Approximate Bayesian computation for finite mixture models

Umberto Simola\textsuperscript{a}, Jessi Cisewski-Kehe\textsuperscript{b} and Robert L. Wolpert\textsuperscript{c}

\textsuperscript{a}Department of Mathematics and Statistics, University of Helsinki, Helsinki, Finland; \textsuperscript{b}Department of Statistics, Yale University, New Haven, CT, USA; \textsuperscript{c}Department of Statistical Science, Duke University, Durham, NC, USA

\textbf{ABSTRACT}

Finite mixture models are used in statistics and other disciplines, but inference for mixture models is challenging due, in part, to the multimodality of the likelihood function and the so-called label switching problem. We propose extensions of the Approximate Bayesian Computation–Population Monte Carlo (ABC–PMC) algorithm as an alternative framework for inference on finite mixture models. There are several decisions to make when implementing an ABC–PMC algorithm for finite mixture models, including the selection of the kernels used for moving the particles through the iterations, how to address the label switching problem and the choice of informative summary statistics. Examples are presented to demonstrate the performance of the proposed ABC–PMC algorithm for mixture modelling. The performance of the proposed method is evaluated in a simulation study and for the popular recessional velocity galaxy data.

\textbf{ARTICLE HISTORY}

Received 26 May 2020
Accepted 25 October 2020

\textbf{KEYWORDS}

Approximate Bayesian computation; label switching; finite mixture models; perturbation kernels; summary statistics

1. Introduction

Mixture models have been used in statistics since the late 1800s when Karl Pearson introduced them in an analysis of crab morphometry \cite{1,2}. Subsequently mixture modelling has grown in popularity in the statistical community as a powerful framework for modelling clustered data; McLachlan and Peel\cite{3} provide a general overview of mixture modelling, and Marin et al.\cite{4} or Schnatter\cite{5} provide a Bayesian perspective. In recent decades, mixture models have become routinely applied in various applications\cite{6–12}. One reason for the general success of mixture models is the ability to specify the number of possibly different component distributions, allowing for flexibility in describing complex systems\cite{4}.

The general definition for a finite mixture model with fixed integer $K > 1$ components is

$$\sum_{i=1}^{K} f_i : p_i(y | \theta_i), \quad (1)$$

\textbf{CONTACT} Umberto Simola \copyright umberto.simola@helsinki.fi

© 2020 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group
This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives License (http://creativecommons.org/licenses/by-nc-nd/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited, and is not altered, transformed, or built upon in any way.
with mixture weights $f_i \in (0,1)$ such that $\sum_{i=1}^{K} f_i = 1$ and $p_i(y \mid \theta_i)$ are the component distributions of the mixture, often parametrically specified with a vector of the unknown parameters, $\theta_i$.

Finite mixture models present computational and methodological challenges due, at least in part, to the complex and possibly multimodal likelihood function along with the invariance under permutation of the component indices. The Expectation–Maximization (EM) algorithm [13] provides a method for numerically obtaining the maximum likelihood estimates, although the possible multimodality of the likelihood function makes finding the global maximum challenging [4]. Extensions of the EM algorithm have been proposed for improving its speed of convergence and avoiding local optima [14,15].

Bayesian methodology and applications for mixture modelling have developed in the last decades [4,16–19]. Bayesian inference for mixture models often relies on Markov Chain Monte–Carlo (MCMC) which can lead to the so-called label switching problem [20–23], because the likelihood function is invariant to the re-labelling of the mixture components. Additionally the resulting posterior distribution is multimodal and asymmetric, which makes summarizing the posterior distribution using common statistics such as the posterior mean or the Highest Posterior Density (HPD) interval unhelpful [19,21,24].

A different framework for inference that can be considered for addressing the issues related to the use of mixture models is Approximate Bayesian Computation (ABC), whose basic algorithm was first presented conceptually by Rubin [25] and computationally by Tavaré et al. [26] and Pritchard et al. [27]. ABC is often used in situations where the likelihood function is complex or not available, but simulation of data through a forward model (FM) is possible. The FM is a simulation model that can take parameters as inputs and returns as output simulated data. With mixture models, though the likelihood function can be available, working with the likelihood is not always straightforward. Though ABC has its own challenges, we propose an ABC algorithm for estimating the posterior distributions of the parameters providing an alternative to the MCMC algorithms.

Assuming $\theta$ is the possibly multidimensional inferential target, the basic ABC algorithm consists of the following four steps [26,27]: (i) sample from the prior distribution, $\theta_{\text{prop}} \sim \pi(\theta)$; (ii) produce a generated sample of the data by using $\theta_{\text{prop}}$ in the FM, $y_{\text{prop}} \sim \text{FM}(Y \mid \theta_{\text{prop}})$; (iii) compare the real data, $y_{\text{obs}}$, with the generated sample, $y_{\text{prop}}$, using a distance function, $\rho(\cdot, \cdot)$, letting $d = \rho(s(y_{\text{obs}}), s(y_{\text{prop}}))$ where $s(\cdot)$ is some (possibly multidimensional) summary statistic of the data; (iv) if the distance, $d$, is smaller than a fixed tolerance $\epsilon$ then $\theta_{\text{prop}}$ is retained, otherwise it is discarded. Repeat until the desired particle sample size $N$ is achieved.

In order to improve the computational efficiency of the basic ABC algorithm, there have been many developments [28–37]. In this work, we extend a sequential version of ABC, the ABC–PMC of Beaumont et al. [30], which is introduced in the next Section. Interestingly, the original PMC sampler [38] was introduced for solving problems related to the multimodality of the likelihood function for settings in which mixture models were used. This suggests that the ABC–PMC version of the ABC algorithm may be more suitable for mixture models compared to other available algorithms, such as the Sequential Monte Carlo version (ABC–SMC) by [32]. ABC may be considered preferable over MCMC if the likelihood function is intractable. However, in the finite mixture model setting, the likelihood function is tractable, but there are computational challenges due to the multimodality of
the likelihood function as well as the label switching problem that can lead to issues for MCMC [19,38].

In any ABC analysis the definition of both suitable summary statistics, \( s(\cdot) \), and a distance function, \( \rho \), for comparing the real data \( y_{\text{obs}} \) to the generated sample \( y_{\text{prop}} \) is needed and is crucial for getting useful inference results [39]. The definition of summary statistics is necessary because ABC suffers of the curse of dimensionality and using the entire dataset is not computationally feasible [40–43]. In the proposed method, we generally use the Hellinger distance for comparing the real data, \( y_{\text{obs}} \), to the generated sample, \( y_{\text{prop}} \). While the proposed method is valid for \( y_{\text{obs}} \in \mathbb{R}^{\text{dim}} \), for illustration purposes, we focus most of the discussion on the one dimensional case where \( \text{dim} = 1 \).

The paper is organized as follows. Section 2 begins with background on the ABC–PMC algorithm by Beaumont et al. [30] and then details our proposed extensions for the finite mixture model setting. Sections 3 and 4 are dedicated, respectively, to a simulation study and a popular real data example using recessional velocities of galaxy data. Concluding remarks are presented in Section 5.

