Author:
Watson, Sophie

Title:
Sequential Methods in Approximate Bayesian Computation

Sequential Methods in Approximate Bayesian Computation

Sophie Watson

A dissertation submitted to the University of Bristol
in accordance with the requirements for award of the degree of
Doctor of Philosophy in the Faculty of Science

School of Mathematics, September 2017

Word count: 60,000

Abstract

This thesis concentrates on improving on existing methodology for sequential Monte Carlo (SMC) algorithms within Approximate Bayesian Computation.

Approximate Bayesian Computation (ABC) provides a methodology for estimating the posterior distribution of parameters θ, given observed data y, in cases where the likelihood function is intractable, provided one can simulate data under the model of interest. ABC algorithms can be highly computationally expensive to implement, due to the large number of model simulations required. This thesis gives alterations to the SMC-ABC algorithm of Del Moral et al. [1], which aim to reduce the computational cost and level of user tuning required in within ABC. Furthermore, the accuracy of the estimated posterior distribution is sensitive to the way in which the data, y is summarised and optimal summary statistics are unknown for non-trivial models. This thesis proposes an iterative method for selecting summary statistics, which is implemented within an SMC-ABC algorithm.

Recently there has been a move towards empirically modelling the likelihood function, within the ABC literature. In the final chapter of this theses, we present two algorithms which use density estimation to model the likelihood, and show that this has the potential to dramatically reduce the computational cost of ABC, by lowering the number of model simulations required.

Acknowledgements

I would like to thank my supervisor Mark Beaumont for his support and direction. His knowledge, time and tutoring has been crucial to my thesis, but most importantly his patience, positivity, kindness and calmness has been indispensable.

I thank my father Paul for filling my life with educational opportunities from an early age, and I thank my mother Andy for teaching me that hard work pays off.

I am grateful to my Grandparents for supporting me in all my educational endeavours.

I thank Elizabeth Wainwright for filling my days in Bristol with coffee, wine and friendship. However, this thesis would have been finished much sooner if you hadn't been such an excellent distraction.

I thank Kathryn, Nick, Emma, Bex, Liz C, Lis, Lindsay, Fionnuala, Patricks Cannon and Rubin-Delanchey, Matt, Justin, Angus, Fionnuala, and all my other friends and colleagues for making Bristol such a fabulous place to be.

Finally, I thank Jory Griffin for his enduring love and support.

Author's declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: ..... DATE:.....

‘

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Data is more abundant than ever. We are now able to quickly generate and store vast amounts of data from an increasing number of systems and models. Due to increasing computer power, data generative models have become abundant and, combined with a better understanding of the world around us, have enabled an increase in the accuracy of models of real life systems. Such models are now common in a variety of fields. For example, geneticists can sequence human DNA and build models which simulate the evolution of populations over millions of years, yet take mere seconds to run on a standard laptop computer. Similarly, meteorologists can build intricate models with which we are able to predict how the global climate will change over the coming years.

However, creating these, often complex, models introduces new statistical challenges. In particular, there is the need for algorithms that can make inference about increasingly high dimensional models with many parameters. Such models are often stochastic which makes them difficult, or impossible, to analyse through standard analytical methods. Furthermore, the models are commonly intractable, meaning that the likelihood function is unknown: given a parameter value θ, it is not possible to write down the probability of observing data x. Because of this, standard statistical methods cannot be used, and alternative approaches are needed. One major cause of this intractability is the presence of latent variables in the models. These variables mean that the likelihood involves an integral over latent states.

Approximate Bayesian Computation (ABC) is a class of methods that enable inference on models with intractable likelihoods, provided that we have access to a data generative model which we can simulate from.

1.1 Aims of Thesis

This thesis aims to develop ABC methodologies which are easily implementable, and require little user tuning. We hope that such methods result in more accurate posterior inference, compared to existing methods, and ideally this would be achieved at a lower computational cost than existing methods.

Chapter 2

An introduction to Approximate Bayesian Computation

In this chapter we introduce the basic concepts of ABC and give an overview of existing ABC methodologies. We illustrate the methods using the example of a univariate Gaussian distribution.

2.1 Approximate Bayesian Computation (ABC)

Suppose we have a model from which we can simulate data $\boldsymbol{x}$, for a given parameter value $\boldsymbol{\theta}$. We are concerned with the task of determining the distribution of parameter values that could have given rise to some observed data $\boldsymbol{y}$. This conditional distribution of interest, $p(\boldsymbol{\theta}|\boldsymbol{y})$, is known as the *posterior distribution*, and is a common target of inference within Bayesian computation. Analytically, this posterior distribution is computed as

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{\pi(\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{\theta})}{p(\boldsymbol{y})}, \tag{2.1}$$

where

- $\pi(\cdot)$ is the *prior distribution* over parameter values $\boldsymbol{\theta} \in \Theta$. The prior distribution captures one's pre-held belief about the parameter values.

- $p(\cdot|\boldsymbol{\theta})$ is the *likelihood function*, and denotes the probability of observing some data, given a particular parameter value $\boldsymbol{\theta}$.

- $p(\boldsymbol{y})$ is the *marginal probability* of the data $\boldsymbol{y}$, and is given by

$$p(\boldsymbol{y}) = \int_{\Theta} p(\boldsymbol{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}. \tag{2.2}$$

For simple models, it is common that the likelihood, $p(\boldsymbol{y}|\boldsymbol{\theta})$, is analytically tractable, and thus the true posterior distribution is computable exactly using Equation (2.1). In cases where the likelihood is tractable up to an unknown normalising constant, algorithms have been developed to enable one to sample from the posterior distribution. However, for complex models of real world systems, the likelihood is often fully intractable, and thus we cannot evaluate Equation (2.1) analytically. In such cases, inference requires the use of a class of *likelihood free* algorithms. Approximate Bayesian computation (ABC) is a subset of such likelihood free methods, and is the focus of this thesis.

In 1984 Rubin [2] presented the first known instance of an ABC algorithm. In order to simulate from the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{y})$, for some observed data $\boldsymbol{y}$, the author proposes a thought experiment in which one samples parameters from the prior distribution $\pi(\boldsymbol{\theta})$ then simulates data under a model, $p(\cdot|\boldsymbol{\theta})$ using the sampled parameter values. Rubin noted that the set of parameters which generated data that is *equal* to the observed data, $\boldsymbol{y}$, are an exact sample from the posterior distribution. However, he went on to point out that for continuous, high dimensional data, the proposed algorithm would require infinite iterations to obtain just one sampled $\boldsymbol{\theta}$ from the posterior distribution.

It was not until 1999 that a variant of Rubin's algorithm addressing this limitation was implemented. Pritchard et al. [3] developed the algorithm so that it would give samples from the target distribution in finite time. This was achieved by relaxing the requirement that simulated data was an exact match for the observed data, and instead accepting parameters that generated data which fell within a distance ϵ of the observed data. But, the increasing number of acceptances of the algorithm comes at a cost - this algorithm now targets an *approximate* posterior

distribution, known as the ABC posterior, and denoted $p_\epsilon(\boldsymbol{\theta}, \boldsymbol{x}|\boldsymbol{y})$, where

$$p_\epsilon(\boldsymbol{\theta}, \boldsymbol{x}|\boldsymbol{y}) \propto \pi(\boldsymbol{\theta}) p(\boldsymbol{x}|\boldsymbol{\theta}) \mathbb{I}\{\tilde{\rho}(\boldsymbol{x}, \boldsymbol{y}) < \epsilon\}, \tag{2.3}$$

$\tilde{\rho}(\boldsymbol{x}, \boldsymbol{y})$ denotes the distance between $\boldsymbol{x}$ and $\boldsymbol{y}$, and $\mathbb{I}(\cdot)$ is the indicator function, defined by

$$\mathbb{I}\{z\} = \begin{cases} 1, & \text{if } z \text{ true,} \\ 0, & \text{otherwise.} \end{cases} \tag{2.4}$$

From Equation (2.3) we see that there is a trade-off between the size of ϵ and the closeness of the ABC posterior to the true posterior distribution. In the limit as $\epsilon \to 0$ the exact posterior distribution is recovered. However, setting $\epsilon = 0$ gives us Rubin's Algorithm, which requires infinite time to run when applied to any non-trivial model.

2.2 Summary Statistics in ABC

Another key feature of ABC algorithms is the use of *summary statistics*, which were also introduced by Pritchard et al. [3]. In their methodology, the observed data was Y chromosome data, taken from 8 loci across 445 samples of human males from around the world. For such data it is apparent that we would not wish to compare two samples at each position along the genome, as this would be extremely time consuming, and likely uninformative, since it is highly unlikely that two samples would 'match'. Thus, the authors summarised each individual data set by its variance, heterozygosity and number of haplotypes. These three summary statistics were selected as they are known to be informative for the parameters of inference in this example, namely mutation rate, population size and time since most recent common ancestor. Therefore, a comparison between the summaries of the simulated and observed can be compared and parameters accepted or rejected accordingly.

Summary statistics enable a *simple* comparison to be made between simulated and observed data.

Let $S(\boldsymbol{x}) = S_1(\boldsymbol{x}), \ldots, S_p(\boldsymbol{x})$ denote the summary statistics of data set $\boldsymbol{x}$, and let ρ be a distance metric on the space of summary statistics. We now accept a proposed parameter $\boldsymbol{\theta}$ if $\rho(S(\boldsymbol{x}), S(\boldsymbol{y})) < \epsilon$. Thus the target of such an algorithm is now

$$p_\epsilon(\boldsymbol{\theta}, S(\boldsymbol{x})|S(\boldsymbol{y})) \propto \pi(\boldsymbol{\theta})p(S(\boldsymbol{x})|\boldsymbol{\theta})\mathbb{I}\{\rho(S(\boldsymbol{x}), S(\boldsymbol{y})) < \epsilon\}, \tag{2.5}$$

where $\mathbb{I}\{\cdot\}$ denotes the indicator function as given in Equation (2.4), and $\rho(\cdot, \cdot)$ is a distance metric on the space of summary statistics. From Equation (2.5) we see that it is no longer necessarily the case that the ABC posterior converges to the true posterior in the limit as $\epsilon \to 0$. Such convergence only occurs in a specific case where the summary statistics are *sufficient* for the parameter of interest.

Theorem 2.1. *Summary statistic $S(\boldsymbol{x})$ is sufficient for parameter $\boldsymbol{\theta}$ if, and only if,*

$$p(\boldsymbol{x}|S(\boldsymbol{x}), \boldsymbol{\theta}) = p(\boldsymbol{x}|S(\boldsymbol{x})). \tag{2.6}$$

An equivalent representation of Theorem 2.1 is to say that summary statistic $S(\boldsymbol{x})$ is sufficient for parameter $\boldsymbol{\theta}$ if and only if $\boldsymbol{x}$ is conditionally independent of $\boldsymbol{\theta}$ given $S(\boldsymbol{x})$. Intuitively this means that the summary statistic captures all the information about the $\boldsymbol{\theta}$ which is held in the data. In practice it is not straightforward to identify sufficient statistics for parameters of interest, except in the cases where the likelihood is tractable, and thus there is little need for ABC. In Section 2.5.2 we give a brief overview of existing algorithms for selection summary statistics. In practice, it can be simpler to show sufficiency by illustrating that the likelihood function can be factorised using the following Fisher-Neyman factorisation Theorem:

Theorem 2.2. *Fisher-Neyman factorisation Theorem. Summary statistic $S(\boldsymbol{x})$ is sufficient for parameter $\boldsymbol{\theta}$ if and only if there exists some non-negative functions g and h such that*

$$f(\boldsymbol{x}|\boldsymbol{\theta}) = h(\boldsymbol{x})g(S(\boldsymbol{x})|\boldsymbol{\theta}). \tag{2.7}$$

Using Theorem 2.2 it is possible to show that sufficient statistics for the Multivariate Gaussian distribution are the sample mean and sample covariance matrix, given by

$$\bar{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i \quad \text{and} \quad \hat{\Sigma} = \sum_{i=1}^{n} (\boldsymbol{x}_i - \bar{\boldsymbol{x}})^T (\boldsymbol{x}_i - \bar{\boldsymbol{x}}), \tag{2.8}$$

In the rest of this Chapter we use the Bivariate Gaussian distribution as an example to illustrate the basics of ABC.

2.3 Rejection ABC

We are now able to combine the concepts introduced earlier in the chapter to present the standard form of Rejection ABC. Using a distance metric on summary statistics, as seen in Pritchard et al. [3] and Tavaré et al. [4], as well as the tolerance, ϵ, used by Pritchard et al. [3]. Algorithm 1 targets the ABC posterior distribution.

Algorithm 1 Rejection ABC

Let $S(\cdot)$ be a summary of data $\boldsymbol{x}$, and $\rho(\cdot, \cdot)$ be a distance metric on the space of summary statistics $S(\boldsymbol{x})$. Let $\pi(\cdot)$ be a prior distribution over the space of parameters $\boldsymbol{\theta}$, and let $\boldsymbol{y}$ denote the observed data.

1: Fix tolerance $\epsilon > 0$ and $N > 0$.
2: Sample $\boldsymbol{\theta} \sim \pi(\cdot)$
3: Simulate $\boldsymbol{x} \sim p(\cdot|\boldsymbol{\theta})$
4: If $\rho(S(\boldsymbol{x}), S(\boldsymbol{y})) < \epsilon$ accept $\boldsymbol{\theta}$.

Repeat until there has been N acceptances

We now implement Rejection ABC on the Gaussian distribution with unknown mean and variance, using sufficient statistics.

