applied to the SMC3 algorithm, 50% reduction was observed. If the ESS adaptive strategy is used instead, similar standard deviation reduction is observed when resampling is performed every iteration but no significant effect was observed when resampling was only performed when ESS $< N/2$ (i.e., using ESS rather than CESS entirely eliminated the benefit).

Effects of adaptive proposal scales When using the SMC2 algorithm, if the adaptive strategy of Andrieu and Moulines (2006) is applied, where the important sampling estimates of the variance of parameters are included in the adaptation, the acceptance rates fall within $[0.2, 0.5]$ dynamically without manual tuning as for the results in Table 3. It should be noted that in this particular example, it is the variance of $\log \lambda_i$ being estimated as the corresponding random walk block operates on the log scale. The same principle applies to the weight parameters, which have random walks on logit scale. Approximately 20% reduction in standard deviation was observed.

4.2 Nonlinear Ordinary Differential Equations

The example from the previous section suggests that SMC2 performs well relative to the other SMC possibilities. Given the wide range of settings in which it can be easily deployed, we will now concentrate further on this method. It also suggests that in the simple case of Gaussian mixtures, a complete adaptive strategy for both distribution specification and proposal scales works well. In this section, this will now be further explored in a more complex model, a nonlinear ordinary differential equations system. This model, which was studied in Calderhead and Girolami (2009), is known as the Goodwin model. The ODE system, for an m-component model, is:

$$\begin{aligned}
\frac{\mathrm{d}\, X_1(t)}{\mathrm{d}\, t} &= \frac{a_1}{1 + a_2 X_m(t)^\rho} - \alpha X_1(t) \\
\frac{\mathrm{d}\, X_i(t)}{\mathrm{d}\, t} &= k_{i-1} X_{i-1}(t) - \alpha X_i(t) && i = 2, \ldots, m \\
X_i(0) &= 0 && i = 1, \ldots, m
\end{aligned}$$

The parameters $\{\alpha, a_1, a_2, k_{1:m-1}\}$ have common prior distribution $\mathcal{G}(0.1, 0.1)$. Under this setting, $X_{1:m}(t)$ can exhibit either unstable oscillation or a constant steady state. The data are simulated for $m = \{3, 5\}$ at equally spaced time points from 0 to 60, with time step 0.5. The last 80 data points of $(X_1(t), X_2(t))$ are used for inference. Normally-distributed noise with standard deviation $\sigma = 0.2$ is added to the simulated data. Following Calderhead and Girolami (2009), the variance of the additive measurement error is assumed to be known. Therefore, the posterior distribution has $m + 2$ parameters for an m-component model.

As shown in Calderhead and Girolami (2009), when $\rho > 8$, due to the possible instability of the ODE system, the posterior can have a considerable number of local modes. In this example, we set $\rho = 10$. Also, as the solution to the ODE system is somewhat unstable, slightly different data can result in very different posterior distributions.

4.2.1 Results

We compare results from the SMC2 and PMCMC algorithms. For the SMC implementation, $1,000$ particles and 500 iterations were used, with the distributions specified specified by Equation (16), with $\alpha(t/T) = (t/T)^5$, or via the completely adaptive specification. For the PMCMC algorithm, $50,000$ iterations are performed for burn-in and another $10,000$ iterations are used for inference. The same tempering as was used for SMC is used here. Note that, in a sequential implementation of PMCMC, with each iteration updating one local chain and attempting a global exchange, the computational cost of after burn-in iterations is roughly the same as the entire SMC algorithm. In addition, changing T within the range of the number of cores available does not substantially change the computational cost of a generic parallel implementation of the PMCMC algorithm. We compare results from $T = 10, 30, 100$.

The results for data generated from the simple model ($m = 3$) and complex model ($m = 5$), again summarising variability amongst 100 runs of each algorithm, are shown in Table 4 and 5, respectively.

As shown in both cases, the number of distributions can affect the performance of PMCMC algorithms considerably. When using 10 distributions, large bias from numerical integration for path sampling estimator was observed, as expected. With 30 distributions, the performance is comparable to the SMC2 sampler, though some bias is still observable. With

T	Proposal Scales	Annealing Scheme	Algorithm	Marginal likelihood ($\log p(\theta_k\|\boldsymbol{y}) \pm$ SD) $m=3$	Marginal likelihood ($\log p(\theta_k\|\boldsymbol{y}) \pm$ SD) $m=5$	Bayes factor $\log B_{3,5}$
10	Manual	Prior (5)	PMCMC	-109.7 ± 3.2	-120.3 ± 2.5	10.6 ± 3.8
30				*-105.0 ± 1.2*	*-116.1 ± 2.2*	*11.2 ± 2.5*
100				-134.7 ± 7.9	-144.1 ± 6.2	9.4 ± 11.2
500	Manual	Prior (5)	SMC2-DS	-104.6 ± 2.0	-112.7 ± 1.8	8.1 ± 2.8
			SMC2-PS	-104.5 ± 1.8	-112.7 ± 1.5	8.2 ± 2.5
500	Manual	Adaptive	SMC2-DS	-104.5 ± 1.1	-112.7 ± 1.1	8.1 ± 1.6
			SMC2-PS	-104.6 ± 1.0	-112.8 ± 1.0	8.2 ± 1.5
500	Adaptive	Adaptive	SMC2-DS	-104.5 ± 0.5	-112.7 ± 0.4	8.1 ± 0.8
			SMC2-PS	**-104.6 ± 0.4**	**-112.8 ± 0.3**	**8.1 ± 0.6**

Table 4: Results for non-linear ODE models with data generated from simple model. *italic*: Minimum variance for the same algorithm. **Bold**: Minimum variance for all samplers.

T	Proposal Scales	Annealing Scheme	Algorithm	Marginal likelihood ($\log p(\theta_k\|\boldsymbol{y}) \pm$ SD) $m=3$	Marginal likelihood ($\log p(\theta_k\|\boldsymbol{y}) \pm$ SD) $m=5$	Bayes factor $\log B_{5,3}$
10	Manual	Prior (5)	PMCMC	-1651.0 ± 27.9	-85.1 ± 36.6	1565.9 ± 42.1
30				*-1639.7 ± 7.4*	*-78.9 ± 11.2*	*1560.8 ± 12.8*
100				-1624.6 ± 15.7	-75.7 ± 24.8	1548.9 ± 25.6
500	Manual	Prior (5)	SMC2-DS	-1640.7 ± 10.8	-78.5 ± 9.8	1562.2 ± 10.1
			SMC2-PS	-1640.8 ± 8.4	-79.2 ± 7.9	1561.6 ± 8.5
500	Manual	Adaptive	SMC2-DS	-1639.7 ± 6.9	-78.6 ± 4.8	1561.1 ± 7.1
			SMC2-PS	-1640.1 ± 5.4	-78.8 ± 3.7	1561.3 ± 6.8
500	Adaptive	Adaptive	SMC2-DS	-1639.8 ± 2.2	-79.4 ± 1.7	1560.4 ± 3.1
			SMC2-PS	**-1640.2 ± 1.9**	**-78.5 ± 1.5**	**1561.7 ± 2.3**

Table 5: Results for non-linear ODE models with data generated from complex model. Number *italic*: Minimum variance for the same algorithm. **Bold**: Minimum variance for all samplers.

100 distributions, there is a much larger variance because, with more chains, the information travels more slowly from rapidly mixing chains to slowly mixing ones and consequently the mixing of the overall system is inhibited.