2. Methodology

Sequential versions of the basic ABC algorithm have been proposed to improve the computational efficiency of the algorithm. This is typically accomplished by considering a sequence of decreasing tolerances, \( \epsilon_{1:T} = (\epsilon_1, \ldots, \epsilon_T) \) such that \( \epsilon_1 \geq \epsilon_2 \geq \cdots \geq \epsilon_T \), where \( T \) is the total number of iterations, or time steps, of the algorithm. Rather than sampling from the prior at each iteration after the initial step, improved proposals are drawn from the ABC posterior sample of the previous step. The ABC–PMC algorithm of Beaumont et al. [30] relies on ideas of importance sampling in order to improve the efficiency of the algorithm by constructing a series of intermediate proposal distributions using the sequential ABC posteriors. The first iteration of the ABC–PMC algorithm uses tolerance \( \epsilon_1 \) and draws proposals from the specified prior distributions; from the second iteration through the \( T^{\text{th}} \) step, proposals are selected from the previous iteration’s ABC posterior and then moved according to some perturbation kernel (e.g. a Gaussian kernel). Since the proposals are drawn from the previous iteration’s ABC posterior and moved according to a kernel, rather than being sampled directly from the prior distributions, importance weights are used. The derivation of the importance weight for particle \( J = 1, \ldots, N \) in iteration \( t \), where \( N \) is the desired particle sample size, is discussed in Beaumont et al. [30] and leads to the following importance weights:

\[
W_t^{(J)} \propto \pi \left( \theta_t^{(J)} \right) / \sum_{L=1}^{N} W_{t-1}^{(L)} \phi \left( \theta_t^{(J)} \mid \theta_{t-1}^{(L)}, \tau^2_{\theta, t-1} \right),
\]

where \( \pi (\cdot) \) is the prior distribution, \( \phi (\cdot \mid a, b) \) is a Gaussian perturbation kernel with mean \( a \) and variance \( b \), where \( \tau^2_{\theta, t-1} \) is twice the (weighted) sample variance of the particles for \( \theta \) at iteration \( t-1 \) (as recommended [30]). The importance weights are then standardized so that \( \sum_{J=1}^{N} W_t^{(J)} = 1 \). Other perturbation kernels can be used and the importance weights would need to be updated accordingly. Further details on the original ABC–PMC algorithm can be found in Beaumont et al. [30], though there are several elements that require additional discussion. In particular, an implementation of ABC–PMC requires the
user to select the tolerance sequence $\epsilon_{1:T}$, the perturbation kernels, the summary statistic(s) and the distance function(s).

First, to select the sequence $\epsilon_{1:T}$ we implement an adaptive approach based on the accepted distances from the previous time step. That is, $\epsilon_t$ is set to a particular quantile of the accepted distances from time step $t-1$ [32,44–48]. Selecting too high of a quantile at which to shrink the tolerance could result in a decrease in computational efficiency because more iterations would be needed to shrink the tolerance to a small enough value. On the other hand, selecting a quantile that is too low can also contribute to computational inefficiencies because more draws from the proposal are needed within each iteration to find datasets that achieve the small tolerance. The rule used to shrink the tolerance is important as imprudent selection can lead to the ABC–PMC algorithm getting stuck in local modes [49,50]. This latter potential issue is particularly crucial when working with mixture models, since the multimodality of the likelihood function is a well-known problem that needs to be addressed in order to produce reliable statistical inference results. The decreasing tolerance sequence’s impact on the achievement of the global maximum is addressed in two ways in the proposed algorithm. First, we initialize the tolerance sequence by oversampling in the first iteration of the ABC–PMC algorithm [47]. Let $N$ be the desired number of particles used to approximate the posterior distribution, then the initial tolerance, $\epsilon_1$, can be adaptively selected by sampling $N_{\text{init}} = lN$ draws from the prior, for some $l \in \mathbb{Z}^+$. Then the $N$ particles of the $lN$ total particles with the smallest distances are retained, and $\epsilon_1 = \max(d_1^{(1)} , \ldots , d_1^{(N)})$ where $d_1^{(1)} , \ldots , d_1^{(N)}$ are the $N$ smallest distances of the $lN$ particles sampled. For all examples considered in this work, we use $l = 5$. Using this adaptive initialization, the parameter space is better sampled than if we only considered $N$ draws from the prior distribution. Second, we use lower quantiles for the first several iterations of the ABC–PMC algorithm per the suggestion in Silk et al. [49] and Simola et al. [50], which can help the algorithm avoid local modes.

One of the challenges in using ABC–PMC for mixture models is related to selecting an appropriate perturbation kernel for the mixture weights because the individual weights must be between 0 and 1, and the weights must sum to 1. Hence, the usual Gaussian perturbation kernel is not a viable option because this kernel can lead to proposed mixture weights that are not within the noted constraints. An additional challenge in using ABC–PMC for mixture models is due to the label switching problem, and for reasons discussed below, this has to be addressed at the end of each iteration. Finally, the selection of an appropriate and informative summary statistic for multimodal data requires a summary that can capture the overall shape of the distribution of the data.

For these reasons, the original version of the ABC–PMC algorithm is not directly able to accommodate mixture models. Our proposed extensions are discussed in more detail later, but can be summarized as follows. In Algorithm 1, we adapt the ABC–PMC algorithm to work with finite Gaussian mixture models. In particular, we propose a procedure for resampling the mixture weights in a way that capitalizes on distributional properties of the prior distribution for the mixture weights (i.e. the Dirichlet distribution), which is presented in Algorithm 2. The mixture weights could be resampled using other methods, such as resampling the particles according to a Gaussian perturbation kernel, but the resampled weights would need to be adjusted in order to preserve properties of weights such as being between or equal to 0 and 1 and summing to 1. Additionally, in order to address
the label switching problem, we designed a procedure that is executed at the end of each iteration of the ABC–PMC algorithm, which is presented in Algorithm 3. An overview of different available algorithms designed for addressing the label switching problem and more details on Algorithm 3 are presented in Section 2.3. If not properly addressed, the label switching problem can hinder the algorithm’s capability of finding the true posterior distribution.

2.1. Finite Gaussian mixture models

A common choice for the $p_i(\cdot | \cdot)$ of Equation (1) is the Gaussian distribution. This particular class of models, called Gaussian Mixture Models (GMMs), is flexible and widely used in various applications (e.g. [10,11,24]). Maintaining the notation of Equation (1), a GMM is defined as

$$
\sum_{i=1}^{K} f_i \cdot \phi(y \mid \theta_i),
$$

where $\phi$ is the density function of the Gaussian distribution, $\theta_i = (\mu_i, \sigma_i^2)$ is the vector of unknown means and variances for each of the $K$ groups with the unknown parameters $\mu = (\mu_1, \ldots, \mu_K) \in \mathbb{R}^K$, $\sigma^2 = (\sigma_1^2, \ldots, \sigma_K^2) \in \mathbb{R}_+^K$, and the mixture weights, $f = (f_1, \ldots, f_K)$. Note that the $f = (f_1, \ldots, f_K)$ lie in the unit simplex, $\Delta^{K-1} \equiv \{x \in \mathbb{R}^K_+ : \sum_j x_j = 1\}$ inside the unit cube $[0,1]^K$.

A common choice of the prior distribution for $f = (f_1, \ldots, f_K)$ is the Dirichlet distribution of order $K$ with hyperparameter $\delta = (\delta_1, \ldots, \delta_K)$, where often $\delta \equiv (1, \ldots, 1)$, as proposed by Wasserman [51]. Another common choice has been proposed by Rousseau and Mengersen [52], who defined the hyperparameter $\delta \equiv (1/2, \ldots, 1/2)$; in this way the prior is marginally a Jeffreys prior distribution. The priors for the mean and the variance of the GMM from Equation (3) can be defined as follows:

$$
\mu_i \sim \phi(\mu \mid \xi, \kappa), \quad \sigma_i^2 \sim \text{Inv - Gamma}(\alpha, \beta),
$$

with mean $\xi$, variance $\kappa$, and shape parameter $\alpha$ and rate parameter $\beta$. There are several methods for selecting the hyperparameters, $\eta = (\xi, \kappa, \alpha, \beta)$, such as the Empirical Bayes approach [53] and the ‘weakly informative principle’ [7]. Both of these options are considered in examples later so as to be consistent with the original analysis of the example.

Our proposed algorithm is specified in Algorithm 1 for the model defined in Equation (3), and the details of important steps are presented next.