Example 2.1. *Bivariate Gaussian with unknown mean and covariance*

Consider a sample from the Bivariate Gaussian distribution with unknown mean $\boldsymbol{\mu} \in \mathbb{R}^2$*, and unknown covariance matrix* $\Sigma \in \mathbb{M}_+^{2\times 2}$*, where* $\mathbb{M}_+^{2\times 2}$ *denotes the set of 2 by 2 symmetric, positive semi-definite matrices. We follow the notation and parametrisation given in Gelman et al. [5].*

We wish to compute the posterior distribution $p(\boldsymbol{\mu}, \Sigma|\mathbf{y})$*. We use the conjugate* $Normal-Inv-Wishart$ *prior distribution, thus we have that*

$$\Sigma \sim Inv - Wishart(\nu_0, \Lambda_0^{-1}) \tag{2.9}$$

$$\boldsymbol{\mu}|\Sigma \sim \mathcal{N}(\boldsymbol{\mu}_0, \Sigma/k_0), \tag{2.10}$$

where $\boldsymbol{\mu}_0, \Lambda_0, \nu_0$ *and* k_0 *are hyper-parameters, and* Λ_0 *is positive definite.*

The posterior distribution also follows a $Normal-Inverse-Wishart$ *distribution, with updated hyper-parameters given by*

$$\boldsymbol{\mu}_n = \frac{\kappa_0 \boldsymbol{\mu}_0}{\kappa_0 + n} + \frac{n}{\kappa_0 + n} \times \bar{\boldsymbol{y}}, \tag{2.11}$$

$$\kappa_n = \kappa_0 + n, \tag{2.12}$$

$$\nu_n = \nu_0 + n, \tag{2.13}$$

$$\Lambda_n = \Lambda_0 + S_{\boldsymbol{y}} + \frac{n\kappa_0}{\kappa_0 + n} \times (\bar{\boldsymbol{y}} - \boldsymbol{\mu}_0)(\bar{\boldsymbol{y}} - \boldsymbol{\mu}_0)^T, \tag{2.14}$$

where

$$\bar{\boldsymbol{y}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{y}_i \quad \text{and } S_{\boldsymbol{y}} = \sum_{i=1}^{n} (\boldsymbol{y}_i - \bar{\boldsymbol{y}})(\boldsymbol{y}_i - \bar{\boldsymbol{y}})^T. \tag{2.15}$$

The marginal posterior distribution for $\boldsymbol{\mu}$ *follows a non-central Students-t distribution:*

$$\boldsymbol{\mu}|\boldsymbol{y} \sim t_{\nu_n - d + 1}\left(\boldsymbol{\mu}_n, \frac{\Lambda_n}{\kappa_n(\nu_n - d + 1)}\right). \tag{2.16}$$

The posterior distribution for Σ *follows an Inverse-Wishart distribution, with* ν_n *degrees of freedom, and scale matrix* Λ_n^{-1}*.*

We select the following hyper-parameters:

$$\boldsymbol{\mu}_0 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \quad \kappa_0 = 1, \quad \nu_0 = 4 \quad \textit{and} \quad \Lambda_0^{-1} = \begin{pmatrix} \frac{4}{3} & -\frac{2}{3} \\ -\frac{2}{3} & \frac{4}{3} \end{pmatrix}. \tag{2.17}$$

The observed data, $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{10})$ *is simulated by*

$$\boldsymbol{y}_i \sim^{iid} \mathcal{N}\left(\begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0.9 & 0.5 \\ 0.5 & 0.9 \end{pmatrix} \right). \tag{2.18}$$

The sample mean $\bar{\boldsymbol{x}}$ *and the sample covariance* $\hat{\Sigma}$ *are sufficient for* $\boldsymbol{\mu}$ *and* Σ*. Thus we use these as summary statistics. The observed summary statistics are*

$$\bar{\boldsymbol{y}} = \begin{pmatrix} -1.0136 \\ 0.9764 \end{pmatrix} \quad \textit{and} \quad S_{\boldsymbol{y}} = \begin{pmatrix} 0.8820 & 0.1474 \\ 0.1474 & 0.8425 \end{pmatrix}. \tag{2.19}$$

We select $\rho(S(\boldsymbol{x}), S(\boldsymbol{y}))$ *to be Euclidean distance between the summary statistics and we scale the summary statistics by the marginal sample standard deviations of the statistics, based on a preliminary sample. (The scaling of summary statistics and the selection of distance metrics is discussed in detail in Section 2.6.2.)*

Using the update rules in Equations (2.11) *to* (2.14)*, and the hyper-parameters given in Equation* (2.17)*, the updated hyper-parameters are given by*

$$\boldsymbol{\mu}_n = \begin{pmatrix} -1.0124 \\ 0.9785 \end{pmatrix}, \quad \kappa_n = 11, \quad \nu_n = 14 \quad \textit{and} \quad \Lambda_n^{-1} = \begin{pmatrix} 0.1170 & -0.0249 \\ -0.0249 & 0.1218 \end{pmatrix}. \tag{2.20}$$

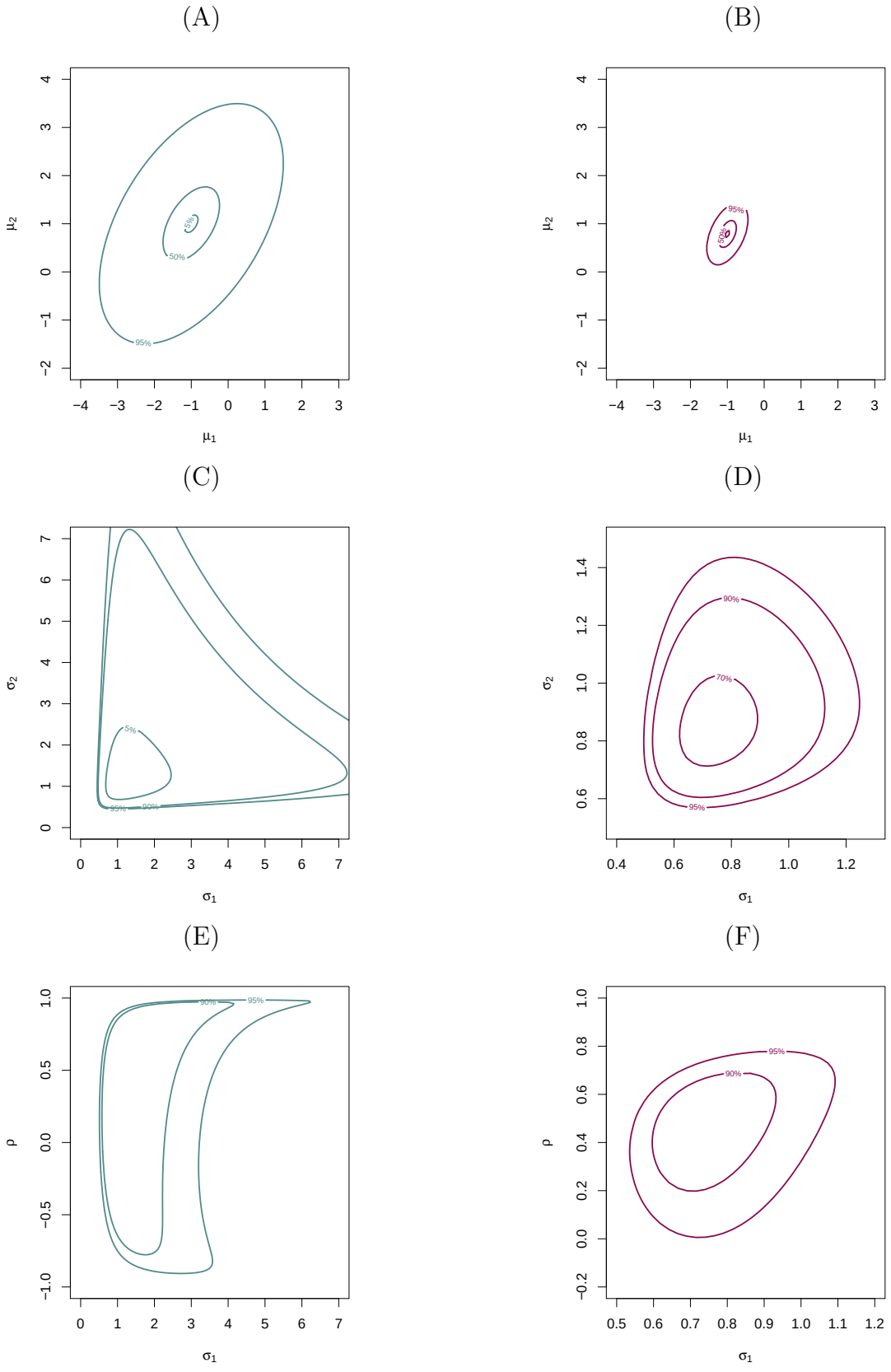


FIGURE 2.1: Prior distributions (left) and posterior distributions (right) for parametrisations of the Bivariate Gaussian distribution, conditional upon the hyper-parameters and observed data given in Example 2.1.

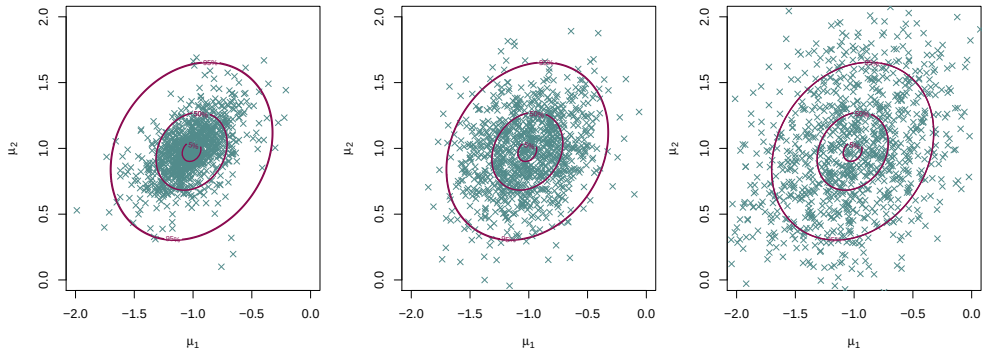
Figure 2.1 shows the contours of the prior distribution and analytic posterior distributions for the parameters of the Bivariate Gaussian Distribution, as given in Example 2.1. The contours for covariance matrix Σ are plotted in terms of σ_1, σ_2 and ρ, where

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}. \tag{2.21}$$

We implement Algorithm 1, iterating the algorithm until the posterior distributions contain $1,000$ *parameters. This is repeated for* $\epsilon \in (0.1, 0.5, 1)$*. Quantitative properties of the resultant ABC posteriors are given in Table 2.1, and the posteriors for* $\boldsymbol{\mu}$ *are plotted in Figure 2.2.*

ϵ	μ_1	μ_2	σ_1^2,	$\rho\sigma_1\sigma_2$,	σ_2^2	*iterations*
0.1	0.0471	0.0409	0.2082	0.0578	0.1948	293,058 (0.34 %)
0.5	0.0925	0.0869	0.2906	0.0890	0.2975	3,325 (30.08 %)
1	0.1855	0.2073	0.3345	0.1542	0.4143	1,552 (64.43 %)

TABLE 2.1: Mean squared error of the ABC posteriors obtained through Rejection ABC, for varying tolerances ϵ. The final column states the number of iterations of the algorithm needed until there are 1,000 acceptances, and the corresponding acceptance rate of the algorithm.



(A) $\epsilon = 0.1$ (B) $\epsilon = 0.5$ (C) $\epsilon = 1$

FIGURE 2.2: 1000 samples from an ABC posterior for $\boldsymbol{\mu}$, obtained through Rejection ABC for three different values of ϵ. 95%, 50% and 5% contours for the true posterior distributions plotted in pink.

Table 2.1 shows that the ABC posteriors give better estimates of the posterior distribution, in terms of reduced MSE, as ϵ *decreases. However, nearly 290,000 extra simulations from the model were needed to obtain 1,000 acceptances for* $\epsilon = 0.1$*, compared to the number needed for* $\epsilon = 0.5$*. For this simple model, which is very fast to simulate from, these extra simulations ran quickly,*

but for an expensive model one can see that as computational cost increases greatly the time taken for the algorithm to run becomes great.

Figure 2.2 shows the three posterior samples for $\boldsymbol{\mu}$, as well as the true posterior contours for the three values of ϵ. It is clear from the picture that for smaller values of ϵ, the ABC posterior distribution is more consistent with the contours.

In Example 2.1 we selected the values of ϵ prior to implementing the algorithm. In practice, it is common to select a tolerance level after all model simulations have been made (Beaumont et al. [6]). One selects ϵ such that the parameters corresponding to the $k\%$ smallest distances are accepted. (Commonly, $k < 1$.) By choosing ϵ retrospectively you are sure to obtain a non-empty sample from the posterior distribution and can fix the computational cost of the algorithm prior to implementation. This is desirable as many of the models on which one wishes to use ABC can be computationally expensive to run.

2.4 Monte Carlo Methods in ABC

A major source of inefficiency in Rejection ABC is that one draws samples from the same distribution, $\pi(\cdot)$, at each iteration. It is of course highly likely that parameters from certain regions of the prior do not give rise to simulations which are close to the observed summary statistics, but Rejection ABC does not learn from historical samples, and thus a proportion of parameter values continue to be sampled from regions of negligible posterior mass. This leads to a low acceptance rate, and thus an inefficient ABC implementation. We now show how incorporating more advanced Monte Carlo methods enables a more efficient implementation of ABC.

2.4.1 Monte Carlo Markov Chain ABC (MCMC-ABC)

The Metropolis Algorithm was derived in 1953 by Metropolis et al. [7] to sample from complex distributions. It generates a sequence of samples by sampling from a Markov Chain which has invariant distribution equal to the distribution of interest. The original algorithm used

symmetric proposal distributions $q(\cdot|\boldsymbol{\theta})$ to propose a new sample, given a current sample $\boldsymbol{\theta}$. In 1970, Hastings [8] adapted the algorithm to allow the use of non-symmetric proposal distributions $q(\cdot|\boldsymbol{\theta})$, and hence the resultant algorithm was called the Metropolis-Hastings Algorithm.

The standard form of the Metropolis-Hastings algorithm [7], [8], to target a distribution $p(\boldsymbol{\theta}|\boldsymbol{y})$, using prior distribution $\pi(\boldsymbol{\theta})$, and proposal distribution $q(\boldsymbol{\theta}|\boldsymbol{\theta}')$, requires the computation of the *Metropolis-Hastings ratio*:

$$\frac{p(\boldsymbol{y}|\boldsymbol{\theta}')}{p(\boldsymbol{y}|\boldsymbol{\theta})}\frac{\pi(\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta})}\frac{q(\boldsymbol{\theta}|\boldsymbol{\theta}')}{q(\boldsymbol{\theta}'|\boldsymbol{\theta})}. \tag{2.22}$$

In the ABC setting this ratio cannot be computed, since the likelihoods $p(\boldsymbol{y}|\boldsymbol{\theta}')$ and $p(\boldsymbol{y}|\boldsymbol{\theta})$ are intractable. Marjoram et al. [9] note that, given a simulation from the model, we are able to estimate these intractable likelihoods in the following way:

Let

$$\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m \overset{iid}{\sim} p(\cdot|\theta). \tag{2.23}$$

Then one can replace the intractable likelihoods using the following approximation:

$$p(S(\boldsymbol{y})|\boldsymbol{\theta}) \approx \frac{1}{m}\sum_{i=1}^{m}\mathbb{I}\{\rho(S(\boldsymbol{x}_i), S(\boldsymbol{y})) < \epsilon\}. \tag{2.24}$$

Note that in the case that $m = 1$, Equation (2.24) gives the approximation of the likelihood function used in Rejection ABC. McKinley et al. [10] showed that little improvement was made in ABC for the case where $m > 1$, compared to $m = 1$, and Bornn et al. [11] showed that $m = 1$ is near-optimal. Using Equation 2.24, Marjoram et al. [9] developed a Metropolis-Hastings Monte Carlo algorithm for ABC (MCMC-ABC), which we give in Algorithm 2, for the case when $m = 1$.

The output of Algorithm 2 is a sequence of parameter vectors $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots \boldsymbol{\theta}_{\text{final}}$, which must be treated to obtain a sample from the ABC posterior distribution. This process commonly involves the following two steps:

1. **Burning-in** The Markov Chain takes many iterations to reach its stationary distribution. All values before it reaches stationarity should be discarded. This is known as *burning in* the chain.

Algorithm 2 MCMC-ABC (Marjoram et al. [9])

Let $q(\cdot|\boldsymbol{\theta})$ be a proposal distribution for $\boldsymbol{\theta}$, and let $\pi(\cdot)$ be a prior distribution over parameter space. Fix $\epsilon > 0$, set $t = 1$ and initialise $\boldsymbol{\theta}^{(1)}$. Let $\rho(\cdot, \cdot)$ be a distance metric on the space of summary statistics.

1: Propose $\boldsymbol{\theta}' \sim q(\cdot|\boldsymbol{\theta}^{(t)})$

2: Simulate $\boldsymbol{x} \sim p(\cdot|\boldsymbol{\theta}')$

3: With probability

$$\min\left(1, \frac{\pi(\boldsymbol{\theta}')q(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}^{(t)})q(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(t)})}\mathbb{I}\{\rho(S(\boldsymbol{x}), S(\boldsymbol{y})) < \epsilon\}\right) \tag{2.25}$$

set $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}'$, else set $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)}$.

4: Set $t = t + 1$.

Repeat steps 1 to 4 until the Markov Chain reaches convergence.

2. **Thinning** The Markov Chain produced in Algorithm 2 is correlated. In order to get an uncorrelated sample from the stationary distribution, every lth parameter should be sampled, thereby thinning the chain. The choice of $l > 1$ can be made by considering the autocorrelation of consecutive samples from the chain, and removing samples until the autocorrelation drops below some acceptable threshold.