The SMC algorithm provides results comparable to the best of three PMCMC implementations in essentially all settings, including one in which both the annealing schedule and proposal scaling were fully automatic. In fact, the completely adaptive strategy was the most successful.

It can be seen that increasing the number of distributions not only reduces the bias of numerical integration for path sampling estimator, but also reduces the variance considerably. On the other hand increasing the number of particles can only reduce the variance of the estimates, in accordance with the central limit theorem (see Del Moral et al. (2006b) for the standard estimator and extensions for path sampling estimator, Proposition 1) (as the bias arises from the numerical integration scheme).

4.3 Positron Emission Tomography Compartmental Model

It is now interesting to compare the proposed algorithm with other state-of-art algorithms using a more realistic example.

Positron Emission Tomography (PET) is a technique used for studying the brain *in vivo*, most typically when investigating metabolism or neuro-chemical concentrations in either normal or patient groups. Given the nature and number of observations typically recorded in time, PET data is usually modeled with linear differential equation systems. For an overview of PET compartmental model see Gunn et al. (2002). Given data $(y_1, \ldots, y_n)^{\mathrm{T}}$, an m-compartmental model has generative form:

$$y_j = C_T(t_j; \phi_{1:m}, \theta_{1:m}) + \sqrt{\frac{C_T(t_j; \phi_{1:m}, \theta_{1:m})}{t_j - t_{j-1}}}\varepsilon_j \tag{26}$$

$$C_T(t_j; \phi_{1:m}, \theta_{1:m}) = \sum_{i=1}^{m} \phi_i \int_0^{t_j} C_P(s) e^{-\theta_i (t_j - s)} \, \mathrm{d}\, s \tag{27}$$

where t_j is the measurement time of y_j, ε_j is additive measurement error and input function C_P is (treated as) known. The parameters $\phi_1, \theta_1, \ldots, \phi_m, \theta_m$ characterize the model dynamics. See Zhou et al. (2013) for applications of Bayesian model comparison for this class of models and details of the specification of the measurement error. In the simulation results below, ε_j are independently and identically distributed according to a zero mean Normal distribution of unknown variance, σ^2, which was included in the vector of model parameters.

Real neuroscience data sets involve a very large number ($\sim 200,000$ per brain) of time series, which are typically somewhat heterogeneous. Figure 2 shows estimates of $V_D = \sum_{j=1}^{m} \phi_j / \theta_j$ from a typical PET scan (generated using SMC2 as will be discussed later). Robustness is therefore especially important. An application-specific MCMC algorithm was developed for this problem in Zhou et al. (2013). A significant amount of tuning of the algorithms was required to obtain good results.The results shown in Figure 2 are very close to those of Zhou et al. (2013) but, as is shown later, they were obtained with almost no manual tuning effort and at similar computational cost.

For SMC and PMCMC algorithms, the requirement of robustness means that the algorithm must be able to calibrate itself automatically to different data (and thus different posterior surfaces). A sequence of distributions which performs well for one time series may not perform even adequately for another series. Specification of proposal scales that produces fast-mixing kernels for one data series may lead to slow mixing for another. In the following experiment, we will use a single simulated time series, and choose schedules that performs both well and poorly for this particular time series. The objective is to see if the algorithm can recover from a relatively poorly specified schedule and obtain reasonably accurate results.

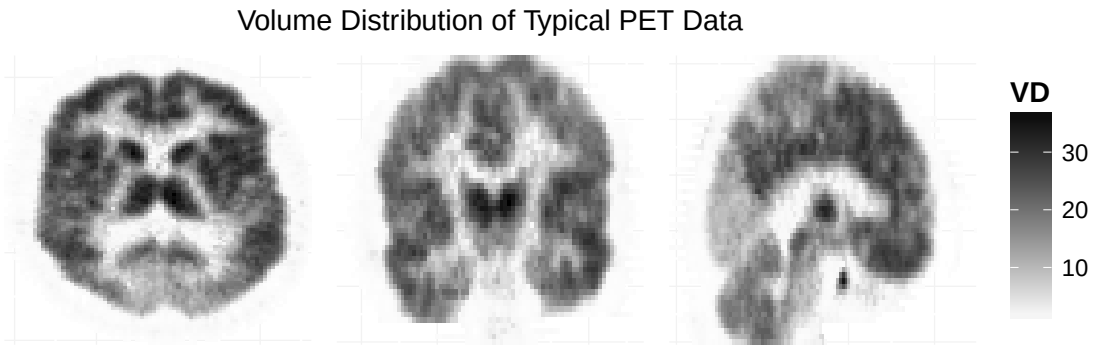

Figure 2: Estimates of V_D from a single PET scan as found using SMC2. The data shows that the volume of distribution exhibits substantial spatial variation. Note that each pixel in the image represent an estimate from an individual time series data set. There are approximately a quarter million of them and each requires a Monte Carlo simulation to select a model.

4.3.1 Results

In this example we focus on the comparison between SMC2 and PMCMC. We also consider parallelized implementations of algorithms. In this case, due to its relatively small number of chains, PMCMC can be parallelized completely (and often cannot fully utilize the hardware capability if a naïve approach to parallelization is taken; while we appreciate that more sophisticated parallelization strategies are possible, these depend instrinsicially upon the model under investigation and the hardware employed and given our focus on automatic and general algorithms, we don't consider such strategies here). The PMCMC algorithm under this setting is implemented such that each chain is updated at each iteration. Further, for the SMC algorithms, we consider two cases. In the first we can parallelize the algorithm completely (in the sense that each core has a single particle associated with it). In this setting we use a relatively small number of particles and a larger number of time steps. In the second, we need a few passes to process a large number of particles at each time step, and accordingly we use fewer time steps to maintain the same total computation time. These two settings allow us to investigate the trade-off between the number of particles and time steps. In both implementations, we consider three schedules, $\alpha(t/T) = t/T$ (linear), $\alpha(t/T) = (t/T)^5$ (prior), and $\alpha(t/T) = 1 - (1 - t/T)^5$ (posterior). In addition, the adaptive schedule based upon CESS is also implemented for the SMC2 algorithm.

Results from 100 replicate runs of the two algorithms under various regimes can be found in Table 6 and 7 for the marginal likelihood and Bayes factor estimates, respectively. The SMC algorithms consistently outperforms the PMCMC algorithms in the parallel settings. The Monte Carlo SD of SMC algorithms is typically of the order of one fifth of the corresponding estimates from PMCMC in most scenarios. In some settings with the smaller number of samples, the two algorithms can be comparable. Also at the lowest computational costs, the samplers with more time steps and fewer particles outperform those with the converse configuration by a fairly large margin in terms of estimator variance. It shows that with limited resources, ensuring the similarity of consecutive distributions, and thus good mixing, can be more beneficial than a larger number of particles. However, when the computational budget is increased, the difference becomes negligible. The robustness of SMC to the change of schedules is again apparent.

Effects of adaptive schedule A set of samplers with adaptive schedules are also used. Due to the nature of the schedule, it cannot be controlled to have exactly the same number of time steps as non-adaptive procedures. However, the CESS was controlled such that the average number of time steps are comparable with the fixed schedules and in most cases slightly less than the fixed numbers.