2.2. Perturbation kernel functions

One of the advantages of ABC–PMC over the basic ABC algorithm is that, starting from the second iteration, rather than drawing proposals from the prior distribution, proposed particles are drawn from the previous step’s ABC posterior according to their importance weights. Then, instead of using the actual proposed value that was drawn, it is perturbed according to some kernel. Constructing computationally efficient perturbation kernels is crucial to improve the computational performance of the ABC algorithm [54]. There are a number of kernel functions, $K(\cdot | \cdot)$, available for perturbing the proposed particles. Beaumont et al. [30] suggest a Gaussian kernel centred on the selected particle from the previous
Algorithm 1 ABC–PMC for Finite Gaussian Mixture Model

Select the number of components $K$
Select the desired number of particles $N$
Select the desired number of particles coming from the prior $N_{\text{init}}$, $N_{\text{init}} > N$, for $t = 1$
if $t = 1$ then
  for $j = 1, \ldots, N_{\text{init}}$ do
    Propose $\mu^{(j)}_{1} = \{\mu_{1,1}, \ldots, \mu_{1,K}\}^{(j)}$ by drawing from prior $\mu^{*}_{k} \sim \pi(\mu)$, $k = 1, \ldots, K$
    Propose $\sigma^{2(j)}_{1} = \{\sigma^{2}_{1,1}, \ldots, \sigma^{2}_{1,K}\}^{(j)}$ by drawing from prior $\sigma^{2*}_{k} \sim \pi(\sigma^2)$, $k = 1, \ldots, K$
    Propose $f^{(j)}_{1} = \{f_{1,1}, \ldots, f_{1,K}\}^{(j)}$ by drawing from prior $f^{*}_{k} \sim \pi(f)$, $k = 1, \ldots, K$
  end for
  Select the desired number of particles $N$
  Select the number of components $K$
  Propose $\mu^{(j)}_{1} = \{\mu_{1,1}, \ldots, \mu_{1,K}\}^{(j)}$ by drawing from prior $\mu^{*}_{k} \sim \pi(\mu)$, $k = 1, \ldots, K$
  Propose $\sigma^{2(j)}_{1} = \{\sigma^{2}_{1,1}, \ldots, \sigma^{2}_{1,K}\}^{(j)}$ by drawing from prior $\sigma^{2*}_{k} \sim \pi(\sigma^2)$, $k = 1, \ldots, K$
  Propose $f^{(j)}_{1} = \{f_{1,1}, \ldots, f_{1,K}\}^{(j)}$ by drawing from prior $f^{*}_{k} \sim \pi(f)$, $k = 1, \ldots, K$
  Generate $y^{\text{prop}}_{j}$ from $\sum_{i=1}^{K} f^{(j)}_{i} \cdot \phi \left( y | \mu^{(j)}_{i}, \sigma^{2(j)}_{i} \right)$
  Calculate distance $d_{1}^{(j)} = \rho \{y_{\text{obs}}, y^{\text{prop}}_{j}\}$
end if
else if $2 \leq t \leq T$ then
  for $j = 1, \ldots, N$ do
    Set $\epsilon_{t} = q$th quantile of $\{d_{t-1}^{(j)}\}_{j=1}^{N}$
    Set $d_{t}^{(j)} = \epsilon_{t} + 1$
    while $d_{t}^{(j)} > \epsilon_{t}$ do
      Select $\left\{f^{*}_{j}, \mu^{*}_{j}, \sigma^{2*}_{j}\right\}$ from $\left\{f_{t-1}^{(j)}, \mu_{t-1}^{(j)}, \sigma_{t-1}^{2(j)}\right\}_{j=1}^{N}$ with probabilities
      $\left\{W_{t-1}^{(j)}/\sum_{M=1}^{N} W_{t-1}^{(M)}\right\}_{j=1}^{N}$
      Propose $f_{i}^{(j)}$ according to the Dirichlet resampling functions (see Algorithm 2 in §2.2)
      Propose $\mu^{(j)}_{i} \sim \phi_{K}(\mu | \mu^{*}_{i}, \tau^{2}_{\mu,t-1})$, where $\phi_{K}$ is a multivariate Gaussian density.
      Propose $\sigma^{2(j)}_{i} \sim \phi_{K}(\sigma^{2} | \sigma^{2*}_{i}, \tau^{2}_{\sigma,t-1} \text{ with } \sigma^{2} > 0)$
      Generate $y^{\text{prop}}_{j}$ from $\sum_{i=1}^{K} f_{i,j}^{(j)} \cdot \phi \left( y | \mu_{i,j}^{(j)}, \sigma^{2(j)}_{i} \right)$
      Calculate distance $d_{t}^{(j)} = \rho \{y_{\text{obs}}, y^{\text{prop}}_{j}\}$
      Address the label switching problem (see Algorithm 3 in §2.3)
    end while
  end for
end if

Set weight $W_{t}^{(j)} \propto \prod_{i=1}^{K} \pi(\mu^{(j)}_{i}) \pi(\sigma^{2(j)}_{i}) / \sum_{M=1}^{N} W_{t-1}^{(M)} \phi_{K}(\mu^{(j)}_{i} | \mu_{t-1,i}^{(M)}, \tau^{2}_{\mu,t-1} \phi_{K}(\sigma^{2(j)}_{i} | \sigma_{t-1,i}^{2}, \tau^{2}_{\sigma,t-1} \text{ with } \sigma^{2} > 0)$
iteration and a variance equal to twice the empirical variance of the previous iteration’s ABC posterior. This is a reasonable choice if there are no constraints on the support of the parameter of interest. For the means and variances of the mixture model, a multivariate Gaussian distribution and a multivariate Gaussian distribution truncated to \( \mathbb{R}_+ \) were used, respectively. Both perturbation kernels are assigned variances per the suggestion from Beaumont et al. [30] (i.e. variances set to be twice the empirical variances of the previous iteration’s ABC posteriors). When constraints on the parameter support are present, such as for variances or mixture weights, a perturbation kernel should be selected so that it does not propose values outside the parameter’s support. When moving the selected values for the mixture weights, not only is there the constraint that each mixture weight component must be in \([0, 1]\), but it is also required that \( \sum_{i=1}^{K} f_t^* = 1 \), making the Gaussian kernel inappropriate.

In the first iteration of the proposed ABC–PMC algorithm and for \( J = 1, \ldots, N_{\text{init}} \), the mixture weights \( f_1^{(J)} = \{f_{1,1}, \ldots, f_{1,K}\}^{(J)} \) are directly sampled from the prior distribution, which is a Dirichlet\( (\delta) \), where \( \delta = (\delta_1, \ldots, \delta_K) \). For \( t > 1 \) and for \( J = 1, \ldots, N \), proposals are drawn from the previous iteration’s particle system according to their importance weights, as defined in Algorithm 1. After randomly selecting some particle \( L \in \{1, \ldots, N\} \) from the \( t-1 \) iteration’s particle system we want to “jitter” or move the selected particle’s mixture weight, \( f_t^* \), in manner that preserves some information coming from the selected particle, but not let it be an identical copy, in order to obtain the resampled mixture weights, \( f_t^{(J)} \). This is carried out using Algorithm 2 and the mathematical justification can be found in Appendix A. In Algorithm 2, for each resampled mixture weight for iteration \( t = 2, \ldots, T \) and mixture component \( i = 1, \ldots, K \), rely on the random variable denoted \( \xi_{t,i}^\ast \) (the particle index \( J \) is suppressed for notational convenience in what follows), which is designed to be the sum of two independent random variables, \( Z_t f_t^* B_{t,i} \sim \text{Gamma}(\bar{\delta}, 1) \) and \( \eta_{t,i} \sim \text{Gamma}((1-p)\delta_i, 1) \). The resampled mixture weight is \( f_t = \xi_{t,i}^\ast / \xi_{t,+}^\ast \), with \( \xi_{t,+}^\ast = \sum_{i=1}^{K} \xi_{t,i}^\ast \), so that \( f_t \sim \text{Dirichlet}(\delta) \). The key benefits of this procedure are that the resampled weight vector, \( f_t \), retains some information from the \( f_t^* \) (i.e. the previous iterations particle that was selected for resampling) through the parameter \( p \) (discussed next) but also follows the same distribution as the prior, Dirichlet\( (\delta) \), so that the importance weights due to this resampling are simply unity.