In the case where the proposal distribution $q(\cdot|\cdot)$ is symmetric, meaning that

$$q(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(t)}) = q(\boldsymbol{\theta}^{(t)}|\boldsymbol{\theta}'), \tag{2.26}$$

the acceptance probability given in Equation (2.25) simplifies to

$$\min\left(1, \frac{\pi(\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}^{(t)})}\mathbb{I}\{\rho(S(\boldsymbol{x}), S(\boldsymbol{y})) < \epsilon\}\right). \tag{2.27}$$

This simplified algorithm corresponds to the ABC generalisation of the original Metropolis algorithm [7].

Example 2.2. *MCMC-ABC for the Bivariate Gaussian with unknown mean $\boldsymbol{\mu}$ and covariance matrix Σ.*

We return to the model first seen in Example 2.1, and use the same prior distribution and hyper-parameters as given in Example 2.1. Implementing Algorithm 2, with $\epsilon = 0.1$, the chain is initialised at parameter values equal to the observed summary statistics. New parameters are

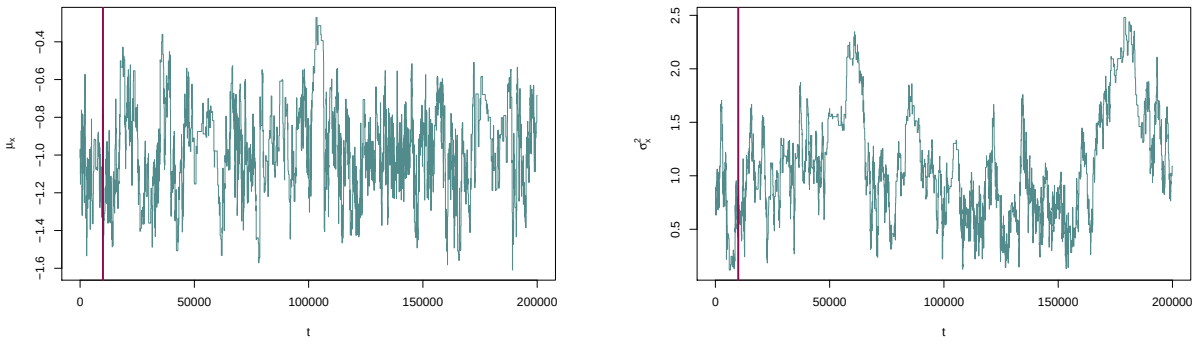


FIGURE 2.3: Trace plots of MCMC-ABC output for μ_1 and σ_1^2. All points to the left of the vertical line are removed in burning in.

proposed marginally for each component of the parameters at time t, using univariate Gaussian distributions, with mean equal to the current parameter value and variance equal to 0.1.

Figure 2.3 shows the marginal trace plots for two dimensions of the Markov Chain, before the burn in iterations are removed and before the chains are thinned. The algorithm is run for $200,000$ *iterations, and the first 10,000 points are removed for burning in. To thin the chain every 290th value is retained, resulting in an ABC posterior of size* $1,000$*. The posterior for* $\boldsymbol{\mu}$ *is plotted in Figure 2.4.*

Method	μ_1	μ_2	σ_1^2,	$\rho\sigma_1\sigma_2$,	σ_2^2	*iterations*
Rejection ABC, $\epsilon = 0.1$	0.0471	0.0409	0.2082	0.0578	0.1948	293,058
MCMC-ABC, $\epsilon = 0.1$	0.0454	0.0263	0.3110	0.0175	0.1149	200,000

TABLE 2.2: Mean squared error of the ABC posteriors for $\epsilon = 0.1$, obtained through MCMC-ABC and Rejection ABC. The final column is the number of iterations of each algorithm.

Table 2.2 gives the posterior MSE for the MCMC-ABC posterior distribution, as well as the MSE for the Rejection ABC posterior using the same tolerance, $\epsilon = 0.1$*. The table shows that although the accuracy of the posterior distributions are comparable, over* $93,000$ *fewer samples from the model were drawn in the MCMC-ABC implementation. This demonstrates that the same accuracy can be obtained for a reduction in computational cost when compared to Rejection ABC.*

In practice, the Metropolis Hastings algorithm can be very sticky, meaning that the Markov Chain can enter regions of low posterior probability, and remain there for many iterations.

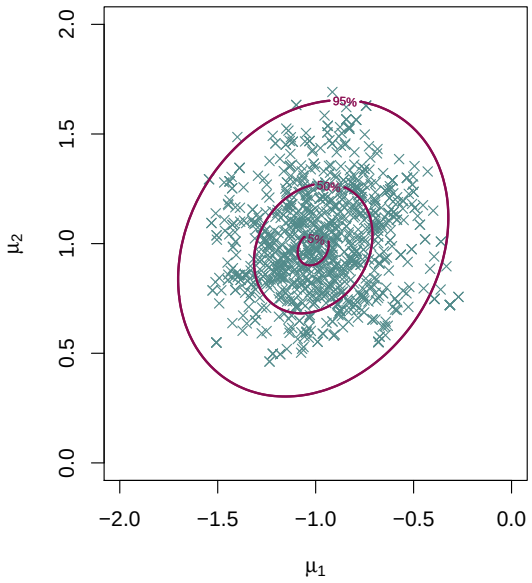


FIGURE 2.4: MCMC-ABC posterior sample for μ_1 and μ_2, with true posterior contours shown.

Furthermore it may take a long time to propose a $\boldsymbol{\theta}'$ for which $\rho(S(\boldsymbol{x}), S(\boldsymbol{y})) < \epsilon$, for ϵ small and, unlike when implementing Rejection ABC, one must specify ϵ prior to running the algorithm.

Bortot et al. [12] proposed an algorithm which removed the need to pre-select a fixed value of ϵ, by specifying a prior distribution $\pi_\epsilon(\cdot)$ over ϵ, and augmenting the state space of the Markov Chain in Algorithm 2, to include the space of tolerances ϵ. Thus proposal distribution is now of the form $q((\cdot,\cdot)|(\boldsymbol{\theta}, \epsilon))$, and at iteration t of the MCMC algorithm, proposal $(\boldsymbol{\theta}', \epsilon')$ is accepted with probability

$$\min\left(1, \frac{\pi(\boldsymbol{\theta}')\pi_\epsilon(\epsilon')q(\boldsymbol{\theta}_t, \epsilon_t|\boldsymbol{\theta}', \epsilon')}{\pi(\boldsymbol{\theta}_t)\pi_\epsilon(\epsilon_t)q(\boldsymbol{\theta}', \epsilon'|\boldsymbol{\theta}_t, \epsilon_t)}\mathbb{I}\{\rho(S(\boldsymbol{x}), S(\boldsymbol{y})) < \epsilon'\}\right). \tag{2.28}$$

Selecting a prior distribution for ϵ which allows for a balance of small and larger tolerance values, the Markov Chain mixes better and is less *sticky*. The chain could later be filtered so that the posterior distribution contains only parameters $\boldsymbol{\theta}$ which correspond to tolerances $\epsilon < \epsilon_T$ for some fixed ϵ_T.

Bortot et al. [12] showed that, in the examples considered in the paper, this augmentation of the state space leads to a much more efficient algorithm, in the sense that the acceptance rate was

significantly higher than when using the standard MCMC-ABC algorithm, as given in Algorithm 2.

However, the output of the MCMC algorithm of Bortot et al. [12] still requires post processing, and in fact may need more post processing than the algorithm of Marjoram et al. [9], since a value of ϵ_T may need to be selected. A sensible prior for ϵ must also be chosen. MCMC algorithms are also sensitive to the choice of perturbation kernel, and the standard deviation of the kernel. Furthermore, Bortot et al. [12] targets a posterior distribution which differs from that targeted in standard ABC-MCMC.

Within the MCMC literature, many adaptations to MCMC have been proposed, and can in many cases lead to improved inference through better mixing of the chain, examples have been presented which can be used to improve MCMC algorithms. One such algorithm is the Metropolis-within-Gibbs algorithm, which updates only one parameter, or a subset of parameters, at iteration t, rather than proposing for all summary statistics, as in Algorithm 2.

Sequential Monte Carlo (SMC) algorithms were developed, removing the need for sample post processing, and with the hope of not suffering from poorly mixing Markov Chains.

2.4.2 Sequential Monte Carlo-ABC (SMC-ABC)

Sequential Monte Carlo algorithms work by iteratively propagating a set of samples from the prior distribution towards the posterior distribution. At each iteration the set of samples, which we refer to individually as *particles*, approximate a distribution and over time this distribution moves towards the targeted posterior distribution. In SMC-ABC, this movement towards the posterior is controlled by decreasing the tolerance ϵ. The first SMC-ABC algorithm was given by Sisson et al. [13]. The algorithm required a pre-specified sequence of decreasing epsilons $\epsilon_1 > \epsilon_2 > \ldots > \epsilon_\tau$ to sample from distributions $p(\boldsymbol{\theta}|\rho(S(\boldsymbol{x}), S(\boldsymbol{y})) < \epsilon_t)$. However, because their algorithm was based on the paper of Del Moral et al. [14], which assumed access to likelihoods, a bias was introduced into the SMC-ABC algorithm. The papers of Beaumont et al. [15], Toni et al. [16] and Sisson [17] amended the algorithm and removed the bias through methods used in importance sampling algorithms. The three algorithms proposed in the three papers are directly comparable, though Sisson [17] presents the algorithm for the more general case where

the initial parameters are drawn from a distribution that may differ from the prior distribution. The algorithm of Beaumont et al. [15] assumes that the perturbation of a particle selected at previous time step is Gaussian, though this is the only difference between the three algorithms. We present this population Monte Carlo algorithm (PMC-ABC) in Algorithm 3.

Algorithm 3 Population Monte Carlo-ABC (PMC-ABC). Beaumont et al. [15]

Fix $\epsilon_1 > \epsilon_2 > \ldots > \epsilon_T$. Set $t = 1$.

1: For $i = 1, \ldots, N$,

a. Simulate $\boldsymbol{\theta}_i^{(1)} \sim \pi(\boldsymbol{\theta})$, and $\boldsymbol{x}_i^{(t)} \sim p(\cdot|\boldsymbol{\theta}_i^{(1)})$
until $\rho(S(\boldsymbol{x}_i^{(t)}), S(\boldsymbol{y})) < \epsilon_1$.

b. Set weights $w_i^{(1)} = 1/N$.

2: Set $t = t + 1$, and set σ_t^2 to twice the empirical variance of the set of $\boldsymbol{\theta}_j^{(t-1)}$

3: For $i = 1 \ldots, N$,
until $\rho(S(\boldsymbol{x}_i^{(t)}), S(\boldsymbol{y})) < \epsilon_t$.

a. sample a $\boldsymbol{\theta}_i'$ from the set of $\theta_j^{(t-1)}$ with probability $w_j^{(t-1)}$.

b. simulate $\boldsymbol{\theta}_i^{(t)} \sim \mathcal{N}(\mu = \boldsymbol{\theta}_i', \sigma^2 = \sigma_t^2)$ and $\boldsymbol{x}_i^{(t)} \sim f(\cdot|\boldsymbol{\theta}_i^{(t)})$

4: For $i = 1 \ldots, N$, set weights

$$w_i^{(t)} \propto \frac{\pi(\boldsymbol{\theta}_i^{(t)})}{\sum_{j=1}^{N} w_j^{(t-1)} \phi\left(\sigma_t^{-1}(\boldsymbol{\theta}_i^{(t)} - \boldsymbol{\theta}_j^{(t)})\right)}, \tag{2.29}$$

where $\phi(z)$ is the probability density of the standard Gaussian distribution.

5: Return to step 2, until $t = T$.

In Algorithm 3, σ_t^2 is a scalar if parameters $\boldsymbol{\theta_i}$ are one dimensional. For higher dimensional $\boldsymbol{\theta_i}$, σ_t^2 is a covariance matrix.

Algorithm 3 uses a Gaussian kernel density estimation of the distribution of $\boldsymbol{\theta}^{(t-1)}$, to sample new parameters at time t. In practice, other density estimation methods can be used, and the weights in Equation 2.29 should be amended accordingly.

The PMC-ABC algorithm is sensitive to the choice of tolerance values $\epsilon_1, \ldots, \epsilon_T$. A poorly selected sequence of tolerance values can lead to a highly inefficient implementation. This is particularly the case when incremental tolerance values are selected to be far apart. Furthermore,

due to the nature of having to select tolerance values prior to implementing the algorithm, any user must have a good idea of a final tolerance value ϵ_T, which will give rise to an ABC posterior distribution which is a good approximation of the true posterior distribution. This is tricky in practice and without sufficient statistics, and under model misspecification, there is no guarantee that small tolerance levels are attainable. Del Moral et al. [1] proposed an SMC-ABC algorithm that removed the need for pre-specification of a sequence of tolerance values. Instead, the algorithm works by retaining the 'fittest' $\alpha\%$ of particles from the previous approximation to the posterior, and using these to form the next proposal distribution. The *fitness* of a parameter set, or *particle*, is determined by the distance between the observed summary statistics and the simulated summary statistics for that parameter vector.

The computation of the weights in Equation (2.29) is of order N^2 for the PMC-ABC algorithm. However the algorithm of Del Moral et al. [1] reduces this weight computation to order N. We discuss the algorithm of Del Moral et al. [1] in detail in Chapter 3.

Figure 2.5 shows the decrease in tolerance values from initialisation to $\epsilon = 0.1$, during an implementation of Del Moral et al. [1]'s SMC-ABC algorithms, for the Bivariate Gaussian Example we first saw in Example 2.1. As you can see, the rate of change in ϵ decreases over time. To obtain a posterior sample with tolerance $\epsilon = 0.1$ required $79,841$ simulations from the model. This is significantly fewer than was required when implementing SMC-ABC and MCMC-ABC with a tolerance of $\epsilon = 0.1$.

2.5 Impact of Summary Statistics on ABC inference

All of the ABC algorithms we have seen so far are sensitive to the choice of summary statistics $S(\boldsymbol{x})$. In the implementations of ABC on Bivariate Gaussian example we have seen earlier in this chapter, we use sufficient statistics for the parameters $\boldsymbol{\mu}$ and Σ. In this section we consider existing methods for selecting summary statistics, when the sufficient statistics are unknown or do not exist. In such cases, it is perhaps natural to consider using a high dimensional summary of the data, in an attempt to capture as much information about the data as possible. However, this is in general a bad idea. Ideally summary statistics should be low dimensional (and of significantly lower dimension than the raw data itself) as, for uncorrelated summary statistics,

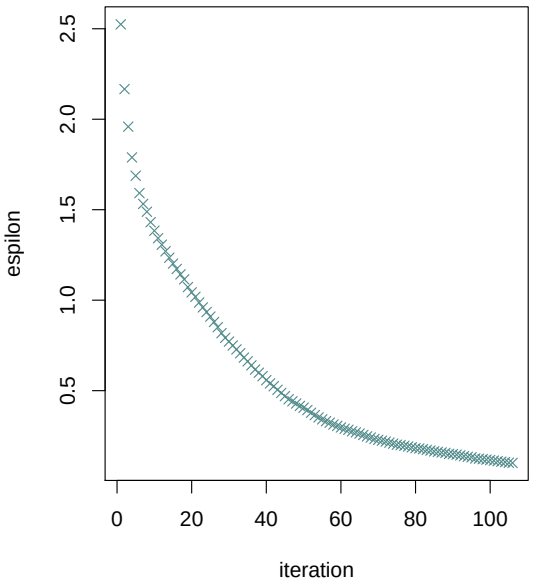


FIGURE 2.5: Decrease in tolerance, ϵ, over iterations, in the SMC-ABC algorithm [1].

there is a negative correlation between the dimensionality of the summary statistics, and the acceptance rate of algorithms such as Rejection ABC, for a fixed tolerance ϵ.