Proposal scales			Manual			Adaptive
Annealing scheme			Prior (5)	Posterior (5)	Adaptive	
T	N	Algorithm	Marginal likelihood estimates ($\log p(\theta_k\|\boldsymbol{y}) \pm$ SD)			
500	30	PMCMC	-39.1 ± 0.56	-926.8 ± 376.99		
500	192	SMC2-DS	*-39.2 ± 0.25*	*-39.7 ± 1.06*	*-39.2 ± 0.18*	**-39.1 ± 0.12**
		SMC2-PS	*-39.2 ± 0.25*	-91.3 ± 21.69	*-39.2 ± 0.18*	-39.1 ± 0.13
100	960	SMC2-DS	-39.3 ± 0.36	-40.6 ± 1.41	-39.2 ± 0.31	-39.2 ± 0.19
		SMC2-PS	-39.3 ± 0.35	302.1 ± 46.29	-39.3 ± 0.31	-39.2 ± 0.18
1000	30	PMCMC	-39.3 ± 0.46	-884.1 ± 307.88		
1000	192	SMC2-DS	*-39.2 ± 0.19*	*-39.4 ± 0.68*	*-39.2 ± 0.17*	**-39.1 ± 0.10**
		SMC2-PS	*-39.2 ± 0.19*	-66.0 ± 13.26	*-39.2 ± 0.17*	**-39.1 ± 0.10**
200	960	SMC2-DS	-39.2 ± 0.22	-39.8 ± 1.21	-39.2 ± 0.18	-39.1 ± 0.11
		SMC2-PS	-39.2 ± 0.22	175.5 ± 26.84	-39.2 ± 0.18	-39.2 ± 0.11
2000	30	PMCMC	-39.3 ± 0.28	-928.7 ± 204.93		
2000	192	SMC2-DS	-39.2 ± 0.14	*-39.3 ± 0.41*	-39.1 ± 0.12	-39.1 ± 0.07
		SMC2-PS	-39.2 ± 0.14	-51.2 ± 4.30	-39.2 ± 0.12	-39.1 ± 0.07
400	960	SMC2-DS	*-39.2 ± 0.13*	-39.4 ± 0.73	*-39.2 ± 0.11*	-39.2 ± 0.07
		SMC2-PS	*-39.2 ± 0.13*	106.0 ± 14.36	*-39.2 ± 0.11*	**-39.2 ± 0.06**
5000	30	PMCMC	-39.3 ± 0.21	-917.6 ± 129.54		
5000	192	SMC2-DS	-39.2 ± 0.09	*-39.2 ± 0.20*	-39.2 ± 0.08	-39.1 ± 0.04
		SMC2-PS	-39.2 ± 0.09	-43.8 ± 2.13	-39.2 ± 0.08	-39.1 ± 0.04
1000	960	SMC2-DS	*-39.2 ± 0.08*	-39.2 ± 0.31	*-39.2 ± 0.07*	**-39.2 ± 0.03**
		SMC2-PS	*-39.2 ± 0.08*	-65.7 ± 5.54	*-39.2 ± 0.07*	**-39.2 ± 0.03**

Table 6: Marginal likelihood estimates of two components PET model. T: Number of distributions in SMC and number of iterations used for inference in PMCMC. N: Number of particles in SMC and number chains in PMCMC. The PMCMC and SMC with $N = 192$ are completely N-way parallelized. SMC with $N = 960$ are $N/5$-way parallelized. *Italic*: Minimum variance for the same computational cost and the same proposal scales and annealing schemes. **Bold**: Minimum variance for the same computaitonal cost and all proposal scales and annealing schemes.

Proposal scales			Manual			Adaptive
Annealing scheme			Prior (5)	Posterior (5)	Adaptive	
T	N	Algorithm	Bayes factor estimates ($\log B_{2,1} \pm$ SD)			
500	30	PMCMC	1.7 ± 0.62	-70.9 ± 525.79		
500	192	SMC2-DS	*1.6 ± 0.27*	*1.3 ± 1.13*	*1.6 ± 0.20*	**1.6 ± 0.15**
		SMC2-PS	*1.6 ± 0.27*	-3.9 ± 30.02	*1.6 ± 0.20*	**1.6 ± 0.15**
100	960	SMC2-DS	1.6 ± 0.37	0.5 ± 1.55	1.6 ± 0.34	1.6 ± 0.21
		SMC2-PS	1.6 ± 0.37	-13.1 ± 66.30	1.6 ± 0.33	1.6 ± 0.21
1000	30	PMCMC	1.6 ± 0.49	-67.3 ± 400.21		
1000	192	SMC2-DS	*1.6 ± 0.21*	*1.5 ± 0.79*	1.6 ± 0.20	1.6 ± 0.13
		SMC2-PS	*1.6 ± 0.21*	-0.6 ± 15.47	1.6 ± 0.20	1.6 ± 0.13
200	960	SMC2-DS	1.6 ± 0.25	1.1 ± 1.25	1.6 ± 0.19	1.6 ± 0.12
		SMC2-PS	1.6 ± 0.24	-11.7 ± 34.68	*1.6 ± 0.18*	**1.6 ± 0.11**
2000	30	PMCMC	1.6 ± 0.31	-95.5 ± 264.74		
2000	192	SMC2-DS	*1.6 ± 0.14*	*1.6 ± 0.44*	1.6 ± 0.13	1.6 ± 0.09
		SMC2-PS	*1.6 ± 0.14*	1.6 ± 6.06	1.6 ± 0.13	1.7 ± 0.09
400	960	SMC2-DS	1.6 ± 0.16	1.5 ± 0.74	*1.6 ± 0.12*	**1.6 ± 0.08**
		SMC2-PS	1.6 ± 0.16	-4.2 ± 17.15	*1.6 ± 0.12*	**1.6 ± 0.08**
5000	30	PMCMC	1.6 ± 0.24	-60.3 ± 198.10		
5000	192	SMC2-DS	1.6 ± 0.10	*1.6 ± 0.23*	1.6 ± 0.09	1.6 ± 0.05
		SMC2-PS	1.6 ± 0.10	1.3 ± 2.98	1.6 ± 0.09	1.6 ± 0.05
1000	960	SMC2-DS	*1.6 ± 0.09*	1.6 ± 0.33	*1.6 ± 0.08*	**1.6 ± 0.04**
		SMC2-PS	*1.6 ± 0.09*	-0.2 ± 6.63	*1.6 ± 0.08*	**1.6 ± 0.04**

Table 7: Bayes factor $B_{2,1}$ estimates of two components PET model. T: Number of distributions in SMC and number of iterations used for inference in PMCMC. N: Number of particles in SMC and number chains in PMCMC. The PMCMC and SMC with $N = 192$ are completely N-way parallelized. SMC with $N = 960$ are $N/5$-way parallelized. *Italic*: Minimum variance for the same computational cost and the same schedule. **Bold**: Minimum variance for the same computational cost and all schedules.

	Number of grid points (compared to sampled iterations)			
Integration rule	$\times 1$	$\times 2$	$\times 4$	$\times 8$
Trapezoid	-52.2 ± 5.01	-45.5 ± 1.93	-42.1 ± 1.21	-40.5 ± 1.06
Simpson	-43.2 ± 1.39	-41.0 ± 1.10	-40.0 ± 1.04	-39.4 ± 1.04
Simpson $3/8$	-42.1 ± 1.21	-40.5 ± 1.06	-39.7 ± 1.04	-39.3 ± 1.04
Boole	-40.9 ± 1.09	-39.9 ± 1.04	-39.4 ± 1.04	-39.2 ± 1.05

Table 8: Path sampling estimator of marginal likelihood of two components PET model. The estimator was approximated using samples from SMC2 algorithm with $1,000$ particles and 20 iterations, with different numerical integration strategies. Large sample result (see Table 6) shows that an unbiased estimate is -39.2.