**Algorithm 2** Resampling the mixture weights to obtain \( f_t^{(J)} = \{f_{t,1}, \ldots, f_{t,K}\}^{(J)} \)

Given \( f_t^* \) randomly selected from \( \{f_{t-1}^{(J)}\}_{J=1}^{N} \) with probabilities \( \{W_t^{(J)}/\sum_{M=1}^{N} W_t^{(M)}\}_{J=1}^{N} \)

1. Draw \( Z_t \sim \text{Gamma}(\delta_+, 1) \), with \( \delta_+ = \sum_{i=1}^{K} \delta_i \) and set \( \xi_{t,i}^\ast = Z_t f_t^* \). Then \( \xi_{t,i}^\ast \sim \text{Gamma}(\delta_i, 1) \), \( i = 1, \ldots, K \).
2. Select a real number \( p \in [0, 1] \). (We suggest \( p = 0.5 \).)
3. Draw \( B_{t,i} \sim \text{Beta}(p\delta_i, (1-p)\delta_i) \) independently for \( i = 1, \ldots, K \), where \( \xi_{t,i} B_{t,i} \sim \text{Gamma}(p\delta_i, 1) \).
4. Draw \( \eta_{t,i} \sim \text{Gamma}((1-p)\delta_i, 1) \) independently.
5. Set \( \xi_{t,i} = Z_t f_t^* B_{t,i} + \eta_{t,i} \) and \( f_t^{(J)} = \xi_{t,i}^\ast / \xi_{t,+}^\ast \), with \( \xi_{t,+}^\ast = \sum_{i=1}^{K} \xi_{t,i}^\ast \). Then \( f_t^{(J)} = \{f_{t,1}, \ldots, f_{t,K}\}^{(J)} \sim \text{Dirichlet}(\delta_1, \ldots, \delta_K) \).
The parameter $p$ is a fixed real number with range $[0, 1]$ that determines how much information to retain from $f_t^*$. The choice of $p$ has an impact on both the allowed variability of the marginal ABC posterior distributions for the mixture weights and the efficiency of the entire procedure. In particular, fixing a $p$ close to 1 leads to a Dirichlet resampling in which the new set of mixture weights $f_t$ is close to the previous set $f_t^*$ (if $p = 1$, then $f_t = f_t^*$). On the other hand, a choice for $p$ close to 0 implies that the information coming from $f_t^*$ is weakly considered (for $p = 0$ a new set of particles is drawn directly from the prior distribution and no information about $f_t^*$ has been retained). We found $p = 0.5$ to be a good choice for balancing efficiency and variability (i.e. it allows for some retention of information of the selected particle, but does not lead to nearly identical copies of it).

### 2.3. Algorithm for addressing the label switching problem

As noted earlier, a common problem arising in the Bayesian mixture models framework is label switching. When drawing a sample from a posterior (for both MCMC and ABC), the sampled values are not necessarily ordered according to their mixture component assignments because the likelihood is exchangeable. For example, suppose a particle \{$(f^{(J)}_1, \ldots, f^{(J)}_K), (\mu^{(J)}_1, \ldots, \mu^{(J)}_K), (\sigma^{2(j)}_1, \ldots, \sigma^{2(j)}_K)$\}, where $J \in \{1, \ldots, N\}$, is accepted for a K component GMM. (Note that in this section, we do not include the algorithm step index, $t$, to simply the notation.) This particle was selected with a particular ordering of the $1, \ldots, K$ components with $f^{(j)}_i$, $\mu^{(j)}_i$ and $\sigma^{2(j)}_i$ coming from the same mixture component, $i = 1, \ldots, K$; however, a new particle that is accepted will not necessarily follow that same ordering of the $i = 1, \ldots, K$ components. Somehow the particles have to be ordered in such a way that aligns different realizations of the $i = 1, \ldots, K$ components in order to eliminate the ambiguity.

Several approaches have been proposed to address the label switching problem and are known as relabelling algorithms. A first group of relabelling algorithms consists of imposing an artificial identifiability constraint in order to arbitrarily pick a parameter (e.g. the mixture weights) and sort all the parameters for each accepted particle according to that parameter’s order [7,20]. However, many of the algorithms proposed for addressing the label switching are deterministic (e.g. Stephen’s method [21] and the pivotal reordering algorithm [4]). A third class of strategies, called probabilistic relabeling algorithms, uses a probabilistic approach for addressing the label switching problem [55]. A detailed overview of methods for addressing label switching is presented in Papastamoulis [56]. In Section 3.1, we provide an example that illustrates a problem with the artificial identifiability constraint approach. Instead, we propose a deterministic strategy for addressing the label switching by selecting a parameter that has at least two well-separated components.

Addressing the label switching problem is especially important for the proposed sequential ABC algorithm because each time step’s ABC posterior is used as the proposal in the subsequent step of the algorithm so the label switching has to be resolved before using it as a proposal distribution. Algorithm 3 outlines the proposed strategy and is carried out at the end of each iteration. The key aspect of Algorithm 3 is to select a parameter for reordering that has at least two well-separated components. To determine this, each set of parameters (e.g. the means, the variances, the mixture weights) is arranged in increasing order. For example, for each particle $J, J = 1, \ldots, N, \mu^{(j)}$ would be ordered so that
Algorithm 3 Addressing the label switching problem

1. For each parameter set, obtain the ordered particles \( \mu_{(1)}^{(j)} \leq \mu_{(2)}^{(j)} \leq \cdots \leq \mu_{(K)}^{(j)} \), with \( \mu_{(i)}^{(j)} \) as the \( i \)-th order statistic. This is carried out for each set of parameters with analogous notation.

The next step is to determine which set of parameters has the best separated components. We propose first shifting and scaling each set of parameters to be supported within the range \([0,1]\) so that scaling issues are mitigated and the different parameter set values are comparable. One option for this adjustment is to use some distribution function, such as a Gaussian distribution with a mean and variance equal to the sample mean and the sample variance of the considered parameter set (e.g. the sample mean for the \( \mu \) is \( \bar{\mu} = (NK)^{-1} \sum_{k=1}^{K} \sum_{i=1}^{N} \mu_{(i)}^{(j)} \), and the sample variance for the \( \mu \) is \( \text{var}(\mu) = (NK)^{-1} \sum_{k=1}^{K} \sum_{i=1}^{N} (\mu_{(i)}^{(j)} - \bar{\mu})^2 \)). The resulting \( k \)-th largest standardized value for the mixture mean is \( \tilde{\mu}_{(k)}^{(j)} \). This is carried out for each set of parameters with analogous notation.

For each component of each ordered and standardized particle a representative value, such as a mean, is computed (e.g. \( \tilde{\mu}_{(k)} = N^{-1} \sum_{i=1}^{N} \tilde{\mu}_{(i)}^{(j)} \) is the representative value of the \( k \)-th component of the mean parameter). This is carried out for each set of parameters with analogous notation. The pairwise distances (pdist) between the representative values within each parameter set is determined. The parameter that has the largest separation between any two of its representative values is selected for the overall ordering of the particle system.

Variations of Algorithm 3 were considered. For example, rather than sorting based on the parameter set with largest separation between any two of its representative values, we considered basing the sorting on the parameter with the largest separation between its two closest representative values (i.e. the maximum of the minimum separations); however, this sorting did not perform well empirically. The issue seemed to be that parameter with the largest separation between its two closest representative values may actually have all of its components relatively close; after multiple iterations, none of the components were well separated from the other components. This lead to iteration after iteration of components that remained a blend of components rather than separating clearly into \( K \) components.
Overall, from empirical experiments, the approach outlined in Algorithm 3 performed the best and thus is our recommendation. However, we emphasize that alterations to this procedure may be necessary for mixture models that have additional structure or correlations among the parameters of the component distributions or to deal with multi-dimensional mixture models.