2.5.1 The Curse of Dimensionality

To illustrate the so-called *curse of dimensionality* we consider a simple example. Suppose we select a two-dimensional summary statistic $S(\boldsymbol{x}) = (S_1(\boldsymbol{x}), S_2(\boldsymbol{x}))$, where S_1 and S_2 are independent, and that the following two equations hold:

$$p(\rho(S_1(\boldsymbol{x}), S_1(\boldsymbol{y}) < \epsilon) = p_1 \tag{2.30}$$

$$p(\rho(S_2(\boldsymbol{x}), S_2(\boldsymbol{y}) < \epsilon) = p_2, \tag{2.31}$$

where $\rho(\cdot, \cdot)$ is the Euclidean distance function. Then the probability that the distance between a simulated sample and the observed summary statistics, in two dimensions, is less than ϵ is given by

$$\mathbb{P}(\rho(S(\boldsymbol{x}), S(\boldsymbol{y})) < \epsilon) = p_1 p_2 < \min(p_1, p_2). \tag{2.32}$$

Thus the probability of acceptance, at tolerance level ϵ, decreases as the dimension of summary statistics increase.

To illustrate this in practice we consider the following example:

Example 2.3. *Uniform Samples in a Box. We begin in the one-dimensional case. Let observed data* $y = 0.5$, $\theta \sim \mathcal{U}(0, 1)$, *and let data* $x = \theta$. *We let* $\epsilon = 0.1$. *Thus*

$$\mathbb{P}(\rho(x, y) < \epsilon) = \mathbb{P}(\theta \in (0.4, 0.6)) = 0.2. \tag{2.33}$$

Now we consider the same example but in two dimensions. Let $\boldsymbol{y} = (0.5, 0.5)$, $\boldsymbol{\theta} \sim \mathcal{U}(0, 1)^2$, *and again set* $\boldsymbol{x} = \boldsymbol{\theta}$.*Let* $B_\epsilon(\boldsymbol{y})$ *denote the ball of radius* ϵ, *centred at* $\boldsymbol{y}$. *Now*

$$\mathbb{P}(\rho(\boldsymbol{x}, \boldsymbol{y}) < \epsilon) = \mathbb{P}(\theta \in B_\epsilon(\boldsymbol{y})) = 0.0314. \tag{2.34}$$

This shows that an increase in dimension from one to two results in a decrease in acceptance probability by over 80%, *for a fixed* ϵ.

In a simulation study of this example, $5,073$ *simulations from the prior distribution were required to obtain 1000 acceptances, when implementing Rejection ABC on the one-dimensional example. The two-dimensional example required* $32,904$ *simulations form the model for the same number of acceptances. This is in line with the calculations given in Equations* (2.33) *and* (2.34).

Blum et al. [18] gives an overview of dimension reduction methods and theoretical results relating to the curse of dimensionality.

2.5.2 Summary Statistic Selection Methods

The choice of summary statistics is not straight forward. Ideally one would select sufficient statistics, that is those which satisfy Equation (2.6). In practice, even for simple models, such statistics are difficult to identify, and may not exist (aside from using the full data itself, which is of high dimension).

One large driving force in the choice of summary statistics is the received body of knowledge accumulated in a field of research. For example, Pritchard et al. [3] selected the three summary statistics given in Section 2.2 as, within the field of population genetics, they are known to be informative for the parameters of interest. In other instances, it may be the case that the summary statistics are limited by recordings made in the field, and we may no longer have access to the raw observed data set $\boldsymbol{y}$, but instead just to $S(\boldsymbol{y})$, where the choice of $S(\cdot)$ was determined by a scientist carrying out the experiment.

In this section we consider two main types of summary statistic selection methods. The first, known as *subset selection methods*, aims to select a good set of summary statistics from an initial proposed set of possible summary statistics. The second aims to minimise the mean squared error of ABC, and thus selects summary statistics accordingly.

2.5.2.1 Subset Selection Techniques

Suppose one has a candidate set of summary statistics $S_1(\cdot), \ldots, S_k(\cdot)$, and wishes to find a minimal subset of these k statistics which is most informative for the parameters of interest. We only wish to include a statistic $S_j(\cdot)$ into our subset if $S_j(\boldsymbol{x})$ provides information about parameter $\boldsymbol{\theta}$ which is not captured by any of the previously selected statistics.

In Theorem 2.1 we saw the idea of sufficient statistics, though as previously discussed they are often unknown for complex models. With this in mind, Joyce et al. [19] introduced the notion of *approximately sufficient statistics*, also known as ϵ-sufficient.

Definition 2.5.1. ϵ-sufficient statistics. Summary statistics $S_1(\cdot), \ldots, S_j(\cdot)$ are ϵ-sufficient for parameter $\boldsymbol{\theta}$, relative to an additional statistic $S_{j+1}(\cdot)$ if

$$\sup_{\boldsymbol{\theta}} \ln p(S_{j+1}(\boldsymbol{x})|S_1(\boldsymbol{x}), \ldots, S_j(\boldsymbol{x}), \theta) - \inf_{\boldsymbol{\theta}} \ln p(S_{j+1}(\boldsymbol{x})|S_1(\boldsymbol{x}), \ldots, S_j(\boldsymbol{x}), \boldsymbol{\theta}) \leq \epsilon. \tag{2.35}$$

Such definition is motivated by first considering the likelihood function for summary statistics $S_1(\cdot), \ldots, S_j(\cdot)$. We have that

$$\begin{aligned} p(S_1(\boldsymbol{x}), \ldots, S_j(\boldsymbol{x})|\boldsymbol{\theta}) &= p(S_1(\boldsymbol{x})|\boldsymbol{\theta}) \times p(S_2(\boldsymbol{x})|S_1(\boldsymbol{x}), \boldsymbol{\theta}) \times \\ &\quad \cdots \times p(S_j(\boldsymbol{x})|S_1(\boldsymbol{x}), S_2(\boldsymbol{x}), \ldots, \boldsymbol{\theta}), \end{aligned} \tag{2.36}$$

$$\begin{aligned} \text{and so } \ln\left(p(S_1(\boldsymbol{x}), \ldots, S_j(\boldsymbol{x})|\boldsymbol{\theta})\right) &= \ln\left(p(S_1(\boldsymbol{x})|\boldsymbol{\theta})\right) \\ &\quad + \ln\left(p(S_2(\boldsymbol{x})|S_1(\boldsymbol{x}), \boldsymbol{\theta})\right) + \cdots \\ &\quad + \ln\left(p(S_j(\boldsymbol{x})|S_1(\boldsymbol{x}), S_2(\boldsymbol{x}), \ldots, \boldsymbol{\theta})\right). \end{aligned} \tag{2.37}$$

From Equation (2.37) it follows that

$$\begin{aligned} &\ln(p(S_1(\boldsymbol{x}), \ldots, S_j(\boldsymbol{x}), S_{j+1}(\boldsymbol{x})|\boldsymbol{\theta}) - \ln\left(p(S_1(\boldsymbol{x}), \ldots, S_j(\boldsymbol{x})|\boldsymbol{\theta})\right) \\ &\qquad = \ln(p(S_{j+1}(\boldsymbol{x})|S_1(\boldsymbol{x}), \ldots, S_j(\boldsymbol{x}), \boldsymbol{\theta})). \end{aligned} \tag{2.38}$$

Thus if two log likelihoods are close, there is little information in the additional summary statistic. The authors capture the notion of how informative a summary statistic is by comparing the ratio, $R_k(\boldsymbol{\theta})$, of the two posterior distributions - one with additional summary statistic S_{k+1} and one without.

$$R_j(\boldsymbol{\theta}) = \frac{p(\boldsymbol{\theta}|S_1(\cdot), \ldots, S_{j+1}(\cdot))}{p(\boldsymbol{\theta}|S_1(\cdot), \ldots, S_j(\cdot))}. \tag{2.39}$$

If $R_j(\boldsymbol{\theta})$ deviates significantly from 1, summary statistic S_{j+1} should be included in the set used to determine the ABC posterior.

One disadvantage of this method, as noted by Marin et al. [20], is that the choice of summary statistics is not independent of the order in which they are 'tested' by Equation 2.39.

Nunes and Balding [21] proposed an alternative subset selection method. Like Joyce et al. [19], they aimed to obtain a measure of how much additional information about $\boldsymbol{\theta}$ was held in a new summary statistic s_j. To do so they considered the *entropy* of the posterior distributions obtained using all possible subsets of summary statistics.

The entropy of a distribution is a measure of how informative that distribution is. The smaller the entropy, the more informative the distribution is. During trial runs, once a posterior has been obtained $p(\boldsymbol{\theta}|S_j(\boldsymbol{x}))$, for $S_j \subset S$, the entropy is determined using the k nearest neighbour entropy estimation for a fixed sample size.

Although both the algorithms of Joyce et al. [19] and Nunes and Balding [21] require the computation of many ABC posterior distributions, all of these can be computed using just one preliminary sample from the joint distribution. However, a large sample may be needed for accurate summary statistic selection, and this can greatly increase the computational cost of any ABC implementation, particularly for models which are expensive to simulate from.

2.5.2.2 Minimising Mean Squared Error

Fearnhead and Prangle [22] proposed an alternative approach to selecting summary statistics. The authors note that it is often sufficient to obtain accurate point estimates of the posterior distribution, such as the posterior mean, rather than recovering a full approximation to the distribution. With this in mind they consider which choice of summary statistic minimises the posterior mean squared loss.

Theorem 2.3. *Fearnhead and Prangle [22] The minimal possible quadratic error loss* $\mathbb{E}\{L(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}; A)|\boldsymbol{y}\}$ *occurs when* $\hat{\boldsymbol{\theta}} = \mathbb{E}(\boldsymbol{\theta}|\boldsymbol{y})$.

Proof.

$$\mathbb{E}(\mathcal{L}(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}; A)|\boldsymbol{y}) := \mathbb{E}((\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T A(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})) \tag{2.40}$$

$$= \int (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^T A(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) p(\boldsymbol{\theta}|\boldsymbol{y})\, d\boldsymbol{\theta} \tag{2.41}$$

So the minimum expected loss is obtained by differentiating Equation 2.41, and equating it to 0. We know that this is a minimum for the function as A is a positive definite matrix.

So we have that

$$0 = 2A \int (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}) p(\boldsymbol{\theta}|\boldsymbol{y}) \, d\boldsymbol{\theta} \tag{2.42}$$

$$0 = \int \boldsymbol{\theta} p(\boldsymbol{\theta}|\boldsymbol{y}) - \int \hat{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\boldsymbol{y}) \, d\boldsymbol{\theta} \tag{2.43}$$

$$\mathbb{E}(\boldsymbol{\theta}|\boldsymbol{y}) = \hat{\boldsymbol{\theta}} \int p(\boldsymbol{\theta}|\boldsymbol{y}) \, d\boldsymbol{\theta} \tag{2.44}$$

$$\hat{\boldsymbol{\theta}} = \mathbb{E}(\boldsymbol{\theta}|\boldsymbol{y}) \tag{2.45}$$

□

Thus from Theorem 2.3 Fearnhead and Prangle [22] showed that posterior loss, or mean squared error, is minimised by the posterior mean, which we denote $\mathbb{E}(\boldsymbol{\theta}|S = S(\boldsymbol{y}))$. However this quantity is unknown to the user when implementing ABC. Indeed, if we had access to such value, there would be no need to implement ABC at all, under the assumption that a point estimate of the posterior distribution is sufficient. Thus an estimate of the posterior mean is required. It suggested in [22] that one estimates the posterior mean through linear regression. The authors note that more sophisticated estimates are of course possible, but at some cost to the overall efficiency of any algorithm, and that in the examples considered in the paper, linear regression was sufficiently accurate.

Based on a preliminary sample from the joint distribution of parameters and summary statistics $(\boldsymbol{\theta}_i, S(\boldsymbol{x}_i))$, for $i \in 1, \ldots, s$, the following linear model is suggested:

$$\boldsymbol{\theta}_i^T = S(\boldsymbol{x}_i)^T \beta + \boldsymbol{\epsilon}_i^T, \tag{2.46}$$

where

$$\theta_i = \begin{bmatrix} \theta_{i,1} \\ . \\ . \\ \theta_{i,d} \end{bmatrix}, \quad S(\boldsymbol{x}_i) = \begin{bmatrix} 1 \\ S_1(\boldsymbol{x_i}) \\ . \\ . \\ S_p(\boldsymbol{x_i}) \end{bmatrix}, \quad \boldsymbol{\epsilon}_i = \begin{bmatrix} \epsilon_{i,1} \\ . \\ . \\ \epsilon_{i,d} \end{bmatrix} \tag{2.47}$$

and $\beta \in \mathbb{R}^{((p+1)\times d)}$ is a matrix of regression coefficients and ϵ_i is white noise.

In Chapter 3 we consider a method for iteratively updating the least squares estimate of matrix β throughout SMC-ABC, thus providing an improved estimate of the posterior mean.

2.6 Distance Metrics and Scaling of Summary Statistics

All of the algorithms introduced in this section, and indeed all ABC algorithms we will see in this thesis, are sensitive to the choice of ABC tuning parameters. In Example 2.1 we saw the impact that ϵ had on the accuracy of the ABC posterior and the computational cost of ABC. We have also discussed the importance of selecting 'good' summary statistics, and have seen, in Section 2.5.2, examples of how one can select summary statistics for ABC. For now, we assume that a *sensible* choice of summary statistics and tolerance has been made, and we consider the impact of the distance metric $\rho(\cdot, \cdot)$ on ABC inference.

2.6.1 Distance Metrics in ABC

In ABC the distance metric is used to measure the discrepancy between observed and simulated summary statistics. In all ABC examples we have implemented so far, we selected $\rho(\cdot, \cdot)$ to be scaled Euclidean distance, meaning that

$$\rho(S(\boldsymbol{x}), S(\boldsymbol{y})) = \sqrt{\sum_{j=1}^{p} \frac{|S_j(\boldsymbol{x}) - S_j(\boldsymbol{y})|^2}{\sigma_j}} \tag{2.48}$$

where σ_j is the marginal sample standard deviation of the set of summary statistics $S_j(\cdot)$. Without such scaling, summary statistics on a large scale would dominate the distance metric, and those on small scales would not be considered equally in the computation of discrepancy. In Section 2.6.2 we look at methods for scaling summary statistics, however for now we assume that all summary statistics are on the same scale, meaning that $\sigma_i = \sigma_j$ for all $i, j \in 1, \dots d$. Furthermore, we assume that all summary statistics hold equal information about the parameters of interest.

Pritchard et al. [3] used the Chebyshev distance metric to measure discrepancy between summary statistics. Also known as the L_∞ metric, the Chebyshev distance between two sets of summary statistics, $S(\boldsymbol{x})$ and $S(\boldsymbol{y})$,

$$\max_j(|S_j(\boldsymbol{x}) - S_j(\boldsymbol{y})|). \tag{2.49}$$

In comparison to Euclidean distance, a Chebyshev distance function requires all marginal summary statistics lie within some distance ϵ of the marginal observed summary statistic. This is perhaps beneficial in situations where one is sure that the summary statistics are known to be informative for the parameters. However, one could imagine a situation in which the model for the data is poor, and cannot create a summary statistic closer than a distance δ for one of the statistics. Using the Chebyshev distance in this case would require the acceptance of data at a distance of at least δ for all the summary statistics, whereas use of a Euclidean distance would instead potentially allow that the other $p-1$ summary statistics were close to their corresponding observed summary statistic.

McKinley et al. [10] compared the use of distance metrics for inference on an epidemic model. They showed that, for sensible choices of distance metrics, there was little difference to inference.