It is found that, with little computational overhead, adaptive schedules do provide the best results (or very nearly so) and do so without user intervention. The reduction of Monte Carlo SD varies among different configurations. For moderate or larger number of distributions, a reduction about 50% was observed. In addition, it shall be noted that, in this example, the bias of the path sampling estimates are much more sensitive to the schedules than the previous Gaussian mixture model example. A vanilla linear schedule does not provide a low bias estimator at all even when the number of distributions is increased to a considerably larger number. The prior schedule though provides a nearly unbiased estimator, there is no clear theoretical evidence showing that this shall work for other situations. The adaptive schedule, without any manual calibration, can provide a nearly unbiased estimator, even when path-sampling is employed, in addition to potential variance reduction.

Bias reduction for path sampling estimator As seen in Table 6 and 7, a bad choice of schedule $\alpha(t/T)$ can results in considerable bias for the basic path sampling estimator, here for SMC2-PS but the problem is independent of the mechanism by which the samples are obtained. Increasing the number of iterations can reduce this bias but at the cost of additional computation time. As outlined in Section 3.3.1, in the case of the SMC algorithms discussed here, it is possible to reduce the bias without increasing computational cost significantly. To demonstrate the bias reduction effect, we constructed SMC sampler for the above PET example with only $1,000$ particles and about 20 iterations specified using the CESS based adaptive strategy. The path sampling estimator was approximated using Equation (21) as well as other higher order numerical integration or by integrating over a grid that contains $\{\alpha_t\}$ at which the samples was generated. The results are shown in Table 8

Real data results Finally, the methodology of SMC2-PS was applied to measured positron emission tomography data using the same compartmental setup as in the simulations. The data shown in Figure 2 comes from a study into opioid receptor density in Epilepsy, with the data being described in detail in Jiang et al. (2009). It is expected that there will be considerable spatial smoothness to the estimates of the volume of distribution, as this is in line with the biology of the system being somewhat regional. Some regions will have much higher receptor density while others will be much lower, yielding higher and lower values of the volume of distribution, respectively. While we did not impose any spatial smoothness but rather estimated the parameters independently for each time series at each spatial location, as can be seen, smooth spatial estimates of the volume of distribution consistent with neurological understanding were found using the approach. This method is computationally feasible for the entire brain on a voxel-by-voxel basis, due to the ease of parallelization of the SMC algorithm. In the analysis performed here, 1000 particles were used, along with an adaptive schedule using a constant CESS$^\star = 0.999$, resulting in about 180 to 200 intermediate distributions. The model selection results are very close to those obtained by a previous study of the same data (Zhou et al., 2013), although the present approach requires much less implementation effort and has roughly the same computational cost.

4.4 Summary

These three illustrative applications have essentially shown three aspects of using SMC as a generic tool for Bayesian model selection. Firstly, as seen in the Gaussian mixture model example, all the different variants of SMC proposed, including both direct and path sampling versions, produce results which are competitive with other model selection methods such as RJMCMC and PMCMC. In addition, in this somewhat simple example, SMC2 performs well, and leads to low variance estimates with no appreciable bias. The effect of adaptation was studied more carefully in the nonlinear ODE example, and it was shown that using both adaptive selection of distributions as well as adaptive proposal variances leads to very competitive algorithms, even against those with significant manual tuning. This suggests that an automatic process of model selection using SMC2 is possible. In the final example, considering the easy parallelization of algorithms such as SMC2 suggests that great gains in variance estimation can be made using settings such as GPU computing for application where computational resources are of particular importance (such as in image analysis as in the PET example). It is also clear that the negligible cost of the bias reduction techniques described means that one should always consider using these to reduce the bias inherent in path sampling estimation.

5 Theoretical Considerations

The convergence results for the standard estimator can be found in Del Moral et al. (2006b) and references therein. In this paper, given our advocation of SMC2-PS, we extend the results for the path sampling estimator from SMC samplers. Here we present Proposition 1, which is specific to path sampling estimator using the simplest Trapezoidal approach to numerical integration. It follows as a simple corollary to a more general result given in Appendix A which could be used to characterize more general numerical integration schemes.

Proposition 1. *Under the same regularity conditions as are required for the central limit theorem given in Del Moral et al. (2006b) to hold, given a SMC sampler that iterates over a sequence of distributions* $\{\pi_t = q_{\alpha_t}/Z_{\alpha_t}\}_{t=0}^{T}$ *and applies multinomial resampling at each iteration, the path sampling estimator,* $\widehat{\Xi}_T^N$*, as defined in Equation* (21) *obeys a central limit theorem in the following sense: Let* $\xi_t(\cdot) = \frac{\mathrm{d}\log q_\alpha(\cdot)}{\mathrm{d}\,\alpha}\Big|_{\alpha=\alpha_t}$*,* $\beta_0 = \alpha_0/2$*,* $\beta_T = \alpha_T/2$ *and for* $t \in \{1,\ldots,T-1\}$ $\beta_t = (\alpha_{t+1} - \alpha_{t-1})/2$*, then, provided* ξ_t *is bounded:*

$$\lim_{N\to\infty} \sqrt{N}(\widehat{\Xi}_T^N - \Xi_T) \xrightarrow{D} \mathcal{N}(0, V_T(\xi_{0:T})) \tag{28}$$

where V_t*,* $0 \leq t \leq T$ *is defined by the following recursion:*

$$V_0(\xi_0) = \beta_0^2 \int \pi_0(x_0)(\xi_0(x_0) - \pi_0(\xi_0))^2 dx_0 \tag{29}$$

$$\begin{aligned} V_t(\xi_{0:t}) =& V_{t-1}\left(\xi_{0:t-2}, \xi_{t-1} + \frac{\beta_t}{\beta_{t-1}}\frac{\pi_t(\cdot)}{\pi_{t-1}(\cdot)}\int K_t(\cdot, x_t)(\xi_t(x_t) - \pi_t(\xi_t))\,\mathrm{d}\,x_t\right) \\ &+ \beta_t^2 \int \frac{\pi_t(x_{t-1})^2}{\pi_{t-1}(x_{t-1})} K_t(x_{t-1}, x_t)(\xi_t(x_t) - \pi_t(\xi_t))^2)\,\mathrm{d}\,x_{t-1}\,\mathrm{d}\,x_t. \end{aligned} \tag{30}$$

We note that much recent analysis of SMC algorithms has focussed on relaxing the relatively strong assumptions used in the results upon which this result is based — looking at more general resampling schemes (Del Moral et al., 2012) and relaxing compactness assumptions (Whiteley, 2013) for example. However, we feel that this simple result is sufficient to show the relationship between the path sampling and simple estimators and that in this instance the relatively simplicity of the resulting expression justifies these stronger assumptions.

6 Discussion

It has been shown that SMC is an effective Monte Carlo method for Bayesian inference for the purpose of model comparison. Three approaches have been outlined and investigated in

several illustrative applications including the challenging scenarios of nonlinear ODE models and PET compartmental systems. The proposed strategy is always competitive and often substantially outperforms the state of the art in this area.

It has been demonstrated that it is possible to use the SMC algorithms to estimate the model probabilities directly (SMC1), or through individual model evidence (SMC2), or pairwise relative evidence (SMC3). In addition, both SMC2 and SMC3 algorithms can be coupled with the path sampling estimator.