### 2.4. Summary statistics

Comparing the entire simulated dataset, $y_{\text{prop}}$, with the entire set of observations, $y_{\text{obs}}$, in an ABC procedure is not computationally feasible. For this reason, the definition of a lower dimensional summary statistic is necessary. For mixture models, due to the multimodality of the data, common summaries such as means or higher order moments (e.g. variance, skewness or kurtosis) do not capture relevant aspects of the distribution which can lead to incorrect ABC posterior distributions. However, an estimate of the density of the data can better account for its key features (e.g. the shape of each component of the mixture). For the examples presented in Sections 3 and 4, we suggest using kernel density estimates (KDEs [57]) of the simulated data, $\hat{f}_{y_{\text{prop}}}$, and the observations, $\hat{f}_{y_{\text{obs}}}$, to summarize the data, and then the Hellinger distance, $H$, to carry out the comparison. We considered other distances, such as the Kolmogorov–Smirnov (KS) distance [58,59], but found the Hellinger distance to perform better than any other tested distance function. The Hellinger distance quantifies the similarity between two density functions, $f$ and $g$, and is defined as

$$H(f, g) = \left( \int \left( \sqrt{f(y)} - \sqrt{g(y)} \right)^2 \, dy \right)^{\frac{1}{2}}. \quad (5)$$

At each iteration $t$ of the proposed ABC–PMC procedure, a proposed $\theta$ is accepted if $H(\hat{f}_{y_{\text{obs}}}, \hat{f}_{y_{\text{prop}}}) < \epsilon_t$, where $\epsilon_t$ is the tolerance.

### 3. Simulation study

In this section, a simulation study is presented to evaluate the behaviour of the proposed ABC–PMC algorithm, Algorithm 1, using popular GMM examples. In particular, we are interested in evaluating the success of the procedure with respect to the label switching problem and the reliability of the Hellinger distance as a summary statistic. To determine the number of iterations, $T$, a stopping rule was defined based on the Hellinger distance between sequential ABC posteriors; once the sequential Hellinger distance dropped below a threshold of 0.05 for each of the marginal ABC posteriors, the algorithm was stopped.

#### 3.1. Mixture model with equal and unequal group sizes

The first example is taken from Mena and Walker [19], which considered a GMM obtained by simulating data coming from $K = 2$ groups of equal size which was designed for evaluating performance with respect to the label switching problem. The 40 observations were simulated as follows:

$$Y_{i=1\ldots,20} \sim \phi(y \mid -20, 1), \quad Y_{i=21\ldots,40} \sim \phi(y \mid 20, 1), \quad (6)$$
Figure 1. Comparison between the ABC and the MCMC marginal posterior distributions for the two-component GMM example from Mena and Walker [19]. The true values for the means and the mixture weights are displayed as vertical dot-dashed red lines. The final ABC posteriors obtained using the label switching (LS) procedure proposed in §2.3 are the solid black lines (ABC Posterior Alg.3 LS), and the naive approach that sorts based on the mixture weights are the dashed blue lines (ABC Posterior naive LS). The MCMC posteriors are displayed with dotted cyan lines (MCMC Posterior). The number of particles for the ABC analysis and the number of elements kept from the MCMC analysis (after the burn-in) are equal to 5000: (a) Marginal posteriors for $\mu$ and (b) marginal posteriors for $f$.

where $\phi$ is the Normal density function as defined previously. The variance is assumed known for both groups and hence the parameters of interest are the mixture weights, $f = (f_1, f_2)$ (with $f_2 = 1 - f_1$), and the means $\mu = (\mu_1, \mu_2)$. The prior distributions selected for this analysis are $\mu_i \sim \phi(\mu | 0, 100)$, and $f = (f_1, f_2) \sim \text{Dirichlet}(1, 1)$. The number of particles is set to $N = 5000$ and the quantile used for shrinking the tolerance is $q = 0.5$. The algorithm was stopped after $T = 20$ iterations, since further reduction of the tolerance did not lead to a notable improvement of the ABC posterior distributions (evaluated by calculating the Hellinger distance between the sequential ABC posterior distributions as noted previously).

Figure 1 displays the resulting ABC posteriors ('ABC Posterior Alg. 3 LS' and 'ABC Posterior naive LS' are discussed later for illustrating the label switching issue) and the corresponding MCMC posteriors, which are used as a benchmark to access the performance of the proposed method. The ABC posteriors closely match the MCMC posteriors. The summary of the results presented in Table 1 demonstrates that the ABC posterior distributions are a suitable approximation of the MCMC posteriors. The Hellinger distances between the marginal ABC and MCMC posteriors are also displayed in the last column of Table 1; the Hellinger distance between the MCMC and the ABC posterior is 0.032 for the mixture weights and 0.21 for the means.

As noted in Section 2.3, the label switching problem has to be carefully addressed when using the ABC–PMC algorithm. For each time step following the initial step, the previous step’s ABC posterior is used as the proposal rather than the prior distribution so the procedure for addressing the label switching proposed in Section 2.3, Algorithm 3, is used at the end of each time step. In order to illustrate the consequences of incorrectly addressing the
Table 1. Posterior means (posterior standard deviations) obtained by using the MCMC and the ABC–PMC algorithm for the two-component GMM example from Mena and Walker [19]. The fourth column indicates the Hellinger distance between the final ABC and the MCMC posteriors. The number of ABC particles and the number of elements retained from the MCMC chain (after the burn-in) are both equal to 5000.

| Parameter (input) | MCMC (SD)       | ABC (SD)       | H    |
|------------------|----------------|---------------|------|
| \( f_1(0.5) \)   | 0.5008(0.076)  | 0.5003(0.076) | 0.032|
| \( f_2(0.5) \)   | 0.4991(0.076)  | 0.4996(0.076) | 0.032|
| \( \mu_1(-20) \) | -19.72(0.18)   | -19.70(0.18)  | 0.21 |
| \( \mu_2(20) \)  | 20.05(0.19)    | 20.10(0.19)   | 0.21 |

Table 2. Posterior means (posterior standard deviations) obtained by using the MCMC and the ABC–PMC algorithm for the three-component GMM example from Mena and Walker [19]. The fourth column is the Hellinger distance between the final ABC posterior distribution and the MCMC posterior. The number of particles and the number of elements retained from the MCMC chain (after the burn-in) are both equal to 5000.

| Parameter (input) | MCMC (SD)       | ABC (SD)       | H    |
|------------------|----------------|---------------|------|
| \( f_1(0.44) \)  | 0.44(0.071)    | 0.44(0.071)   | 0.024|
| \( f_2(0.12) \)  | 0.12(0.048)    | 0.12(0.048)   | 0.018|
| \( f_3(0.44) \)  | 0.44(0.071)    | 0.44(0.071)   | 0.033|
| \( \mu_1(-20) \) | -19.61(0.26)   | -19.73(0.22)  | 0.27 |
| \( \mu_2(0) \)   | -0.33(0.45)    | -0.30(0.48)   | 0.17 |
| \( \mu_3(20) \)  | 20.06(0.24)    | 20.19(0.22)   | 0.29 |

label switching, we ran the proposed ABC algorithm on the example proposed by Mena and Walker [19], except rather than using Algorithm 3, the ordering of the particle system is carried out using the ordering of the mixture weights; the mixture weights are both equal to 0.5 for this example, making them a poor choice for attempting to separate out the mixture components. The resulting ABC posteriors are displayed in Figure 1 (blue dashed lines). The means, \( \mu = (\mu_1, \mu_2) \), of the mixture components are shuffled and not close to the MCMC posterior, while the mixture weights are sorted such a way all the elements of \( f_1 \) are smaller than 0.5 and all the elements of \( f_2 \) are larger than 0.5.