2.6.2 Scaling of Summary Statistics

Briefly mentioned earlier in this section, the scaling of summary statistics also influences the ABC inference. Having summary statistics on differing scales can result in one statistic dominating the distance metric computation. One way to overcome this issue is to scale summary statistics.

Example 2.4. *To illustrate the impact of scaling on the ABC posterior distribution, we implement Rejection ABC (Algorithm 1) in two dimensions using the following prior distributions:*

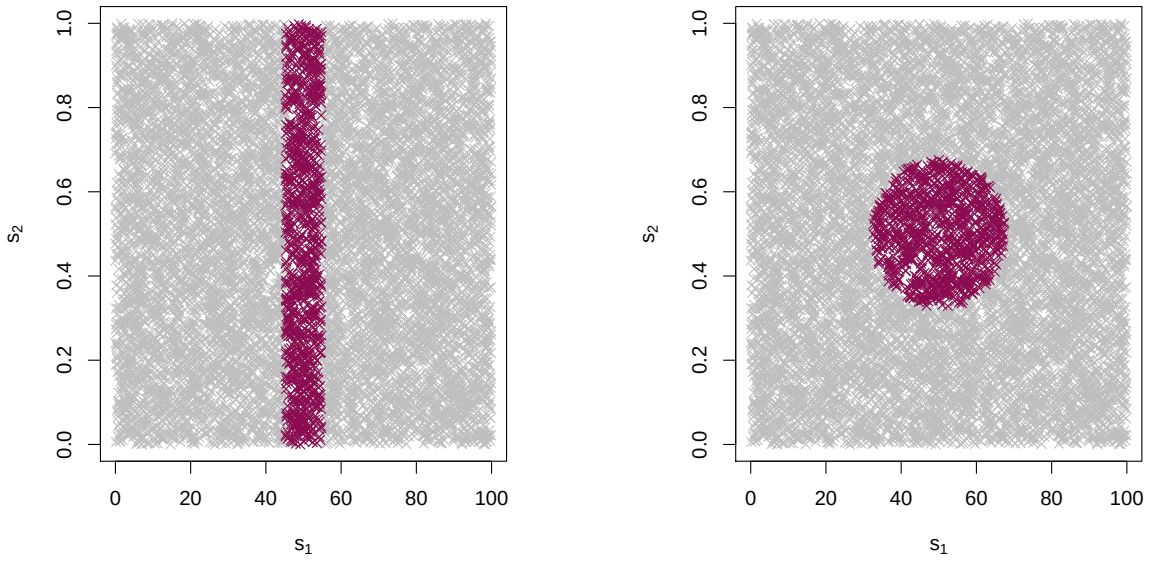
$$\theta_1 \sim \mathcal{U}(0, 1), \tag{2.50}$$

$$\theta_2 \sim \mathcal{U}(0, 100). \tag{2.51}$$

We set summary statistics to be equal to the sampled parameter values:

$$S(\boldsymbol{x}) = (\theta_1, \theta_2). \tag{2.52}$$

Observed summary statistics are given by $(0.5, 50)$ *and the posterior distribution is the closest* $1,000$ *points to the observed data. In implementation* (a) *we use the unscaled Euclidean distance as our distance metric. In implementation* (b) *we scale the summary statistics by their marginal standard deviation.*



(A) (B)

FIGURE 2.6: Posterior distributions (pink) obtained through Rejection ABC without scaling (left) and with scaling by standard deviation (right)

Figure 2.6 shows that the two scaling regimes lead to different posterior distributions. In implementation (a) *the distance from the observed data is dominated by the value of* S_1*. The scaling in implementation* (b) *means that the relative influence of* S_1 *and* S_2 *on the distance is more equal. This is reflected in the circular shape posterior distribution.*

We have previously considered scaling by the standard deviation of the summary statistics, based on a sample from the prior predictive distribution. Given a summary statistic function $S(\boldsymbol{x}) = S_1(\boldsymbol{x}), \ldots, S_p(\boldsymbol{x})$, and N simulated data sets $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, the ith marginal standard

deviation is given by

$$\sigma_i = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(S_i(\boldsymbol{x}_j) - \overline{S_i(\boldsymbol{x})})^2}, \tag{2.53}$$

where

$$\overline{S_i(\boldsymbol{x})} = \frac{1}{N}\sum_{j=1}^{N} S_i(\boldsymbol{x}_j). \tag{2.54}$$

This is the scaling used in most early ABC papers. However, Csillery et al. [23] implement in their R package a more robust scaling method, namely scaling by Median Absolute Deviation (MAD), given by

$$\textbf{median}_j(|S_i(\boldsymbol{x}_j) - \text{median}_i(S_i(\boldsymbol{x}))|). \tag{2.55}$$

By comparing equations 2.54 and 2.55, it is apparent that the MAD is less sensitive to outliers, because of the inclusion of the squared term in the computation of standard deviation.

In practice, when implementing ABC with multiple summary statistics it is not uncommon for one particular summary statistic to be constant across all the population, in certain regions of parameter space. We use the term 'monomorphic' to describe a summary statistic which takes the same value across all of the data samples we consider. We will see this in a population genetics example in Section 4.4. In such cases, both the standard deviation and the Median absolute deviation would be 0, and so one would not wish to scale by the reciprocal of this, since it will lead to infinite distances. Instead, we note that, if a summary statistic is monomorphic across the sample, we do not wish to scale that statistic, as we wish for its contribution to the distance metric to be 0. Equivalently we can think of this as converting the scaling factor from 0 to 1 in such cases, resulting in no scaling.

In Example 4.4 we scale by MAD from observed data, which differs from the implementation given in Equation 2.55. In the example considered, the summary statistics take very wide distributions, with a huge mass centred at one point. For example, in a sample of 1,000 from the prior predictive distribution it is not uncommon to see 980 summary statistics equal to "0", and 20 summary statistics of the order 10^4. Suppose now the observed summary statistic is 0. Scaling by the standard deviation would lead to the summary statistics which were originally of order 10^4 being scaled to an order of 10. This means that there is little contribution to the distance from these summary statistics, despite them being in the 98th percentile of distance

from the other summary statistics. Thus they are not penalised enough. However, computing the median absolute deviation from the observed summary statistics gives a value of 0, which, under the methodology above, we take to mean "no scaling needed", or equivalently, scale by factor 1. Thus the particles with large summary statistics in this dimension are unlikely to be favoured in any ABC algorithm, which is the desirable outcome. Note that in such a case the MAD from the mean, defined for a sample of data $X = x_1, \ldots, x_n$ as

$$MAD = \text{median}(x_i - \text{mean}(X)), \tag{2.56}$$

(as is the standard implementation in the R abc package [23]) is undesirable, as it still leads to a large scaling, resulting in the values being equally weighted.

2.6.3 Duality of Scaling and Distance metrics and Summary Statistics

There is a duality between the choice of distance metric and the choice of summary statistics and scaling selected. Using a scaled Euclidean distance is the same as scaling summary statistics, followed by using an unscaled Euclidean distance. To illustrate this, we consider the following example:

Example 2.5. *Let ρ denote the Euclidean distance. Then for summary statistic $S(\boldsymbol{x}) = S_1(\boldsymbol{x}), \ldots, S_p(\boldsymbol{x})$ and scaling factors σ_i, the scaled Euclidean distance between summaries $S(\boldsymbol{x})$ and $S(\boldsymbol{y})$ is*

$$\rho(S(\boldsymbol{x}), S(\boldsymbol{y})) = \sqrt{\sum_{i=1}^{p} \left(\frac{S_i(\boldsymbol{x}) - S_i(\boldsymbol{y})}{\sigma_i} \right)^2}. \tag{2.57}$$

Now if we use the following (trivial) summary statistics S_i/σ_i in conjunction with the Euclidean distance metric, the distance is given by

$$\sqrt{\sum_{i=1}^{p} \left(\frac{S_i(\boldsymbol{x})}{\sigma_i} - \frac{S_i(\boldsymbol{y})}{\sigma_i} \right)^2} = \sqrt{\sum_{i=1}^{p} \left(\frac{S_i(\boldsymbol{x}) - S_i(\boldsymbol{y})}{\sigma_i} \right)^2}, \tag{2.58}$$

which is identical to the right hand side of Equation (2.57).

This trivial example illustrates the interdependence between summary statistics and scaling.

One intuitive way to think of the scaling of summary statistics or selection of distance metric is in terms of defining an ellipsoidal critical region on the space of summary statistics, and accepting any points which fall within that region. Specifically, for simulated summary statistics $S(\boldsymbol{x})$, and observed summary statistics $S(\boldsymbol{y})$, where $S(\cdot) \in \mathbb{R}^p$, the critical region is given by the set of $\boldsymbol{x}$ such that

$$(S(\boldsymbol{x}) - S(\boldsymbol{y}))^T A(S(\boldsymbol{x}) - S(\boldsymbol{y})) < K, \tag{2.59}$$

where $A \in \mathbb{R}^{p \times p}$ is a positive definite matrix, and K is some real number. The left hand side of Equation (2.59) corresponds to the posterior loss, or scaled mean squared error, and is seen in Fearnhead and Prangle [22]. It is possible to scale both $S(\cdot)$ and A in Equation 2.59 such that the critical region remains the same, despite this change of scale.

Traditionally, distance metrics, scaling and summary statistics are selected at the start of the process. Prangle [24] noted that updating the scaling of the summary statistics at each iteration is equivalent to updating the distance metric, and indeed using Euclidean distance on summary statistics which have been scaled by their standard deviations is equivalent to using weighted Euclidean distance, with weights given by the reciprocal of the standard deviations.

2.7 Post Processing Techniques in ABC

Post processing techniques consider how one can improve the accuracy of the ABC posterior distribution, after the sample has been obtained. Beaumont et al. [6] proposed the use of local linear regression to correct posterior samples which were of distance ϵ from the observed summary statistics $S(\boldsymbol{y})$.

Given an ABC posterior sample, $\boldsymbol{\theta}_i$, for $i \in 1, \ldots, N$, and corresponding summary statistics $S(\boldsymbol{x}_i)$, the regression-based approach of Beaumont et al. [6] uses local linear regression to estimate $\mathbb{E}(\boldsymbol{\theta}|S(\boldsymbol{x}))$. This enables a 'corrected' set of parameters $\boldsymbol{\theta}_i^*$ to be computed, where

$$\boldsymbol{\theta}_i^* = \boldsymbol{\theta}_i - \hat{\mathbb{E}}(\boldsymbol{\theta}|S(\boldsymbol{x}_i)) + \hat{\mathbb{E}}(\boldsymbol{\theta}|S(\boldsymbol{y})). \tag{2.60}$$

Blum and Francois [25] further developed this model to be able to deal with a Heteroscedastic model (where variance changes). In the examples considered in Chapter 4, we implement the Regression Correction method of [6], and show that it greatly improves the inference, compared with the standard Rejection ABC.

2.8 Outline of Thesis

In Chapter three we explain the SMC-ABC algorithm of Del Moral et al. [1], and propose adaptations to the algorithm with the aim of increasing the efficiency of the algorithm by reducing the computational cost. In this chapter we also propose an iterative summary statistic selection method, which requires little user tuning.

Chapter four applies the adapted SMC-ABC algorithm with iterative summary statistic selection to a range of models. We show that the algorithms developed in chapter three can lead to improved posterior inference.

Chapter five discusses algorithms which use samples from the model to give an approximation of the likelihood function. This empirical approximation is used in place of the unknown exact likelihood function. We propose the use of k-Nearest Neighbour density estimation to approximate the joint density function and we show that, for preliminary examples, incorporating this density estimation into sequential Monte Carlo algorithms leads to good posterior inference.

Chapter 3

Methods in SMC-ABC

As we saw in Chapter 2, ABC is sensitive to the choice of summary statistics. The optimal summary statistics for ABC inference are those that capture all of the information about the parameters of interest that are held in the data, and are known as sufficient statistics (Equation (2.6)). However, these statistics are often not available to us, so it is common to use summary statistic selection methods to identify *good* summary statistics. Furthermore, as stated by the Pitman–Koopman–Darmois Theorem, for distributions of fixed domain, it is only in the exponential family that the dimension of sufficient statistics is bounded as the sample size is increased. In Section 2.5.2 we compared existing summary statistic selection methods for ABC. Of the methods considered, we deem the automatic summary statistic selection method of Fearnhead and Prangle [22] to be the most easily implementable, since little user input and tuning is required. The method hinges on estimating the posterior mean, based on a preliminary (smaller) run of ABC. However, in high dimensional examples, the number of samples required to get an accurate estimate of the posterior mean is likely to be large, and thus the method becomes inefficient, or results in biased inference. This is particularly the case when the relationship between the parameters and summary statistics is non-linear.

In this chapter we incorporate summary statistic selection into the SMC-ABC algorithm of Del Moral et al. [1]. The iterative summary statistic selection uses the SMC-ABC posterior estimate at the previous iteration to estimate the posterior mean, which is then used as the

summary statistics which are alive at the current algorithm iteration, as well as summary statistics of particles which were previously alive. This results in a better estimate of the posterior mean.

We also propose two additional amendments to the algorithm of Del Moral et al. [1] that make the algorithm more efficient, and reduce the required amount of user tuning. The first is an alternative stopping rule for the algorithm, which is less dependent on the scale of the summary statistics. The second is a method for reducing the number of model simulations required, by splitting the Metropolis-Hastings acceptance ratio (2.25) into two separate ratios, one of which can be evaluated before data is simulated. Thus some trial parameter values are rejected without simulating from the model, thereby reducing the computational cost.

3.1 SMC-ABC, Del Moral et al. [1]

We now present the SMC-ABC algorithm of Del Moral et al. [1]. The pseudo-code is given in Algorithm 4. Here we outline the main processes:

1. Draw an initial sample of N particles from the joint distribution $p(\boldsymbol{\theta}, S(\boldsymbol{x}))$, and compute their distance from the observed summary statistics $S(\boldsymbol{y})$.

2. Retain the $100 \times \alpha\%$ of these particles which are closest to the observed summary statistic $S(\boldsymbol{y})$, where $\alpha \in (0, 1)$.

3. If the *effective sample size* [26] of the current set of retained particles has dropped below a pre-defined threshold, resample N particles from the current set of live particles. The effective sample size of a set of N particles $(\boldsymbol{\theta}_i, S(\boldsymbol{x}_i))$ with associated weights w_i is given by
$$ess(w_i; \boldsymbol{\theta}_i, S(\boldsymbol{x}_i)) = \left(\sum_{i=1}^{N} (w_i^2) \right)^{-1}. \tag{3.1}$$

4. Perturb the live particles using a Metropolis-Hastings kernel.

5. Repeat steps 2 to 5 until all live summary statistics sit within pre-selected distance, ϵ_T, of the observed summary statistics $S(\boldsymbol{y})$.

Algorithm 4 SMC-ABC, Del Moral et al. [1]

1: Set $t = 0$, $\epsilon_0 = \infty$, select ϵ_T and fix $\alpha \in (0, 1)$.

2: For $j = 1, \ldots, N$, sample

$$\boldsymbol{\theta}_j^{(t)} \sim \pi(\cdot), \text{ and } \boldsymbol{x}_j^{(t)} \sim p(\cdot|\boldsymbol{\theta}_j^{(t)}). \tag{3.2}$$

Set

$$w_j = N^{-1}. \tag{3.3}$$

3: For $j \in 1, \ldots, N$, compute $d_j = \rho(S(\boldsymbol{x}_j^{(t)}), S(\boldsymbol{y}))$.

4: Set $t = t + 1$. Compute ϵ_t such that

$$\alpha \sum_{j=1}^{N} \mathbb{I}\{(d_j < \epsilon_{t-1})\} = \sum_{j=1}^{N} \mathbb{I}\{(d_j < \epsilon_t)\}. \tag{3.4}$$

For all $j \in (1, \ldots, N)$ such that $d_j > \epsilon_t$, set $w_j = 0$.