Among the three approaches, SMC1 is applicable to very general settings. It can provide a robust alternative to RJMCMC when inference on a countable collection of models is required (and could be readily combined with the approach of Jasra et al. (2008) at the expense of a little additional implementation effort). However, like all Monte Carlo methods involving between model moves, it can be difficult to design efficient algorithms in practice. The SMC3 algorithm is conceptually appealing. However, the existence of a suitable sequence of distributions between two posterior distributions may not be obvious.

The SMC2 algorithm, which only involves within-model simulation, is most straightforward to implement in many interesting problems. It has been shown to be exceedingly robust in many settings. As it depends largely upon a collection of within-model MCMC moves, any existing MCMC algorithms can be reused in the SMC2 framework. However, much less tuning is required because the algorithm is fundamentally less sensitive to the mixing of the Markov kernel and it is possible to implement effective adaptive strategies at little computational cost. With adaptive placement of the intermediate distributions and specification of the MCMC kernel proposals, it provides a robust and essentially automatic model comparison method.

Compared to the PMCMC algorithm, SMC2 has greater flexibility in the specification of distributions. Unlike PMCMC, where the number and placement of distributions can affect the mixing speed and hence performance considerably, increasing the number of distributions will always benefit a SMC sampler given the same number of particles. When coupled with a path sampling estimator, this leads to less bias and variance. Compared to its no-resampling variant, it has been shown that SMC samplers with resampling can reduce the variance of normalizing constant estimates considerably.

Even after three decades of intensive development, no Monte Carlo method can solve the Bayesian model comparison problem completely automatically without any manual tuning. However, SMC algorithms and the adaptive strategies demonstrated in this paper show that even for realistic, interesting problems, these samplers can provide good results with very minimal tuning and few design difficulties. For many applications, they could already be used as near automatic, robust solutions. For more challenge problems, the robustness of the algorithms can serve as solid foundation for specific algorithm designs.

A Proof of Proposition 1

We begin by making some identifications which allow the connection between the SMC sampler algorithm presented above and Feynman-Kac formula to be made explicit as the proof relies on approaches pioneered in Del Moral (2004). Throughout this appendix we write $\eta K(\cdot) = \int \eta(dx)K(x,\cdot)$ for any compatible measure η and Markov kernel K and $\eta(\varphi) = \int \eta(dx)\varphi(x)$ for any η-integrable function φ.

A Feynman-Kac formula describes the law of a Markov chain on $\{(E_t, \mathcal{E}_t)\}_{t\geq 0}$ (with initial distribution $\widehat{\eta}_0$ and transitions M_t) evolving in the presence of a (time-varying) potential (described by G_t) such that the marginal law of the t^{th} coordinate is:

$$\widehat{\eta}_t(A) = \frac{\int_{E_1\times\ldots\times E_{t-1}\times A} \widehat{\eta}_0(d\widetilde{x}_0)\prod_{i=1}^{t} M_i(\widetilde{x}_{i-1}, d\widetilde{x}_i)G(\widetilde{x}_i)}{\int_{E_1\times\ldots\times E_t} \widehat{\eta}_0(d\widetilde{x}'_0)\prod_{i=1}^{t} M_i(\widetilde{x}'_{i-1}, d\widetilde{x}'_i)G(\widetilde{x}'_i)}$$

for any measurable set A.

It is convenient to define the operator $\widehat{\Phi}_t(\eta)(\mathrm{d}\,\widetilde{x}_t) = G_t(\widetilde{x}_t)\eta M_t(\mathrm{d}\,\widetilde{x}_t)/\eta M_t(G_t)$ which allows us to write, recursively, $\widehat{\eta}_t = \widehat{\Phi}_t(\widehat{\eta}_{t-1})$ and to define the intermediate distributions $\eta_t = \widehat{\eta}_{t-1}M_t$ such that $\widehat{\eta}_t(\mathrm{d}\,\widetilde{x}_t) = G_t(\widetilde{x}_t)\eta_t(d\widetilde{x}_t)/\eta_t(G_t)$.

If $\mathcal{X}$ denotes the space upon which an SMC sampler with MCMC proposal K_t at time t and sequence of target distributions π_t operates, then we obtain π_t as the final coordinate

marginal of the Feynman-Kac distribution at time t if we identify $E_t = \mathcal{X}^t$, $M_t(\widetilde{x}_{t-1}, d\widetilde{x}_t) = \delta_{\widetilde{x}_{t-1}}(d\widetilde{x}_{t,1:t-1})K_t(\widetilde{x}_{t,t-1}, d\widetilde{x}_t)$ and $G_t(\widetilde{x}_t) = \pi_t(\widetilde{x}_{t,t-1})/\pi_{t-1}(\widetilde{x}_{t,t-1})$.

To provide symmetry between the simulation system and the ideal system which it targets, it is convenient to let $\widetilde{X}_t^i$ denote the extended sample corresponding to X_t^i at iteration t together with the full collection of values which its ancestors took during previous iterations (i.e., $\widetilde{X}_t^i$ corresponds to the particle system obtained by sampling according to M_t above rather than K_t at each iteration). It is then convenient to write the particle approximation at time t as

$$\widehat{\eta}_t^N(d\widetilde{x}_t) = \sum_{i=1}^{N} \frac{G_t(\widetilde{X}_t^i)}{\sum_{j=1}^{N} G_t(\widetilde{X}_t^j)} \delta_{\widetilde{X}_t^i}(d\widetilde{x}_t).$$

We refer the reader to Del Moral (2004) for further details of the connection between such particle systems and the Feynman-Kac formula.

In order to proceed, we prove the following more general result to which Proposition 1 is a direct corollary.

Proposition 2. *Under the regularity conditions given in (Del Moral, 2004, section 9.4, pp. 300–306), a weighted sum of integrals obtained from successive generations of the particle approximation of a Feynman-Kac flow $\{\widehat{\eta}_t\}_{t=0}^T$, with the application of multinomial resampling at every iteration, obeys a central limit theorem in the following sense, for a collection of finite weights $\beta_t \in \mathbb{R}$ and bounded measurable functions $\xi_t : E_t \to \mathbb{R}$ (where, in the historical process case described above it is required that $\xi_t(\widetilde{x}_t) = \xi_t(\widetilde{x}_{t,t})$):*

$$\lim_{N\to\infty} \sqrt{N} \sum_{t=0}^{T} \beta_t(\widehat{\eta}_t^N(\xi_t) - \widehat{\eta}_t(\xi_t)) \xrightarrow{D} \mathcal{N}(0, V_T(\xi_{0:T})) \tag{31}$$

where V_t, $0 \leq t \leq T$ is defined by the following recursion:

$$\begin{aligned} V_0(\xi_0) =& \beta_0 \int \widehat{\eta}_0(x_0)(\xi_0(x_0) - \eta_0(\xi_0))^2 dx_0 \\ V_t(\xi_{0:t}) =& V_{t-1}\left(\xi_{0:t-2}, \xi_{t-1} + \frac{\beta_t}{\beta_{t-1}} \frac{M_t(G_t[\xi_t - \widehat{\eta}_t(\xi_t)])}{\widehat{\eta}_{t-1}M_t(G_t)}\right) + \beta_t^2 \widehat{\eta}_t\left(\frac{G_t(\cdot)(\xi_t(\cdot) - \widehat{\eta}_t(\xi_t))^2}{\widehat{\eta}_t(G_t)}\right). \end{aligned} \tag{32}$$