Mena and Walker [19] added a third component to the mixture in Equation (6), simulating five additional observations from a standard Gaussian distribution, obtaining a three-component GMM with known variances. The ABC–PMC algorithm was run with the same specifications as the first part of the example, but required \( T = 25 \) steps to achieve our stopping rule.

Figure 2 shows the MCMC and the ABC posteriors for the weights and the means of the mixture components. The behaviour of the ABC posterior distributions is consistent with their MCMC benchmarks. A summary of the results presented in Table 2 shows that the posterior means and the posterior standard deviations for the ABC posterior distributions are consistent with the ones retrieved using MCMC. Finally, in the third column, the Hellinger distances between the ABC and MCMC posteriors are provided.

3.2. Mixture model with unequal group size

Even in those cases in which the definition of the mixture model does not lead to the label switching problem, a second category of issues related to the multimodality of the
Figure 2. Comparison between the ABC and MCMC marginal posterior distributions for the three-component GMM example from Mena and Walker [19]. The true values for the means and the mixture weights are displayed as vertical dot-dashed red lines. The final ABC posteriors obtained using the label switching (LS) procedure proposed in §2.3 are the solid black lines (ABC Posterior) while the MCMC posteriors are displayed with dotted cyan lines (MCMC Posterior). The number of particles for the ABC analysis and the number of elements kept from the MCMC analysis (after the burn-in) are equal to 5000: (a) marginal posteriors for $\mu$ and (b) marginal posteriors for $f$.

likelihood function is present. This behaviour has been studied from both the frequentist and Bayesian perspectives. In particular, Marin et al. [4] defined the following simple two-component mixture model to illustrate the multimodality issue,

$$ f_1 \cdot \phi(\mu | \mu_1, 1) + (1 - f_1) \cdot \phi(\mu | \mu_2, 1), $$  \hspace{1cm} (7)

where the weight $f_1$ is assumed known and different from 0.5 (avoiding the label switching problem). According to the specifications by Marin et al. [4], $n = 500$ samples were drawn from Equation (7), with $\theta = (f_1, \mu_1, \mu_2) = (0.7, 0, 2.5)$. The bimodality of the likelihood function (see Figure 3) makes the use of both the EM algorithm [13] and the Gibbs Sampler [20] risky because their success depends on the set of initial values selected for initiating the algorithms.

The PMC sampler [4,38] is used as a benchmark for the proposed ABC–PMC solution. Figure 3 displays the log likelihood function (note the two modes), and the final ABC–PMC posteriors with MCMC posteriors using good and bad starting values along with the posteriors using the PMC algorithm. Table 3 lists the means for the final ABC, MCMC, and PMC posteriors for $\mu_1$ and $\mu_2$, along with the Hellinger distance between the final ABC–PMC posteriors and the PMC posterior. The ABC–PMC posteriors closely match the PMC posteriors.

4. Application to galaxy data

The galaxy dataset was introduced to the statistical community in Roeder [60] and has been commonly used for testing clustering methods. The data contains the recessional velocities
Figure 3. (a) The log-likelihood surface of the Gaussian mixture model proposed by Marin et al. [4]. There are two modes in the log-likelihood function, one close to the true value, (2.5, 0), highlighted with a red cross, and a local mode indicated with a blue circle. (b) The marginal PMC (solid orange lines), ABC (dashed black lines) and MCMC posterior distributions; the displayed MCMC posteriors include the results for good initial starting values as dotted blue lines (MCMC Posterior (good start)) and bad initial starting values as dot-dashed cyan lines (MCMC Posterior (bad start)). The true means are displayed as vertical long-dashed red lines. (a) Log–likelihood surface for $\mu$ and (b) marginal posteriors for $\mu$.

Table 3. Posterior means (posterior standard deviations) obtained by using MCMC (with good and poor choices for initializing the procedure), PMC, and ABC–PMC algorithms in the example by Marin et al. [4]. The last column indicates the Hellinger distance between the final ABC posterior distributions and the PMC posteriors.

| Parameter (input) | $\mu_1$ (0) | $\mu_2$ (2.5) |
|------------------|-------------|---------------|
| PMC (SD)         | -0.17 (0.11)| 2.29 (0.17)   |
| ABC–PMC (SD)     | -0.16 (0.11)| 2.29 (0.17)   |
| MCMC$_{\text{good}}$ (SD) | -0.16 (0.11) | 2.31 (0.16)   |
| MCMC$_{\text{bad}}$ (SD) | 1.86 (0.80)  | -0.18 (0.29)  |
| H                | 0.051       | 0.048         |

of 82 galaxies (km/ s) from six well separated sections of the Corona Borealis region. The galaxy data has been well-studied by the statistical community [7, 8, 61, 62]. The recessional velocities of the galaxies are typically considered realizations of independent and identically distributed Gaussian random variables, but there is discrepancy in the conclusions about the number of groups in the GMM; estimates vary from three components [8] to five or six [7].

In this analysis, we focused on the model with three components [8], in order to be consistent with [19] and [4]. For the hyperparameters, we use the Empirical Bayes approach suggested by Casella et al. [53]. Additionally, since each recessional velocity was also assigned a measurement error, the ABC forward model has been modified to take into account this information. In order to include the measurement errors in the forward model, each simulated recessional velocity is assigned one of the observed measurement errors.
The simulated and observed recessional velocities were matched according to their ranks, and the measurement error of the observations were assigned to the simulated data according to this matching. Then, Gaussian noise was added to each simulated recessional velocity with a standard deviation equal to its assigned measurement error.

The posterior means for each component’s parameters are listed in Table 4. The third component was found to have a weight equal to 0.057, and a mean and variance equal to 32.94 $km/s$ and 1.16 $km^2/s^2$, respectively. The main difference with the proposed ABC–PMC estimates and those from Marin et al. [4], in which they also fixed $K = 3$, is with the estimated mean and variance of the third component; however, the proposed ABC–PMC estimates are consistent with those obtained by Mena and Walker [19].

By using the additional information about the measurement errors, the proposed ABC–PMC algorithm can provide posterior distributions that better reflect the uncertainty in the observations. Including measurement errors in the forward model affects the resulting ABC posterior as reported in Table 4 and the relative estimates plotted in

![Figure 4. Histogram of the recessional velocity of 82 galaxies and the estimated three-component Gaussian mixture models for each study. The results obtained by the proposed ABC–PMC algorithm are comparable with the ones obtained using MCMC. When accounting for the measurement error in the data, the first and third components of the mixture model have more clearly defined modes (dashed orange line). The posterior means for the mixture weights, means and variances used are displayed in Table 4.](image-url)
orange in Figure 4. In particular, the estimated variances are smaller for the proposed ABC–PMC algorithm when measurement errors are accounted for, which results in more clearly defined clusters for the first and third components of the mixture model.

5. Concluding remarks

The popularity of ABC is, at least in part, due to its capacity to handle complex models. Extensions of the basic algorithm have lead to improved efficiency of the sampling, such as the ABC–PMC algorithm of Beaumont et al. [30]. We propose an ABC–PMC algorithm that can successfully handle finite mixture models. Some of the challenges with inference for finite mixture models are due to the complexity of the likelihood function including its possible multimodality and the exchangeability of the mixture component labels leading to the label switching problem. Fortunately, ABC can handle complicated likelihood functions, but the label switching problem needs to be addressed. We suggest a procedure for addressing the label switching problem within the proposed ABC algorithm that works well empirically. Some additional challenges with using ABC for mixture models include the selection of informative summary statistics, and defining a kernel for moving the mixture weights since they are constrained to be between 0 and 1 and must sum to 1. For the summary statistics, we propose using the Hellinger distance between kernel density estimates of the real and simulated observations, which allows the multimodality of the data to be accounted for and compared between the two sets of data. We propose a Dirichlet resampling algorithm for moving the mixture component weights that preserves some information from the sampled particle, but also improves the efficiency of the ABC–PMC procedure (by not having to draw from the Dirichlet prior at each time step).