5: Renormalize the weights such that

$$\sum_{j=1}^{N} w_j = 1. \tag{3.5}$$

6: If $ess\{(w_j; \boldsymbol{\theta}_j^{(t-1)}, S(\boldsymbol{x}_j^{(t-1)})\} < N/2$, resample N particles from the set of

$$(\boldsymbol{\theta}_j^{(t-1)}, S(\boldsymbol{x}_j)^{(t-1)}) \text{ for which } w_j > 0. \tag{3.6}$$

7: For all j such that $w_j > 0$, sample

$$\boldsymbol{\theta}'_j \sim q_t(\cdot|\boldsymbol{\theta}_j^{(t-1)}) \text{ and } \boldsymbol{x}'_j \sim p(\cdot|\boldsymbol{\theta}'_j). \tag{3.7}$$

Compute $d'_j = \rho(S(\boldsymbol{x}'_j), S(\boldsymbol{y}))$ and set

$$\alpha_j = min\left(1, \frac{\pi(\boldsymbol{\theta}'_j)}{\pi(\boldsymbol{\theta}_j^{(t-1)})} \frac{q_t(\boldsymbol{\theta}_j^{(t-1)}|\boldsymbol{\theta}'_j)}{q_t(\boldsymbol{\theta}'_j|\boldsymbol{\theta}_j^{(t-1)})} \mathbb{I}\{(d'_j < \epsilon_t)\}\right). \tag{3.8}$$

Simulate $u_j \sim \mathcal{U}(0, 1)$.

If $\alpha_j \geq u_j$, set $\boldsymbol{\theta}_j^{(t)} = \boldsymbol{\theta}_j^{(t-1)}$. Else, set $\boldsymbol{\theta}_j^{(t)} = \boldsymbol{\theta}'_j$ and $d_j = d'_j$.

8: While $\epsilon_t \geq \epsilon_T$, return to Step 4.

Del Moral et al. [1] suggest that $q_t(\cdot|\boldsymbol{\theta}_j^{(t-1)})$ a Normal Random Walk proposal is an appropriate choice of proposal distribution for the Metropolis-Hastings kernel, with variance given by twice the empirical variance of the set of $\boldsymbol{\theta}_j^{(t-1)}$. In practice, we found that the algorithm mixed better with a smaller choice of variance.

As with all ABC algorithms discussed in earlier chapters, Algorithm 4 can be implemented when summary statistics are discrete or continuous. In Section 4.3 we apply Algorithm 4 to an example which uses a mixture of both continuous and discrete summaries.

Example 3.1. *We use the ABC-SMC algorithm of Del Moral et al. [1] to obtain ABC posterior distributions for the Bivariate Gaussian example, first seen in Example 2.1. By running the algorithm at a range of values of α, we investigate its impact on the ABC posterior distribution. We select $\epsilon_T = 0.1$.*

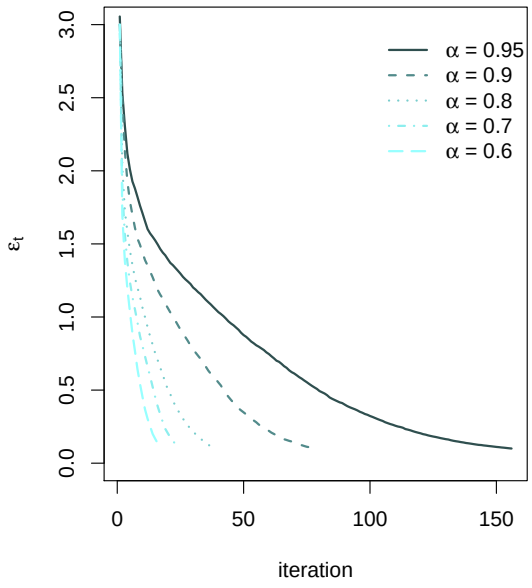


FIGURE 3.1: Change in tolerance, ϵ, over iterations of the SMC-ABC algorithm, for a range of values of α. The algorithm is implemented until it reaches a tolerance $\epsilon_T = 0.1$

Figure 3.1 shows that as α increases, the number of iterations needed to reach a tolerance of $\alpha = 0.1$ decreases. This suggests that, in terms of computational cost, selecting a small value of α is beneficial. To some extent this is the case: Table 3.1 shows that there is little difference between the MSE of the posterior distributions for varying α. However, the final column of Table 3.1 shows that, for smaller values of α, the posterior distributions contain fewer unique

parameter values. This is due to an increase in the frequency of resampling steps for smaller α. Thus, more parameters are duplicated. To further illustrate this behaviour, we implemented the algorithm at $\alpha = 0.5$, with the resultant ABC posterior distribution containing only one particle, repeated 1,000 times.

α	μ_1	μ_2	σ_1^2,	$\rho\sigma_1\sigma_2$,	σ_2^2	iterations	unique values
0.6	0.0861	0.1065	0.1655	**0.0518**	0.5349	18	122
0.7	**0.0493**	0.0993	**0.1442**	0.0840	0.1197	25	188
0.8	0.0716	0.0749	0.1979	0.2271	0.1550	38	207
0.9	0.0504	**0.0556**	0.1904	0.1859	**0.1185**	79	326
0.95	0.0852	0.0759	0.2726	0.1724	0.2639	158	372

TABLE 3.1: Mean squared error of the ABC posteriors obtained through Del Moral et al. [1]'s SMC-ABC, for varying values of α. The smallest value in each column is given in bold.

We now present three amendments to the algorithm of Del Moral et al. [1]. The first is a more robust stopping rule. The second shows that, by splitting the Metropolis-Hastings acceptance ratio into two accept or reject steps, an implementation of the algorithm which requires fewer simulations from the model can be put in place. Finally, we give a framework for incorporating automatic summary statistic selection into SMC-ABC.

3.2 Amending the stopping rule

SMC-ABC, as given in Algorithm 4, terminates once the tolerance ϵ_t becomes no greater than some user-specified value, ϵ_T. In practice, selecting a *sensible* value for ϵ_T is not straightforward because it depends on both the choice of distance metric, and inherent properties of the summary statistics such as their scale, dimension, and physical interpretation. Furthermore, under model misspecification, (when the observed summary statistics do not fit the real data,) it may not be possible for the tolerance to tend to 0, since there is some minimal non-zero distance between the observed summary statistics and the summaries of data simulated under the model.

One way to select an appropriate terminating tolerance value would be first to carry out a preliminary run of Rejection ABC. This would enable the user to select a tolerance that corresponded to being *close* to the observed summary statistics.

In practice, one wishes to implement SMC-ABC in an automatic fashion, with as little user tuning as possible. Because of this, we suggest the following stopping rule (Algorithm 5):

Algorithm 5 Alternative Stopping Rule for SMC-ABC

8: If $\alpha_j = 0$ for all j, terminate the algorithm.

This alternative stopping rule causes the algorithm to terminate if all summary statistics generated in step 7. lie further than distance ϵ_t from the observed summary statistics. Formally, this means that, for all j such that $w_j > 0$, $d'_j \geq \epsilon_t$. Using this stopping rule means that the algorithm terminates only when it is unable to propose *closer* particle values. Table 3.2 gives the final value of the tolerance, when the algorithm is run using the stopping rule above, for the different values of α. It shows that selecting a larger value of α leads to the algorithm terminating at a smaller value of ϵ_T. Hence, for higher α we can reach ABC posterior distributions that better approximate the true posterior distribution.

α	ϵ_{final}
0.6	0.118
0.7	0.053
0.8	0.036
0.9	0.033
0.95	0.017

TABLE 3.2: Final tolerance when SMC-ABC algorithm of Del Moral et al. [1] is implemented with the stopping rule given in Algorithm 5.

3.3 Splitting the Metropolis-Hastings acceptance ratio

The SMC-ABC algorithm of Del Moral et al. [1] as well as the MCMC-ABC algorithm of Marjoram et al. [9] both require the evaluation of the ABC representation of the Metropolis-Hastings

acceptance probability, given by

$$\min\left(1, \frac{\pi(\boldsymbol{\theta}')q(\boldsymbol{\theta}_t|\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}_t)q(\boldsymbol{\theta}'|\boldsymbol{\theta}_t)}\mathbb{I}\{\rho(S(\boldsymbol{x}), S(\boldsymbol{y})) < \epsilon\}\right). \tag{3.9}$$

In the case where the likelihood is tractable, the equivalent Metropolis-Hastings acceptance probability is given by

$$\min\left(1, \frac{p(\boldsymbol{y}|\boldsymbol{\theta}')}{p(\boldsymbol{y}|\boldsymbol{\theta})}\frac{\pi(\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta})}\frac{q(\boldsymbol{\theta}|\boldsymbol{\theta}')}{q(\boldsymbol{\theta}'|\boldsymbol{\theta})}\right). \tag{3.10}$$

Peskun [27] noted that the second term in Equation (3.10), known as the Metropolis-Hastings ratio, can be split into two terms, one of which depends only on $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$, the second given by the ratio of likelihoods:

$$\frac{p(\boldsymbol{y}|\boldsymbol{\theta}')}{p(\boldsymbol{y}|\boldsymbol{\theta})} \text{ and } \frac{\pi(\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta})}\frac{q(\boldsymbol{\theta}|\boldsymbol{\theta}')}{q(\boldsymbol{\theta}'|\boldsymbol{\theta})}. \tag{3.11}$$

From this, it follows that the Metropolis-Hastings acceptance ratio can be split into two steps as follows:

Algorithm 6 Two-Stage Metropolis-Hastings Acceptance Decision

Let $\boldsymbol{\theta}' \sim q(\cdot|\boldsymbol{\theta}^{(t-1)})$, let $\pi(\cdot)$ denote the prior distribution and $p(\cdot|\theta)$ be the likelihood function.

1: With probability

$$\min\left(1, \frac{\pi(\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}^{(t-1)})}\frac{q(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}')}{q(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(t-1)})}\right), \tag{3.12}$$

proceed to step 2, else set $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)}$ and stop.

2: With probability

$$\min\left(1, \frac{p(\boldsymbol{y}|\boldsymbol{\theta}')}{p(\boldsymbol{y}|\boldsymbol{\theta}^{(t-1)})}\right), \tag{3.13}$$

set $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}'$, else set $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)}$.

Peskun [27] states that the two-stage Metropolis-Hastings acceptance step, as given in Algorithm 6 is less efficient than the standard Metropolis-Hastings ratio, as given in Equation (3.10), since

$$\min\left(1, \frac{p(\boldsymbol{y}|\boldsymbol{\theta}')}{p(\boldsymbol{y}|\boldsymbol{\theta})}\frac{\pi(\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta})}\frac{q(\boldsymbol{\theta}|\boldsymbol{\theta}')}{q(\boldsymbol{\theta}'|\boldsymbol{\theta})}\right) \geq \min\left(1, \frac{\pi(\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta})}\frac{q(\boldsymbol{\theta}|\boldsymbol{\theta}')}{q(\boldsymbol{\theta}'|\boldsymbol{\theta})}\right)\min\left(1, \frac{p(y|\boldsymbol{\theta}')}{p(y|\boldsymbol{\theta})}\right). \tag{3.14}$$

When we return to the ABC setting, where the likelihood function is replaced by an approximation, the second term in Equation (3.14) becomes

$$\min\left(1, \mathbb{I}\{\rho(S(\boldsymbol{x}), S(\boldsymbol{y})) < \epsilon\}\right) = \mathbb{I}\{\rho(S(\boldsymbol{x}), S(\boldsymbol{y})) < \epsilon\}. \tag{3.15}$$

Because of the nature of the indicator function $\mathbb{I}\{\cdot\}$, this minimum can only take two values, 0 or 1. This means that, in the ABC setting we have equality in Equation (3.14), and thus no efficiency is lost by splitting the acceptance ratio into two steps. Furthermore, the value of

$$\min\left(1, \frac{\pi(\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta})}\frac{q(\boldsymbol{\theta}|\boldsymbol{\theta}')}{q(\boldsymbol{\theta}'|\boldsymbol{\theta})}\right) \tag{3.16}$$

can be evaluated without to simulating data from the model. It is this model simulation which commonly dominates the computational cost of an ABC algorithm, so this is extremely beneficial as some components of the Markov Chain can now be updated, without simulating from the model. We present this efficient Two-Stage Metropolis-Hastings Acceptance Decision for ABC in Algorithm 7. These steps should be implemented in the place of step 7 of Algorithm 4.

Algorithm 7 Two-Stage Metropolis-Hastings Acceptance Decision for ABC

For all j such that $w_j > 0$:

1: Sample $\boldsymbol{\theta}'_j \sim q_t(\cdot|\boldsymbol{\theta}_j^{(t-1)})$.

2: With probability

$$\min\left(1, \frac{\pi(\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta})}\frac{q(\boldsymbol{\theta}|\boldsymbol{\theta}')}{q(\boldsymbol{\theta}'|\boldsymbol{\theta})}\right) \tag{3.17}$$

proceed to step 2, else set $\boldsymbol{\theta}_j^{(t)} = \boldsymbol{\theta}_j^{(t-1)}$ and stop.

3: Simulate $\boldsymbol{x}'_j \sim p(\cdot|\boldsymbol{\theta}'_j)$ and compute $d'_j = \rho(S(\boldsymbol{x}'_j), S(\boldsymbol{y}))$.

4: With probability

$$\mathbb{I}\{\rho(S(\boldsymbol{x}'_j), S(\boldsymbol{y})) < \epsilon\} \tag{3.18}$$

set $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}'$, else set $\boldsymbol{\theta}_j^{(t)} = \boldsymbol{\theta}_j^{(t-1)}$.

In Chapter 4 we go on to show that, for a range of applications, a great computational saving is made by implementing this alternative acceptance decision.

Algorithm 7 is in the same vein as the early-rejection ABC methods of Everitt and Rowińska [28] and Picchini and Forman [29]. Everitt and Rowińska [28] propose an MCMC-ABC algorithm which assumes there exists two data simulators: the first being accurate but computationally expensive; the second being faster but less accurate. Initial, 'early' acceptance decisions are based on simulations from the second simulator. Given an early acceptance, a final accept or reject decision is then made based on data simulated from the first, expensive simulator. Their algorithm removes the need to carry out expensive simulations at some parameter values which have negligible posterior mass. A more general example of this is presented in Picchini and Forman [29], where the observed data is assumed to be a time series. Initial acceptance decisions are based on summaries of the sub-sampled time series, thus there is no need to simulate the full trajectory. Given an acceptance at the initial stage, the remainder of the time series trajectory is simulated, conditional upon the sub-sampled series, and a final acceptance decision is made.

3.4 Automatic Summary Statistic Selection

In Chapter 2 we saw that the choice of summary statistics greatly influences the accuracy of the ABC posterior distribution. We now amend the algorithm of Del Moral et al. [1] so that it incorporates automatic summary statistic selection, removing the need for user tuning or preliminary simulations on which to base the summary statistic selection method. We begin by discussing the automatic summary statistic selection method of Fearnhead and Prangle [22]. We go on to describe how we localise the regression iteratively within the SMC-ABC algorithm.