The strategy of the proof is to decompose the error as that propagated forward from previous times and that due to sampling at the current time, just as in Del Moral (2004). First note that the term $\widehat{\eta}_t^N(\xi_t) - \widehat{\eta}_t(\xi_t)$ can be rewritten as

$$\widehat{\eta}_t^N(\xi_t) - \widehat{\eta}_t(\xi_t) = \widehat{\eta}_t^N(\xi_t) - \widehat{\Phi}_t(\widehat{\eta}_{t-1}^N)(\xi_t) + \widehat{\Phi}_t(\widehat{\eta}_{t-1}^N)(\xi_t) - \widehat{\eta}_t(\xi_t) \tag{33}$$

and the weighted sum,

$$T_t^N(\xi_{0:t}) = \sqrt{N} \sum_{j=0}^{t} \beta_j(\widehat{\eta}_j^N(\xi_j) - \widehat{\eta}_j(\xi_j)) \tag{34}$$

can therefore be written as

$$\begin{aligned} T_t^N(\xi_{0:t}) &= T_{t-1}^N(\xi_{0:t-1}) + \sqrt{N}\beta_t(\widehat{\eta}_t^N(\xi_t) - \widehat{\eta}_t(\xi_t)) \\ &= \bar{T}_t^N(\xi_{0:t}) + \chi_t^N(\xi_t) \end{aligned} \tag{35}$$

where

$$\bar{T}_t^N(\xi_{0:t}) = T_{t-1}^N(\xi_{0:t-1}) + \sqrt{N}\beta_t(\widehat{\Phi}_t(\widehat{\eta}_{t-1}^N)(\xi_t) - \widehat{\eta}_t(\xi_t)) \tag{36}$$

$$\chi_t^N(\xi_t) = \sqrt{N}\beta_t(\widehat{\eta}_t^N(\xi_t) - \widehat{\Phi}_t(\eta_{t-1}^N)(\xi_t)) \tag{37}$$

Lemma 1 shows that error propagation leads to controlled normal errors; Lemma 2 shows that the act of sampling during each iteration also produces a normally-distributed error and Lemma 3 shows that these two normal errors can be combined leading by induction to Proposition 2.

Lemma 1. *Under the conditions of Proposition 2, if* $T_{t-1}^N(\xi_{0:t-1}) \xrightarrow{D} \mathcal{N}(0, V_{t-1}(\xi_{0:t-1}))$*, then*

$$\bar{T}_t^N(\xi_{0:t}) \xrightarrow{D} \mathcal{N}(0, \bar{V}_t(\xi_{0:t})) \tag{38}$$

where

$$\bar{V}_t = V_{t-1}\left(\xi_{0:t-2}, \xi_{t-1} + \frac{\beta_t}{\beta_{t-1}} \frac{M_t(G_t[\xi_t - \widehat{\eta}_t(\xi_t)])}{\widehat{\eta}_{t-1} M_t(G_t)}\right). \tag{39}$$

Proof. We begin by re-expressing the difference of interest in a more convenient form:

$$\begin{aligned}\widehat{\Phi}(\widehat{\eta}_{t-1}^N)(\xi_t) - \widehat{\eta}_t(\xi_t) &= \frac{1}{\widehat{\eta}_{t-1}^N M_t(G_t)}\{\widehat{\eta}_{t-1}^N M_t(G_t\xi_t) - \widehat{\eta}_{t-1}^N M_t(G_t)\widehat{\eta}_t(\xi_t)\} \\ &= \frac{1}{\widehat{\eta}_{t-1}^N M_t(G_t)}\{(\widehat{\eta}_{t-1}^N - \widehat{\eta}_{t-1}) M_t(G_t[\xi_t - \widehat{\eta}_t(\xi_t)])\}\end{aligned} \tag{40}$$

where the final equality is a simple consequence of the fact that for any integrable test function φ:

$$\widehat{\eta}_{t-1} M_t(G_t\varphi) = \eta_t(G_t)\widehat{\eta}_t(\varphi) \Rightarrow \widehat{\eta}_{t-1} M_t(G_t[\xi_t - \widehat{\eta}_t(\xi_t)]) = \eta_t(G_t)\underbrace{\widehat{\eta}_t(\xi_t - \widehat{\eta}_t(\xi_t))}_{=0}.$$

Substituting this representation into Equation (36),

$$\begin{aligned}\bar{T}_t^N(\xi_{0:t}) &= T_{t-1}^N(\xi_{0:t-1}) + \frac{\sqrt{N}\beta_t}{\widehat{\eta}_{t-1}^N M_t(G_t)}\{(\widehat{\eta}_{t-1}^N - \widehat{\eta}_{t-1}) M_t(G_t[\xi_t - \widehat{\eta}_t(\xi_t)])\} \\ &= T_{t-1}^N\left(\xi_{0:t-2}, \xi_{t-1} + \frac{\beta_t}{\beta_{t-1}} \frac{M_t(G_t[\xi_t - \widehat{\eta}_t(\xi_t)])}{\widehat{\eta}_{t-1}^N M_t(G_t)}\right)\end{aligned} \tag{41}$$

The proof is completed by using the result (Del Moral, 2004, cf. Sec. 7.4.3), that if G_t is essentially bounded below then,

$$\frac{1}{\widehat{\eta}_{t-1}^N M_t(G_t)} \xrightarrow{p} \frac{1}{\widehat{\eta}_{t-1} M_t(G_t)}$$

together with the induction hypothesis. □

Lemma 2. *Under the conditions of Proposition 2,*

$$\lim_{t\to\infty} \chi_t^N(\xi_t) \xrightarrow{D} \mathcal{N}(0, \widehat{V}_t(\xi_t)) \tag{42}$$

where

$$\widehat{V}_t(\xi_t) = \beta_t^2 \widehat{\eta}_t((\xi_t - \widehat{\eta}_t(\xi_t))^2) \tag{43}$$

Proof. Consider first the particle system before reweighting with the potential function G_t:

$$\sqrt{N}\beta_t \sum_{j=1}^N \frac{\xi_t(\widetilde{X}_t^{(j)}) - \widehat{\eta}_{t-1}^N M_t(\xi_t)}{N} = \sum_{j=1}^N U_{t,j}^N \tag{44}$$

where $U_{t,j}^N = \frac{\beta_t}{\sqrt{N}}\{\xi_t(\widetilde{X}_t^{(j)}) - \widehat{\eta}_{t-1}^N M_t(\xi_t)\}$. Define, recursively, the σ-algebras $\mathcal{H}_t^N = \mathcal{H}_{t-1}^N \vee \sigma(\{\widetilde{X}_t^{(j)}\}_{j=1}^N)$, $\mathcal{H}_{t-1} = \sigma\left(\cup_{N=0}^\infty \mathcal{H}_{t-1}^N\right)$ and the increasing (in j) sequence of σ-algebras $\mathcal{H}_{t,j}^N = \mathcal{H}_{t-1} \vee \sigma(\{\widetilde{X}_t^{(l)}\}_{l=1}^j)$. It is clear that

$$\mathbb{E}[U_{t,j}^N | \mathcal{H}_{t,j-1}^N] = \mathbb{E}[U_{t,j}^N | \mathcal{H}_{t-1}] = 0 \tag{45}$$

and so the sequence $U_{t,j}^N$, $j = 1, \ldots, N$ comprises a collection of $\mathcal{H}_{t,j}^N$-martingale increments. Further it can be verified that these martingale increments are square integrable,

$$\begin{aligned}\mathbb{E}[(U_{t,j}^N)^2 | \mathcal{H}_{t,j-1}^N] &= \mathbb{E}[(U_{t,j}^N)^2 | \mathcal{H}_{t-1}] \\ &= \frac{\beta_t^2}{N}\{\widehat{\eta}_{t-1}^N M_t(\xi_t^2) - [\widehat{\eta}_{t-1}^N)(\xi_t)]^2\} < c_t \frac{\beta_t^2}{N}\end{aligned}$$

where $c_t < \infty$ exists by the boundedness of ξ_t. The conditional Linderberg condition is also clearly satisfied. That is, for any $0 < u \leq 1$ and $\varepsilon > 0$,

$$\lim_{N\to\infty} \sum_{j=1}^{\lfloor Nu \rfloor} \mathbb{E}[(U_{t,j}^N)^2 \mathbb{I}_{(\varepsilon,\infty)}(|U_{t,j}^N|)|\mathcal{H}_{t,j}^N] \xrightarrow{P} 0.$$