The proposed ABC algorithm has been explored and tested empirically using popular examples from the literature. The resulting ABC posteriors were compared to the corresponding MCMC posteriors, and in all cases considered the proposed ABC and MCMC posteriors were similar, as desired. We also considered the galaxy velocity data from the Corona Borealis Region [60], which is often used in assessing the performance of mixture models. An advantage of ABC over other commonly used methods is that the forward model can be easily expanded to better represent the physical process that is being modelled. For the galaxy data, measurement errors are available in the original data set, but are not generally used when analysing the data. We extend the proposed ABC–PMC forward model used in this example to include the measurement errors, which provides a more accurate assessment of the uncertainty in the data.

Though the presented examples focused on one-dimensional GMMs, the component distributions can be adapted to other distributions in the ABC forward model. Overall, the proposed ABC–PMC algorithm performs well and is able to recover the benchmark MCMC posteriors (when they were available) suggesting that the proposed ABC–PMC algorithm is a viable approach for carrying out inference for FMMs.

Acknowledgments

Simola thanks the CSC project 2002013 Oletusprojekti. Simola and Cisewski-Kehe also thank the Yale Center for Research Computing for guidance and use of the research computing infrastructure.
Disclosure statement

No potential conflict of interest was reported by the author(s).

References

[1] Pearson K. Contributions to the mathematical theory of evolution. Philos Trans R Soc Lond A. 1894;185:71–110.
[2] Pearson K. Mathematical contributions to the theory of evolution. ii. skew variation in homogeneous material. Proc Royal Soc London. 1894;57(340–346):257–260.
[3] McLachlan G, Peel D. Finite mixture models. John Wiley & Sons; 2004.
[4] Marin J-M, Mengersen K, Robert CP. Bayesian modelling and inference on mixtures of distributions. Handbook of Stat. 2005;25:459–507.
[5] Frühwirth-Schnatter S. Finite mixture and Markov switching models. Springer Science & Business Media; 2006.
[6] Lancaster T, Imbens G. Case-control studies with contaminated controls. J Econom. 1996;71(1):145–160.
[7] Richardson S, Green PJ. On bayesian analysis of mixtures with an unknown number of components (with discussion). J Royal Stat Soc Ser B (Stat Methodol). 1997;59(4):731–792.
[8] Roeder K, Wasserman L. Practical Bayesian density estimation using mixtures of normals. J Am Stat Assoc. 1997;92(439):894–902.
[9] Henderson R, Shimakura S. A serially correlated gamma frailty model for longitudinal count data. Biometrika. 2003;90(2):355–366.
[10] Gianola D, Heringstad B, Odegaard J. On the quantitative genetics of mixture characters. Genetics. 2006;173(4):2247–2255.
[11] Lin C-Y, Lo Y, Ye KQ. Genotype copy number variations using gaussian mixture models: theory and algorithms. Stat Appl Genet Mol Biol. 2012;11(5):0–00.
[12] Wu G, Holan SH, Nilon CH, et al. Bayesian binomial mixture models for estimating abundance in ecological monitoring studies. Ann Appl Stat. 2015;9(1):1–26.
[13] Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the em algorithm. J Roy Stat Soc Ser B. 1977;1(39):1–38.
[14] Naim I, Gildea D. Convergence of the em algorithm for gaussian mixtures with unbalanced mixing coefficients. 2012. arXiv preprint arXiv:1206.6427.
[15] Miyahara H, Tsumura K, Sugihiyama Y. Relaxation of the em algorithm via quantum annealing for gaussian mixture models. Decision and Control (CDC 2016, IEEE 55th Conference on; IEEE; 2016. p. 4674–4679.
[16] Robert C. The Bayesian choice: from decision-theoretic foundations to computational imple-mentation. Springer Science & Business Media; 2007.
[17] Casella G, Mengersen KL, Robert CP, et al. Perfect samplers for mixtures of distributions. J Royal Stat Soc Ser B (Stat Methodol). 2002;64(4):777–790.
[18] Shabram M, Demory B-O, Cisewski J, et al. The eccentricity distribution of short-period planet candidates detected by Kepler in occultation. Preparing for submission. 2014.
[19] Mena RH, Walker SG. On the Bayesian mixture model and identifiability. J Comput Graph Stat. 2015;24(4):1155–1169.
[20] Diebolt J, Robert CP. Estimation of finite mixture distributions through Bayesian sampling. J Royal Stat Soc Ser B (Methodol). 1994;56(2):363–375.
[21] Stephens M. Dealing with label switching in mixture models. J Royal Stat Soc Ser B (Stat Methodol). 2000;62(4):795–809.
[22] Jasra A, Stephens DA, Holmes CC. On population-based simulation for static inference. Stat Comput. 2007;17(3):263–279.
[23] Rodriguez CE, Walker SG. Label switching in Bayesian mixture models: deterministic relabeling strategies. J Comput Graph Stat. 2014;23(1):25–45.
[24] Stoneking CJ. Bayesian inference of gaussian mixture models with noninformative priors. 2014. arXiv preprint arXiv:1405.4895.
[25] Rubin DB. Bayesianly justifiable and relevant frequency calculations for the applied statistician. Ann Stat. 1984;12(4):1151–1172.

[26] Tavaré S, Balding DJ, Griffiths RC, et al. Inferring coalescence times from DNA sequence data. Genetics. 1997;145:505–518.

[27] Pritchard JK, Seielstad MT, Perez-Lezaun A. Population growth of human Y chromosomes: A study of Y chromosome microsatellites. Mol Biol Evol. 1999;16(12):1791–1798.

[28] Sisson SA, Fan Y, Tanaka MM. Sequential Monte Carlo without likelihoods. Proc Natl Acad Sci. 2007;104(6):1760–1765.

[29] Joyce P, Marjoram P. Approximately sufficient statistics and Bayesian computation. Stat Appl Genet Mol Biol. 2008;7(1):1–16.

[30] Beaumont MA, Cornuet J-M, Marin J-M, et al. Adaptive approximate Bayesian computation. Biometrika. 2009;96(4):983–990.

[31] Csilléry K, Blum MGB, Gaggiotti OE, et al. Approximate Bayesian computation (ABC) in practice. Trends Ecol Evol (Amst). 2010;25(7):410–418.

[32] Del Moral P, Doucet A, Jasra A. An adaptive sequential Monte Carlo method for approximate Bayesian computation. Stat Comput. 2012;22(5):1009–1020.

[33] Koutoumpas K, Ballarini P, Votsi I, et al. Bayesian parameter estimation for the WNT pathway: an infinite mixture models approach. Bioinformatics. 2016;32(17):i781–i789.

[34] Marin J-M, Pudlo P, Robert CP, et al. Approximate Bayesian computational methods. Stat Comput. 2012;22(6):1167–1180.

[35] Ratmann O, Camacho A, Meijer A, et al. Statistical modelling of summary leads to accurate approximate Bayesian computations. Unpublished. 2013.

[36] Blum MGB. Approximate Bayesian computation: A nonparametric perspective. J Amer Stat Assoc. 2010;105(491):1178–1187.

[37] Blum MGB, Nunes MA, Prangle D, et al. A comparative review of dimension reduction methods in approximate Bayesian computation. Stat Sci. 2013;28(2):189–208.

[38] Prangle D. Summary statistics in approximate Bayesian computation. 2015. arXiv preprint arXiv:1512.05633.

[39] Weyant A, Schafer C, Wood-Vasey WM. Likelihood-free cosmological inference with type Ia supernovae: approximate Bayesian computation for a complete treatment of uncertainty. Astrophys J. 2013;764:116.