3.4.1 Estimating the Posterior Mean

Fearnhead and Prangle [22] showed that the optimal summary statistic for ABC, in terms of minimising mean squared loss, is the posterior mean, which we denote $\mathbb{E}(\boldsymbol{\theta}|S = S(\boldsymbol{y}))$. The posterior mean is unknown, since we do not have access to the posterior distribution, or an estimate of it, prior to implementing ABC. It is suggested in Fearnhead and Prangle [22] that one estimates the posterior mean through linear regression. The following regression model is

suggested by the authors:

$$\boldsymbol{\theta}_j^T = \mathbb{E}(\boldsymbol{\theta}|S = S(\boldsymbol{x})) + \boldsymbol{\gamma}_j^T = S(\boldsymbol{x}_j)^T\beta + \boldsymbol{\gamma}_j^T, \tag{3.19}$$

where $\boldsymbol{\theta}_j \in \mathbb{R}^{p\times 1}$ and $S(\boldsymbol{x}_j) \in \mathbb{R}^{q+1\times 1}$ are given by

$$\boldsymbol{\theta}_j = \begin{bmatrix} \theta_{1,j} \\ . \\ . \\ \theta_{p,j} \end{bmatrix}, \quad S(\boldsymbol{x}_j) = \begin{bmatrix} 1 \\ S_1(\boldsymbol{x}_j) \\ . \\ . \\ S_q(\boldsymbol{x}_j) \end{bmatrix}, \tag{3.20}$$

$\beta \in \mathbb{R}^{((q+1)\times p)}$ is a matrix of regression coefficients and $\boldsymbol{\gamma}_j \in \mathbb{R}^{p\times 1}$ is white noise.

By substituting the unknown matrix β with its least squares estimate $\hat{\beta}$, we are able to obtain an estimate of $\mathbb{E}(\boldsymbol{\theta}|S = S(\boldsymbol{x}_j))$. This is used as the summary statistics for the model, and thus the distance between simulated summary statistics $S(\boldsymbol{x})$ and observed summary statistics $S(\boldsymbol{y})$ is given by

$$\begin{aligned} \rho(S(\boldsymbol{x}), S(\boldsymbol{y})) &= \|S(\boldsymbol{x})^T\hat{\beta} - S(\boldsymbol{y})^T\hat{\beta}\| \\ &= \|(S(\boldsymbol{x}) - S(\boldsymbol{y}))^T\hat{\beta}\|. \end{aligned} \tag{3.21}$$

Based on a set of m samples $(\boldsymbol{\theta}_j, S(\boldsymbol{x}_j))$, for $j \in 1, \ldots m$, where $S_{\boldsymbol{X}}$ is the $(q+1) \times m$ dimensional matrix with jth column given by $S(\boldsymbol{x}_j)$, and $\boldsymbol{\theta}_{[i,]}$ is the row vector containing elements $\theta_{i,1}, \theta_{i,2}, \ldots, \theta_{i,m}$ the least squares estimate of the matrix β in Equation (3.19) is given by

$$\hat{\beta} = \left[\hat{\beta}_{[,1]}\hat{\beta}_{[,2]} \cdots \hat{\beta}_{[,p]}\right], \tag{3.22}$$

where $\hat{\beta}_{[,i]}$ is a $(q \times 1)$ column vector, with components given by

$$\hat{\beta}_{[,i]} = (S_{\boldsymbol{X}}S_{\boldsymbol{X}}^T)^{-1}S_{\boldsymbol{X}}\boldsymbol{\theta}_{[i,]}, \tag{3.23}$$

3.4.2 Updating the distance metric

From Equation (3.21) it follows that transforming the summary statistics from $S(\boldsymbol{x})$ to $S(\boldsymbol{x})\hat{\beta}$ is equivalent to using summary statistics $S(\boldsymbol{x})$ with an alternative distance metric, which we denote $\rho_{\hat{\beta}}(S(\boldsymbol{x}), S(\boldsymbol{y}))$, where

$$\rho_{\hat{\beta}}(S(\boldsymbol{x}), S(\boldsymbol{y})) = \|(S(\boldsymbol{x}) - S(\boldsymbol{y}))^T \hat{\beta}\|. \tag{3.24}$$

For the remainder of this chapter we consider the automatic summary statistic selection method of Fearnhead and Prangle [22] as a way of transforming the distance metric, rather than transforming the summary statistics.

The notion of updating the distance metric within ABC has been previously introduced by Prangle [24], which proposed that the distance metric should be updated within the SMC-PMC Algorithm of Beaumont et al. [15]. The update of the distance metric in that case ensures that the weighting of the summary statistics within at iteration t of the algorithm is equal to the marginal standard deviation of the summary statistics which approximate the ABC posterior distribution at that time, thereby accounting for the changing scales of the summary statistics at each iteration. This is equivalent to changing the diagonal elements of matrix $\hat{\beta}$. In the next section we propose a methodology to update $\hat{\beta}$ as a whole, rather than just altering the diagonal elements.

3.4.3 Summary Statistic Selection within SMC-ABC

In the semi-automatic summary statistic selection method of Fearnhead and Prangle [22], the authors estimate the unknown regression matrix β through ordinary least squares regression.

However, suppose that the prior distribution has a wider support than the posterior distribution, and that there is a non-linear relationship between the parameters and the summary statistics. In this case, unless the linear regression is computed in a local region, it is likely that the resultant estimate of $\hat{\beta}$ will be poor. To this end, Fearnhead and Prangle [22] propose that a preliminary run of ABC is carried out, in order to determine regions of the parameter space that

have *non-negligible posterior mass*. The regression can then be carried out using samples from this region to obtain a value of $\hat{\beta}$, and a final ABC implementation can be run, using distance metric $\rho_{\hat{\beta}}(\cdot, \cdot)$.

3.4.4 Weighted Least Squares Regression

Weighted least squares regression is commonly used when one has a data set which contains data of variable quality as it provides a method of favouring some points in the sample set over others. For every particle $\boldsymbol{\theta}_i, S(\boldsymbol{x}_i)$ in our preliminary set, on which we wish to base our linear regression, suppose we also have a weight ω_i, associate do the particle. Then weighted least squares finds a matrix $\hat{\beta} = \left[\hat{\beta}_{[,1]}\hat{\beta}_{[,2]} \cdots \hat{\beta}_{[,p]}\right]$, where column $\hat{\beta}[, i]$ is given by

$$\hat{\beta}_{[,i]} = (S_{\boldsymbol{X}}^T W S_{\boldsymbol{X}})^{-1} S_{\boldsymbol{X}}^T W \boldsymbol{\theta}_{[i,]}, \tag{3.25}$$

where

$$W = diag(\omega_1, \ldots, \omega_m). \tag{3.26}$$

Altering the weights matrix W leads to different estimates of β. For example, in the case where $W = diag(1, \ldots, 1)$ Equation (3.25) is the same as Equation (3.23), and thus all particles $(\theta_j, S(x_j))$ in the training set have equal influence over the value of $\hat{\beta}$.

For our implementation, we wish to select the weights matrix W such that the points close to the observed summary statistics have a greater influence on the estimate of β than those that sit far from the observed summary statistics. For this reason, we use kernel weights for the diagonal weights matrix, W, as is done in locally linear least squares regression (Ruppert and Wand [30], Beaumont et al. [6].)

We select our weights matrix to W be of the form

$$W = diag\left(K_H(S(\boldsymbol{x}_1) - S(\boldsymbol{y})), \ldots, K_H(S(\boldsymbol{x}_n) - S(\boldsymbol{y}))\right), \tag{3.27}$$

where $K_H(u)$ is a kernel matrix. We select a K_H to be a Gaussian kernel, since we wish for exponential decay of the weights as you move away from the observed summary statistics $S(\boldsymbol{y})$.

The Gaussian kernel is centred at the observed summary statistics, $S(\boldsymbol{y})$, and the standard deviation of the Gaussian is selected marginally at each iteration of the algorithm.

The standard deviation of the Gaussian kernel in dimension k, σ_k, at time t is determined based on the set SMC-ABC of particles with non-zero weights at time t of the SMC algorithm. We select σ_k to be

$$\sigma_k = \frac{1}{3} \max_{\{S(\boldsymbol{x_j}):w_j>0\}} \|S_k(\boldsymbol{x}_j) - S_k(\boldsymbol{y})\| \tag{3.28}$$

where $S_k(\boldsymbol{x})$ denotes the kth component of summary statistic vector $S(\boldsymbol{x})$. Equation (3.28) is motivated by the Gaussian rule of thumb, which states that, for Gaussian random variables 99.7% of sampled points lie within three standard deviations of the mean. In practice we have found that such a choice of standard deviation works well.

Thus for any point $(\theta, S(\boldsymbol{x}) = S_1(\boldsymbol{x}), S_2(\boldsymbol{x}), \ldots, S_q(\boldsymbol{x}))$, we can compute the corresponding kernel weight $K_H(S(\boldsymbol{x}) - S(\boldsymbol{y}))$ by

$$K_H(S(\boldsymbol{x}) - S(\boldsymbol{y})) \propto \prod_{j=1}^{q} \exp\{-(2\sigma_j^2)^{-1}(S_j(\boldsymbol{x}) - S_j(\boldsymbol{y}))\}. \tag{3.29}$$

With these kernel weights we are able to compute an estimate of β which satisfies equation (3.25). We denote this estimate $\widehat{\beta}_t$ since it is dependent on the particles that are alive at time t. We can therefore compute the distance between observed summary statistics and simulated summary statistics with distance metric $\rho_{\widehat{\beta}_t}$, where

$$\rho_{\widehat{\beta}_t}(S(\boldsymbol{x}), S(\boldsymbol{y})) = \|(S(\boldsymbol{x}) - S(\boldsymbol{y}))^T \widehat{\beta}_t\|. \tag{3.30}$$

Although the computation of the standard deviations of the kernels, given in Equation (3.28), depends only on 'live' particles at the current time, we use additional particles which have been simulated at previous stages of the algorithm to train the regression model. At iteration t of the algorithm we select the training set used in Equation (3.25) to be all those particles which, at any time step $\tau < t$ have been accepted in the Metropolis-Hastings step of the SMC-ABC algorithm. Our motivation for using such particles, is that any such particle $(\theta, S(x))$ is drawn from the joint distribution of θ, S, conditional upon $\rho_{\beta_\tau}(S(x), S(y)) < \epsilon_\tau$ for some $\tau < T$. Thus

conditional on $\rho_{\beta_\tau}(S(x), S(y)) < \epsilon_\tau$, and assuming local linearity, β can be used to provide an unbiased estimate of the posterior mean, following Fearnhead and Prangle [22].

3.4.4.1 Impact of updating the distance metric on ϵ

In the standard implementation of SMC-ABC (Algorithm 4), as well as PMC-ABC (Algorithm 3) the tolerance ϵ_t decreases over iterations of the algorithm. However, with the introduction of an adaptive distance metric, as given in Equation (3.30), there is no longer certainty that the tolerance will decrease at each iteration. As such, we think of the update of tolerance in terms of number of alive particles. The next tolerance value is selected such that $100 \times \alpha\%$ of the particles which were alive at time $t-1$ are alive at time t.

In each pass through Algorithm 8, the tolerance ϵ_t is computed twice. In step 3, ϵ_t is determined using the notion described above, so that a percentage of those particles which were alive at time $t-1$ are alive at time t. However, ϵ_t must be recomputed in step 7 because the distance metric is updated in step 6. By updating the distance metric, the distance between the observed summary statistics and the simulated summary statistics may have changed, so ϵ_t is recomputed to ensure that all particles which were alive, prior to the distance metric being updated, are still alive.

3.4.5 Convergence

We have not explored results on convergence of Algorithm 8. Certainly this would be ideal, and should be undertaken in further work. In Chapter 4 we see that, for the examples considered, the algorithm performs well, and appears to converge to the posterior distribution.

3.4.6 Pseudo Code: Auto-SS SMC-ABC

In Algorithm 8 we give the pseudo-code for the adapted algorithm of Del Moral et al. [1] which contains three amendments discussed in this chapter. From here on we refer to this Algorithm as Auto-SS SMC-ABC.

Algorithm 8 Auto-SS SMC-ABC

Set t=0. Fix $\alpha \in (0,1)$. Let $q(\cdot|\boldsymbol{\theta})$ be a proposal distribution for $\boldsymbol{\theta}$ and let $\rho_0(a,b)$ be the Euclidean distance between a and b.

1: For $j = 1, \ldots, N$ sample

$$\boldsymbol{\theta}_j^{(t)} \sim \pi(\cdot), \text{ and } \boldsymbol{x}_j^{(t)} \sim p(\cdot|\boldsymbol{\theta}_j^{(t)}). \tag{3.31}$$

Set

$$w_j = N^{-1}. \tag{3.32}$$

2: For $j \in 1, \ldots, N$ compute $d_j = \rho_0(S(\boldsymbol{x}_j^{(t)}), S(\boldsymbol{y}))$

3: Set t=t+1. Compute ϵ_t such that

$$\alpha \sum_{j=1}^{N} \mathbb{I}\{w_j > 0\} = \sum_{j=1}^{N} \mathbb{I}\{(d_j < \epsilon_t)\}. \tag{3.33}$$

For all j $\in (1, \ldots, N)$ such that $d_j \geq \epsilon_t$, set $w_j = 0$.

4: Renormalize the weights such that

$$\sum_{j=1}^{N} w_j = 1. \tag{3.34}$$

5: If $ess\{(w_j; \boldsymbol{\theta}_j^{(t-1)}, S(\boldsymbol{x}_j^{(t-1)})\} < N/2$, resample N particles from the set of

$$(\boldsymbol{\theta}_j^{(t-1)}, S(\boldsymbol{x}_j^{(t-1)})) \text{ for which } w_j > 0. \tag{3.35}$$

6: Using Equations (3.25), (3.28) and (3.29), compute $\widehat{\beta}_t$.

7: For all j such that $w_j > 0$, set $d_j = \rho_{\widehat{\beta}_t}(S(\boldsymbol{x}_j^{(t)}), S(\boldsymbol{y}))$, and set $\epsilon_t = \max_{\{j:w_j>0\}} d_j$.

8: For all j such that $w_j > 0$:

a. Sample $\boldsymbol{\theta}'_j \sim q_t(\cdot|\boldsymbol{\theta}_j^{(t-1)})$.

b. With probability

$$\min\left(1, \frac{\pi(\boldsymbol{\theta}'_j)}{\pi(\boldsymbol{\theta}_j^{(t-1)})} \frac{q(\boldsymbol{\theta}_j^{(t-1)}|\boldsymbol{\theta}'_j)}{q(\boldsymbol{\theta}'_j|\boldsymbol{\theta}_j^{(t-1)})}\right) \tag{3.36}$$

proceed to step c, else set $\boldsymbol{\theta}_j^{(t)} = \boldsymbol{\theta}_j^{(t-1)}$, $S(\boldsymbol{x}_j^{(t)}) = S(\boldsymbol{x}_j^{(t-1)})$ and do not proceed to step c.

c. Simulate $\boldsymbol{x}'_j \sim p(\cdot|\boldsymbol{\theta}'_j)$ and compute $d'_j = \rho_{\widehat{\beta}_t}(S(\boldsymbol{x}'_j), S(\boldsymbol{y}))$.

d. With probability

$$\mathbb{I}\{d'_j < \epsilon_t\} \tag{3.37}$$

set $\boldsymbol{\theta}_j^{(t)} = \boldsymbol{\theta}'_j$ $d_j = d'_j$ and $S(\boldsymbol{x}_j^{(t)}) = S(\boldsymbol{x}'_j)$ else set $\boldsymbol{\theta}_j^{(t)} = \boldsymbol{\theta}_j^{(t-1)}$ and $S(\boldsymbol{x}_j^{(t)}) = S(\boldsymbol{x}_j^{(t-1)})$.

9: Whilst the acceptance rate of step 9 is greater than 0, return to Step 3.

Chapter 4

Examples

In this chapter we apply the ABC methods which were introduced in Chapters 2 and 3 to a range of examples. We consider both models for which the likelihood is tractable, and thus the true posterior distribution is known, as well as those for which the likelihood is intractable. In all of the examples we consider the benefits gained by using Auto-SS SMC-ABC, as presented in Algorithm 8, compared to other ABC implementations. Comparisons between the algorithms are made in terms of computational cost and improvements in the accuracy of inference.