Thus we have

$$\begin{aligned}\sum_{j=1}^{N} \mathbb{E}[(U_{t,j}^N)^2|\mathcal{H}_{t,j-1}^N] &= \frac{\beta_t^2}{N} \sum_{j=1}^{N} \{\widehat{\eta}_{t-1}^N M_t(\xi_t^2) - [\widehat{\eta}_{t-1}^N M_t(\xi_t)]^2\} \\ &= \beta_t^2 \{\widehat{\eta}_{t-1}^N M_t(\xi_t^2) - [\widehat{\eta}_{t-1}^N M_t(\xi_t)]^2\}\end{aligned}$$

and we can invoke the martingale central limit theorem (Shiryaev, 1995, pp. 543),

$$\lim_{N\to\infty} \chi_t^N(\xi_t) \xrightarrow{D} \mathcal{N}(0, \check{V}_t(\xi_t)) \tag{46}$$

where the asymptotic variance, $\check{V}_t(\xi_t)$, may be written as the limit of the sequence defined by

$$\check{V}_t^N(\xi_t) = \beta_t^2 \{\widehat{\eta}_{t-1}^N M_t(\xi_t^2) - [\widehat{\eta}_{t-1}^N M_t(\xi_t)]^2\} \tag{47}$$

and as (again, see (Del Moral, 2004, Section 7.4))

$$\check{V}_t^N(\xi_t) \xrightarrow{P} \beta_t^2 \{\widehat{\eta}_{t-1} M_t(\xi_t^2) - [\widehat{\eta}_{t-1} M_t(\xi_t)]^2\}$$

the proof is completed using Slutzky's lemma and applying Chopin (2004, Lemma A2) which yields that:

$$\lim_{N\to\infty} \mathcal{X}_t^N \xrightarrow{D} \mathcal{N}(0, \widehat{V}_t(\xi_t))$$

with

$$\widehat{V}_t(\xi_t) = \check{V}_t \left(\frac{G_t(\cdot)}{\widehat{\eta}_{t-1} M_t(G_t)} (\xi_t(\cdot) - \widehat{\eta}_t(\xi_t)) \right) = \beta_t^2 \widehat{\eta}_t \left(\frac{G_t(\cdot)}{\widehat{\eta}_{t-1} M_t(G_t)} (\xi_t(\cdot) - \widehat{\eta}_t(\xi_t)) \right)$$

□

Lemma 3. *Under conditions of Proposition 2, and the inductive assumption of Lemma 1, $T_t^N(\xi_{0:t})$ is asymptotically normal with variance stated as in Proposition 2.*

Proof. Consider the characteristic function,

$$\begin{aligned}\varphi(T_t^N(\xi_{0:t}))(s) &= \mathbb{E}[\exp(isT_t^N(\xi_{0:t}))] \\ &= \mathbb{E}[\exp(is\bar{T}_t^N(\xi_{0:t}))\exp(is\chi_t^N(\xi_t))] \\ &= \mathbb{E}[\exp(is\bar{T}_t^N(\xi_{0:t}))\mathbb{E}[\exp(is\chi_t^N(\xi_t))|\mathcal{H}_{t-1}^N]] \\ &= \mathbb{E}[(A_t - \exp(-s^2\widehat{V}_t(\xi_t)/2))B_t] + \exp(-s^2\widehat{V}_t(\xi_t)/2)\mathbb{E}[B_t]\end{aligned}$$

where $A_t = \mathbb{E}[\exp(is\chi_t^N(\xi_t))|\mathcal{H}_{t-1}^N]$ and $B_t = \exp(is\bar{T}^N(\xi_{0:t}))$. The first term can easily be shown to be converge a.s. to zero as $N \to \infty$ by the asymptotic normality of ξ_t^N and the conditional independence of the particles at iteration t given $\mathcal{H}_{t-1}^N$. The second term is the product of two Gaussian characteristic functions and thus we have that T_t^N also follows a Gaussian distribution (see the argument of Künsch (2005) shown in more detail in Lemma 10 in Johansen et al. (2006), which uses essentially the same argument, for details). □

Using Lemma 1 to 3, the proof of Proposition 2 follows by mathematical induction and a trivial base case (the first iteration is simple importance sampling).

References

Andrieu, C. and E. Moulines (2006, August). On the ergodicity properties of some adaptive MCMC algorithms. *The Annals of Applied Probability 16*(3), 1462–1505.

Atchadé, Y. F., G. O. Roberts, and J. S. Rosenthal (2010). Towards optimal scaling of Metropolis-coupled Markov chain Monte Carlo. *Statistics and Computing 21*(4), 555–568.

Bartolucci, F., L. Scaccia, and A. Mira (2006). Efficient Bayes factor estimation from the reversible jump output. *Biometrika 93*(1), 41–52.

Bernardo, J. M. and A. F. M. Smith (1994). *Bayesian Theory*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc.

Calderhead, B. and M. Girolami (2009). Estimating Bayes factors via thermodynamic integration and population MCMC. *Computational Statistics & Data Analysis 53*(12), 4028–4045.

Cappé, O., S. J. Godsill, and E. Moulines (2007). An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE 95*(5), 899–924.

Cappé, O., C. P. Robert, and T. Ryden (2003). Reversible jump, birth-and-death and more general continuous time Markov chain Monte Carlo samplers. *Journal of Royal Statistical Society B 65*(3), 679–700.

Carlin, B. P. and S. Chib (1995). Bayesian model choice via Markov chain Monte Carlo methods. *Journal of Royal Statistical Society B 57*(3), 473–484.

Chib, S. (1995). Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association 90*(432), 1313–1321.

Chib, S. and I. Jeliazkov (2001). Marginal likelihood from the Metropolis-Hastings output. *Journal of the American Statistical Association 96*(453), 270–281.

Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika 89*(3), 539–552.

Chopin, N. (2004). Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *The Annals of Statistics 32*(6), 2385–2411.

Del Moral, P. (1996). Nonlinear filtering: interacting particle solution. *Markov Processes and Related Fields 4*(2), 555–580.

Del Moral, P. (2004). *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. New York: Springer-Verlag.

Del Moral, P., A. Doucet, and A. Jasra (2006a). Sequential Monte Carlo methods for Bayesian computation. In *Bayesian Statistics 8*. Oxford University Press.

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

Del Moral, P., A. Doucet, and A. Jasra (2012). On adaptive resampling strategies for sequential Monte Carlo methods. *Bernoulli 18*(1), 252–278.