[40] Lenormand M, Jabot F, Deuant G. Adaptive approximate Bayesian computation for complex models. Comput Stat. 2013;28(2):2777–2796.

[41] Ishida EEO, Vitenti SDP, Penna-Lima M, et al. cosmoabc: Likelihood-free inference via population Monte Carlo approximate Bayesian computation. 2015. arXiv preprint arXiv:1504.06129.

[42] Cisewski-Kehe J, Weller G, Schafer C, et al. A preferential attachment model for the stellar initial mass function. Electron J Stat. 2019;13(1):1580–1607.

[43] Simola U, Pelssers B, Barge D, et al. Machine learning accelerated likelihood-free event reconstruction in dark matter direct detection. J Instrum. 2019;14(03):P03004.

[44] silk D, Filippi S, Stumpf MPH. Optimizing threshold-schedules for sequential approximate Bayesian computation: applications to molecular systems. Stat Appl Genet Mol Biol. 2013;12(5):603–618.
Appendix 1. Mixture weight resampling kernel

The steps for resampling the mixture weights at iteration $t$, $\{f_t^{(j)}\}_{j=1}^N$, are outlined in Algorithm 2 with details presented in Section 2.2. In this section, we present mathematical justification for the proposed resampling procedure. In particular, we want to show that the resampled weights, $f_t = (f_{t,1}, \ldots, f_{t,K})$ with $f_{t,i} = \xi_{t,i}^{*} / \sum_{i=1}^{K} \xi_{t,i}^{*}$, $i = 1, \ldots, K$, $\xi_{t,+} = \sum_{i=1}^{K} \xi_{t,i}$, and $t = 2, \ldots, T$, follows a Dirichlet($\delta_1, \ldots, \delta_K$).

Recall that to resample a particle at iteration $t$, a particle from the previous iteration, $t-1$, is randomly sampled according to the importance weights, $\{W_{t-1}^{(j)}/ \sum_{j=1}^{N} W_{t-1}^{(j)}\}_{j=1}^N$. Suppose some mixture weight vector $f_{t-1} = (f_{t-1,1}, \ldots, f_{t-1,K})$, $t \in \{2, \ldots, T\}$ has been selected for resampling with some finite number of components $K > 1$. To show that carrying out the resampling steps as outlined in Algorithm 2 results in $f_t \sim \text{Dirichlet}(\delta_1, \ldots, \delta_K)$, consider the following propositions. Proposition A.1 and A.4 correspond to Step 5 of Algorithm 2, and Proposition A.2 and A.3 correspond to Steps 1 and 3, respectively.

**Proposition A.1:** Let $f_{t-1} = \left(\frac{\xi_{t-1,1}}{\xi_{t-1,+}}, \ldots, \frac{\xi_{t-1,K}}{\xi_{t-1,+}}\right)$ where $\xi_{t-1,i} \overset{\text{ind}}{\sim} \text{Gamma}(\delta_i, \beta)$ and $\xi_{t-1,+} = \sum_{i=1}^{K} \xi_{t-1,i}$, then $f_{t-1} \sim \text{Dirichlet}(\delta)$.

**Proof:** This is a well-known result that can be proved by using the change-of-variables approach with $y_1 = \xi_{t-1,1}/y_K, \ldots, y_{K-1} = \xi_{t-1,K-1}/y_K$, and $y_K = \xi_{t-1,+}$. The determinant of the Jacobian of the transformation is $y_K^{K-1}$. Noting the joint distribution $\xi_{t-1,1}, \ldots, \xi_{t-1,K}$...
This follows directly from Theorem 1 of Yeo and Milne [63].

**Proof:**

\[ e^{-\sum_{i=1}^{K} \beta \xi_{t-1,i}} \prod_{i=1}^{K} \beta^{\xi_{t-1,i}} \frac{\xi_{t-1,i}^{\delta_i-1}}{\Gamma(\delta_i)}, \]

then the joint distribution using the change-of-variables is

\[ y_1, \ldots, y_K \sim \beta^{\sum_{i=1}^{K} \delta_i} e^{-\beta y_K} \left( y_K \left( 1 - \sum_{i=1}^{K-1} y_i \right) \right)^{\delta_k-1} \prod_{i=1}^{K} \frac{(y_i y_k)^{\delta_i-1}}{\prod_{i=1}^{K} \Gamma(\delta_i)}. \]

Then, integrating out the \( y_K \) results in

\[ y_1, \ldots, y_{K-1} \sim \left( 1 - \sum_{i=1}^{K-1} y_i \right)^{\delta_k-1} \prod_{i=1}^{K-1} \Gamma(\delta_i) \prod_{i=1}^{K-1} y_i^{\delta_i-1}. \]

Noting that \( 0 \leq y_1, \ldots, y_{K-1} \leq 1 \) and \( 0 \leq \sum_{i=1}^{K-1} y_i \leq 1 \), then we can let \( y_K = 1 - \sum_{i=1}^{K-1} y_i \), resulting in

\[ y_1, \ldots, y_K \sim \frac{\Gamma(\sum_{i=1}^{K} \delta_i)}{\prod_{i=1}^{K} \Gamma(\delta_i)} \prod_{i=1}^{K} y_i^{\delta_i-1}, \]

which is a Dirichlet(\( \delta_1, \ldots, \delta_K \)) distribution.

**Proposition A.2:** Let \( f_{t-1} \sim \text{Dirichlet}(\delta) \) with \( \delta = (\delta_1, \ldots, \delta_K) \) and \( Z_t \sim \text{Gamma}(\delta_+, \beta) \) where \( \delta_+ = \sum_{i=1}^{K} \delta_i \). Then \( \xi_{t,i} = Z_{t} f_{t-1,i} \sim \text{Gamma}(\delta_i, \beta) \).

**Proof:** This follows from Theorem 1 of Yeo and Milne [63] by noting that the marginal distribution of a Dirichlet(\( \delta \)), that is, the distribution of \( f_{t-1,i} \), is a Beta(\( \delta_i, \sum_{i=1}^{K} \delta_j - \delta_i \)).

**Proposition A.3:** Given \( p \in [0, 1] \) and independent \( B_{t,i} \sim \text{Beta}(p \delta_i, (1-p) \delta_i) \) for \( i = 1, \ldots, K \), then \( \xi_{t,i} B_{t,i} \sim \text{Gamma}(p \delta_i, \beta) \).

**Proof:** This follows directly from Theorem 1 of Yeo and Milne [63].

**Proposition A.4:** Given \( p \in [0, 1] \), independent \( B_{t,i} \sim \text{Beta}(p \delta_i, (1-p) \delta_i) \), independent \( \eta_{t,i} \sim \text{IndGamma}((1-p) \delta_i, \beta) \), \( i = 1, \ldots, K \), and set \( \xi_{t,i}^* = Z_t f_{t-1,i} B_{t,i} + \eta_{t,i} \) and \( f_{t,i} = \xi_{t,i}^*/\xi_{t,i}^+ \), with \( \xi_{t,i}^+ = \sum_{i=1}^{K} \xi_{t,i}^* \). Then \( f_t = (f_{t,1}, \ldots, f_{t,K}) \sim \text{Dirichlet}(\delta_1, \ldots, \delta_K) \).

**Proof:** By Propositions A.2 and A.3, \( Z_t f_{t-1,i} B_{t,i} \sim \text{Gamma}(p \delta_i, \beta) \), and then \( \xi_{t,i}^* \sim \text{Gamma}(\delta_i, \beta) \) by noting that the sum of the two independent gamma random variables with the same rate parameter, \( \beta \), is also a gamma random variable with rate parameter \( \beta \) and with the sum \( \delta_i = p \delta_1 + (1-p) \delta_1 \) as the shape parameter. Then Proposition A.1 gives the distribution for \( f_t \).