Didelot, X., R. G. Everitt, A. M. Johansen, and D. J. Lawson (2011). Likelihood-free estimation of model evidence. *Bayesian Analysis 6*(1), 49–76.

Doucet, A. and A. M. Johansen (2011). A tutorial on particle filtering and smoothing: Fifteen years later. In *The Oxford Handbook of Non-linear Filtering*. Oxford University Press.

Fan, Y., D. Leslie, and M. P. Wand (2008). Generalised linear mixed model analysis via sequential Monte Carlo sampling. *Electronic Journal of Statistics 2*, 916–938.

Fearnhead, P. and B. Taylor (2010). An adaptive sequential Monte Carlo sampler. Mathematics Preprint 1005.1193v2, ArXiv.

Friel, N., M. Hurn, and J. Wyse (2012, September). Improving power posterior estimation of statistical evidence. *ArXiv 1209.3198*, 1–24.

Gelfand, A. E. and D. K. Dey (1994). Bayesian model choice: Asymptotics and exact calculations. *Journal of Royal Statistical Society B 56*(3), 501–514.

Gelman, A. and X.-L. Meng (1998). Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science 13*(2), 163–185.

Geyer, C. (1991). Monte Carlo maximum likelihood. In Keramigas (Ed.), *Proceedings of Computing Science and Statistics: The 23rd Symposium on the Interface*, Fairfax, pp. 156–161. Interface Foundation.

Godsill, S. J. (2001). On the relationship between Markov chain Monte Carlo for model uncerntainty. *Journal of Computational and Graphical Statistics 10*(2), 230–248.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika 82*(4), 711–732.

Grelaud, A., C. P. Robert, J.-M. Marin, F. Rodolphe, and J.-F. Taly (2009). ABC likelihood-free methods for model choice in Gibbs random fields. *Bayesian Analysis 4*(2), 317–336.

Grenander, U. and M. I. Miller (1994). Representations of knowledge in complex systems. *Journal of Royal Statistical Society B 56*(4), 549–603.

Gunn, R. N., S. R. Gunn, F. E. Turkheimer, J. A. D. Aston, and V. J. Cunningham (2002). Positron emission tomography compartmental models: A basis pursuit strategy for kinetic modeling. *Journal of Cerebral Blood Flow & Metabolism 22*(12), 1425–1439.

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

Jasra, A., D. A. Stephens, A. Doucet, and T. Tsagaris (2010, December). Inference for Lévy-Driven Stochastic Volatility Models via Adaptive Sequential Monte Carlo. *Scandinavian Journal of Statistics 38*(1), 1–22.

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

Jasra, A., D. A. Stephens, and C. C. Holmes (2007b). Population-based reversible jump Markov chain Monte Carlo. *Biometrika 94*(4), 787–807.

Jiang, C.-R., J. A. D. Aston, and J.-L. Wang (2009). Smoothing dynamic positron emission tomography time courses using functional principal components. *NeuroImage 47*(1), 184–193.

Johansen, A. M., P. Del Moral, and A. Doucet (2006). Sequential Monte Carlo samplers for rare events. In *Proceedings of the 6th International Workshop on Rare Event Simulation*, pp. 256–267.

Johansen, A. M., A. Doucet, and M. Davy (2008). Particle methods for maximum likelihood estimation in latent variable models. *Statistics and Computing 18*(1), 47–57.

Johansen, A. M., S. S. Singh, A. Doucet, and B.-N. Vo (2006). Convergence of the SMC implementation of the PHD filter. *Methodology and Computing in Applied Probability 8*(2), 265–291.

Kong, A., J. S. Liu, and W. H. Wong (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association 89*(425), 278–288.

Künsch, H. R. (2005). Recursive Monte Carlo filters: Algorithms and theoretical analysis. *Annals of Statistics 33*(5), 1983–2021.

Lee, A., C. Yau, M. B. Giles, A. Doucet, and C. 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.

Liang, F. and W. H. Wong (2001, June). Real-parameter evolutionary Monte Carlo with applications to Bayesian mixture models. *Journal of the American Statistical Association 96*(454), 653–666.

Marinari, E. and G. Parisi (1992). Simulated tempering: a new Monte Carlo scheme. *Europhysics Letters 19*(6), 451–458.

Neal, R. M. (1994). Discussion of "Approximate Bayesian inference with the weighted likelihood bootstrap" by Newton and Raftery. *Journal of the Royal Statistical Society, Series B 56*(1), 41–42.

Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing 11*(2), 125–139.

Newton, M. A. and A. E. Raftery (1994). Approximate Bayesian inference with the weighted likelihood bootstrap. *Journal of Royal Statistical Society B 56*(1), 3–48.

Peters, G., K. Hayes, and G. Hossack (2010). Ecological non-linear state space model selection via adaptive particle Markov chain Monte Carlo (AdPMCMC). Mathematics Preprint 1005.2238, ArXiv.

Peters, G. W. (2005). Topics in sequential Monte Carlo samplers. Master's thesis, University of Cambridge, Department of Engineering.

Raftery, A. E., M. A. Newton, J. M. Satagopan, and P. N. Krivitsky (2006, November). Estimating the Integrated Likelihood via Posterior Simulation Using the Harmonic Mean Identity. In *Bayesian Statistics 8*, pp. 1–45. Oxford University Press.

Richardson, S. and P. J. Green (1997). On Bayesian analysis of mixtures with an unknown number of components. *Journal of Royal Statistical Society B 59*(4), 731–792.

Robert, C. P. (2007). *The Bayesian Choice: From Decision-theoretic Foundations to Computational Implementation* (2nd ed.). New York: Springer.

Robert, C. P., J.-M. Marin, and N. S. Pillai (2011). Why approximate Bayesian computational (ABC) methods cannot handle model choice problems. *Proceedings of the National Academy of Sciences (USA) 108*(37), 15112–15117.

Roberts, G. O. and J. S. Rosenthal (2001). Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science 16*(4), 351–367.

Roberts, G. O. and R. L. Tweedie (1996). Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli 2*(4), 341–363.

Rousset, M. and G. Stoltz (2006). Equilibrium sampling from nonequilibrium dynamics. *Journal of Statistical Physics 123*(6), 1251–1272.

Schäfer, C. and N. Chopin (2013). Sequential Monte Carlo on large binary sampling spaces. *Statistics and Computing 23*(2), 163–184. In press.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics 6*(2), 461–464.

Shiryaev, A. N. (1995). *Probability*. Graduate Texts in Mathematics. New York: Springer-Verlag.

Stephens, M. (2000). Bayesian analysis of mixture models with an unkown number of components – An alternative to reversible jump methods. *The Annals of Statistics 28*(1), 40–74.

Vyshemirsky, V. and M. A. Girolami (2008). Bayesian ranking of biochemical system models. *Bioinformatics 24*(6), 833–839.

Whiteley, N. (2013). Sequential monte carlo samplers: error bounds and insensitivity to initial conditions. *Stochastic Analysis and Applications 30*(5), 774–798. In press.

Zhou, Y., J. A. D. Aston, and A. M. Johansen (2013). Bayesian model comparison for compartmental models with applications in positron emission tomography. *Journal of Applied Statistics*. In press.

Zhou, Y., A. M. Johansen, and J. A. D. Aston (2012). Bayesian model selection via path-sampling sequential Monte Carlo. In *Proceedings of IEEE Statistical Signal Procesing Workshop*, Ann Arbor, Michigan, USA, pp. 245–248.