4.1 Bivariate Gaussian Model

In Chapter 1 we applied ABC methods to the Bivariate Gaussian distribution with unknown mean $\boldsymbol{\mu} \in \mathbb{R}^2$ and unknown covariance matrix $\Sigma \in \mathbb{M}_+^{2\times 2}$. This model has a tractable likelihood, given by

$$p(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma) = \prod_{i=1}^{n} (2\pi)^{-1} |\Sigma|^{-1/2} \exp\Big\{-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu})\Big\}. \tag{4.1}$$

We use conjugate *Normal-Inverse-Wishart* priors, as seen in Example 2.1:

$$(\boldsymbol{\mu}, \Sigma) \sim \mathcal{N} - Inv - Wishart(\boldsymbol{\mu}_0, \kappa_0, \nu_0, \Lambda_0^{-1}). \tag{4.2}$$

Thus the posterior distribution is tractable and also follows a *Normal-Inverse-Wishart* distribution. Formulae for updating the hyper-parameters can be found in Equations (2.11) to (2.14).

We select the following hyper-parameters:

$$\boldsymbol{\mu}_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \kappa_0 = 1, \quad \nu_0 = 4 \quad \text{and } \Lambda_0^{-1} = \begin{pmatrix} \frac{4}{3} & -\frac{2}{3} \\ -\frac{2}{3} & \frac{4}{3} \end{pmatrix}, \tag{4.3}$$

and set $n = 10,000$, so simulated data is of the form,

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_{10,000} \end{pmatrix} = \begin{pmatrix} x_{1,1} & x_{1,2} \\ \vdots & \vdots \\ x_{10,000,1} & x_{10,000,2} \end{pmatrix}, \tag{4.4}$$

where

$$\boldsymbol{x}_i | \boldsymbol{\mu}, \Sigma \sim^{iid} \mathcal{N}(\boldsymbol{\mu}, \Sigma), \tag{4.5}$$

and

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}. \tag{4.6}$$

The selection of $n = 10,000$ is large enough to ensure that the data is informative about the parameters $\boldsymbol{\mu}$ and Σ, and thus sensibly selected summary statistics will also be informative.

In order to evaluate and compare the ABC methods, we apply the ABC algorithms to 500 pseudo-observed data sets that are simulated with parameters drawn from the prior distribution given above.

4.1.1 Summary Statistics

To illustrate the impact of summary statistics on the ABC posterior distribution we implement ABC algorithms using two sets of summary statistics. We will compare the ABC posteriors returned with sufficient statistics to those returned when using statistics that are not sufficient for the data.

- **Sufficient Statistics** For this distribution, the sample mean $\bar{\boldsymbol{x}}$ and the empirical covariance matrix $\hat{\Sigma}$ are sufficient statistics for the parameters $\boldsymbol{\mu}$ and Σ respectively. We

implement ABC with these summary statistics, and take this to be an illustration of the 'best' possible inference we can expect to achieve from an ABC algorithm. These sufficient statistics are five dimensional (since the covariance is symmetric and therefore contains a repeated element).

- **Naive Statistics** These are selected to reflect the process of a scientist summarising their data excessively and naively, leading to non-linear relationships with the parameters and high dimensional summary statistics. The 13 naive summary statistics we use are given in Table 4.1.

	Summary Statistic	Interpretation
$s_1(\boldsymbol{x})$	$n^{-1}\sum_{i=1}^{n}(x_{i,1}^2 \cdot sign(x_{i,1}))$	location
$s_2(\boldsymbol{x})$	$n^{-1}\sum_{i=1}^{n}(x_{i,2}^2 \cdot sign(x_{i,2}))$	location
$s_3(\boldsymbol{x})$	$n^{-1}\sum_{i=1}^{n}\left(x_{i,1}^2 \cdot sign(x_{i,1}) - s_1(\boldsymbol{x})\right)^2$	scale
$s_4(\boldsymbol{x})$	$n^{-1}\sum_{i=1}^{n}\left(x_{i,1}^2 \cdot sign(x_{i,1}) - s_1(\boldsymbol{x})\right)\left(x_{i,2}^2 \cdot sign(x_{i,2}) - s_2(\boldsymbol{x})\right)$	scale
$s_5(\boldsymbol{x})$	$n^{-1}\sum_{i=1}^{n}\left(x_{i,2}^2 \cdot sign(x_{i,2}) - s_2(\boldsymbol{x})\right)^2$	scale
$s_6(\boldsymbol{x})$	$n^{-1}\sum_{i=1}^{n}\left(x_{i,1} - \bar{\boldsymbol{x}}_{\cdot,1}\right)^3 / \left(n^{-1}\sum_{i=1}^{n}\left(x_{i,1} - \bar{\boldsymbol{x}}_{\cdot,1}\right)^2\right)^{3/2}$	skewness
$s_7(\boldsymbol{x})$	$n^{-1}\sum_{i=1}^{n}\left(x_{i,2} - \bar{\boldsymbol{x}}_{\cdot,2}\right)^3 / \left(n^{-1}\sum_{i=1}^{n}\left(x_{i,2} - \bar{\boldsymbol{x}}_{\cdot,2}\right)^2\right)^{3/2}$	skewness
$s_8(\boldsymbol{x})$	$n^{-1}\sum_{i=1}^{n}\left(x_{i,1} - \bar{\boldsymbol{x}}_{\cdot,1}\right)^4 / \left(n^{-1}\sum_{i=1}^{n}\left(x_{i,1} - \bar{\boldsymbol{x}}_{\cdot,1}\right)^2\right)^{4/2}$	kurtosis
$s_9(\boldsymbol{x})$	$n^{-1}\sum_{i=1}^{n}\left(x_{i,2} - \bar{\boldsymbol{x}}_{\cdot,2}\right)^4 / \left(n^{-1}\sum_{i=1}^{n}\left(x_{i,2} - \bar{\boldsymbol{x}}_{\cdot,2}\right)^2\right)^{4/2}$	kurtosis
$s_{10}(\boldsymbol{x})$	$n^{-1}\sum_{i=1}^{n}\left(x_{i,1} - \bar{\boldsymbol{x}}_{\cdot,1}\right)^5 / \left(n^{-1}\sum_{i=1}^{n}\left(x_{i,1} - \bar{\boldsymbol{x}}_{\cdot,1}\right)^2\right)^{5/2}$	hyper skewness
$s_{11}(\boldsymbol{x})$	$n^{-1}\sum_{i=1}^{n}\left(x_{i,2} - \bar{\boldsymbol{x}}_{\cdot,2}\right)^5 / \left(n^{-1}\sum_{i=1}^{n}\left(x_{i,2} - \bar{\boldsymbol{x}}_{\cdot,2}\right)^2\right)^{5/2}$	hyperskewness
$s_{12}(\boldsymbol{x})$	$n^{-1}\sum_{i=1}^{n}\left(x_{i,1} - \bar{\boldsymbol{x}}_{\cdot,1}\right)^6 / \left(n^{-1}\sum_{i=1}^{n}\left(x_{i,1} - \bar{\boldsymbol{x}}_{\cdot,1}\right)^2\right)^{6/2}$	hyperflatness
$s_{13}(\boldsymbol{x})$	$n^{-1}\sum_{i=1}^{n}\left(x_{i,2} - \bar{\boldsymbol{x}}_{\cdot,2}\right)^6 / \left(n^{-1}\sum_{i=1}^{n}\left(x_{i,2} - \bar{\boldsymbol{x}}_{\cdot,2}\right)^2\right)^{6/2}$	hyperflatness

TABLE 4.1: Naive Summary Statistics used in the Bivariate Gaussian Example

In Figure 4.1 we plot the relationship between the five marginal parameters of interest, and a selection of summary statistics. Samples are drawn from the prior predictive distribution. It is clear that there is a strong linear relationship between the parameters and the sufficient

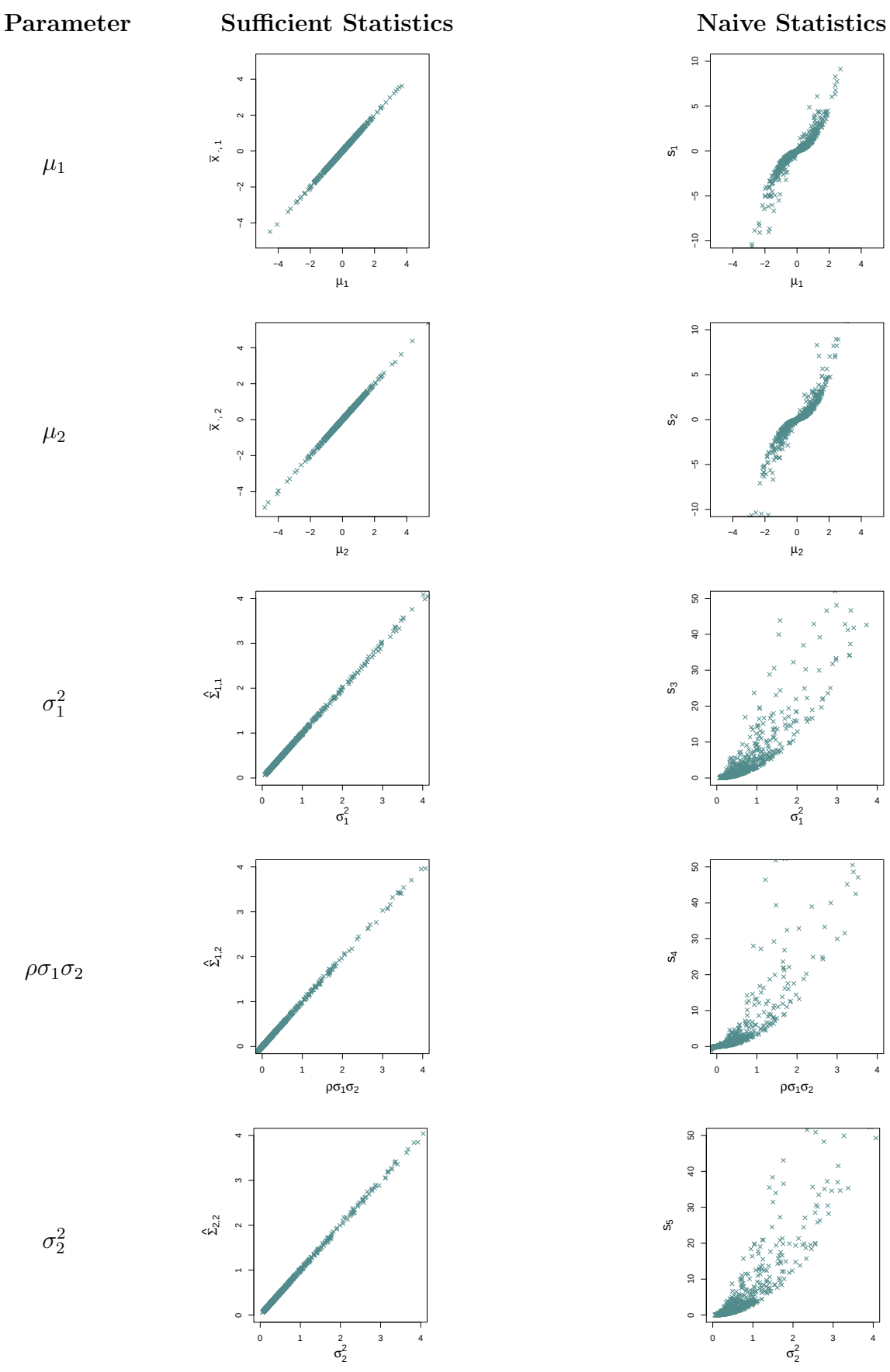


FIGURE 4.1: The relationship between the parameters of the Bivariate Gaussian model, and the summary statistics. The left hand column shows the relationship between the parameters and the sufficient statistics, whilst the right hand column shows the relationship between the parameters and five of the 13 naively selected statistics.

statistics, and that there is noisy, non-linear relationship between the parameters and the naive summary statistics.

4.1.2 Implementation Details

Using the sufficient statistics for the parameters of the Bivariate Gaussian model, we implement the SMC-ABC algorithm of Del Moral et al. [1], as given in Section 3.1. We wish to use these ABC posteriors as a representation of the best results that ABC can give, and so use the adapted stopping rule given in Algorithm 5 to ensure that the algorithm reaches a low tolerance before terminating. Furthermore, to ensure the best results, we implement the algorithm with parameter $\alpha = 0.99$ and set $N = 1,000$. In Section 3.1 we showed that as α tends to 1, the ABC posterior distribution contains more unique particles.

We implement the following ABC algorithms using the naive summary statistics given in Table 4.1:

- **Rejection ABC** 750,000 data sets are sampled from the prior predictive distribution. We use the 'abc' R package of Csillery et al. [23], and as is standard in that package, summary statistics are scaled by the median absolute deviation (MAD) from the median, and the distance metric used is Euclidean distance. Posterior distributions are chosen to be the 1000 points which are closest to the observed data. (This corresponds to an acceptance rate of ≈ 0.0013).

- **Rejection ABC with Regression Correction** The posteriors obtained by Rejection ABC, as explained above, are then treated with the regression correction method of Beaumont et al. [6]. Again, we use the 'abc' package in R [23] to implement this.

- **SMC-ABC** We use the same tuning parameters and stopping rule (Algorithm 5) as the SMC-ABC implementation on sufficient statistics, as given above. The implementation differs only in the summary statistics used.

- **SMC-ABC with Summary Statistic Selection at $t = 0$** We implement the SMC-ABC algorithm, as above. However, summary statistics are selected at time $t = 0$, using linear regression, following the method of Fearnhead and Prangle [22].

- **Auto-SS SMC-ABC** We implement Algorithm 8 as given in Chapter 3, with $N = 1000$ and $\alpha = 0.9$.

In all ABC-SMC implementations we need to select a proposal distribution $q_t(\cdot, \cdot)$ which maps from parameters $\boldsymbol{\mu}$ and Σ to proposed parameters $\boldsymbol{\mu}'$ and Σ'. We use two independent proposal distributions for the two parameters.

Given a current parameter value $\boldsymbol{\mu}_t$, we propose a value $\boldsymbol{\mu}'$ using

$$\boldsymbol{\mu}' \sim \mathcal{N}(\boldsymbol{\mu_t}, \tilde{\Sigma}_t), \tag{4.7}$$

where

$$\tilde{\Sigma}_t = \begin{pmatrix} 2\hat{\sigma}_1^2 & 0 \\ 0 & 2\hat{\sigma}_2^2 \end{pmatrix}, \tag{4.8}$$

and $\hat{\sigma}_j^2$ is the empirical variance of μ_j at time t.

To propose an update for Σ_t we require a proposal distribution which proposes only positive semi-definite, symmetric matrices.

Given a covariance matrix Σ_t, we begin by carrying out a Cholesky decomposition of the matrix, so we write $\Sigma = LL^T$, where

$$L = \begin{pmatrix} a & 0 \\ b & c \end{pmatrix}, \tag{4.9}$$

and $a, c > 0$.

Conditional upon the diagonal elements of L being greater than 0, this decomposition is unique. We then perturb L to a lower diagonal matrix $\tilde{L}$, by perturbing each non-zero component of L according to a truncated Gaussian distribution:

$$\tilde{L} = \begin{pmatrix} \tilde{a} & 0 \\ \tilde{b} & \tilde{c} \end{pmatrix}, \tag{4.10}$$

where

$$\tilde{a} \sim f(a, 2\sigma_a^2, 0, \infty) \tag{4.11}$$

$$\tilde{b} \sim \mathcal{N}(b, 2\sigma_b^2) \tag{4.12}$$

$$\tilde{c} \sim f(c, 2\sigma_c^2, 0, \infty) \tag{4.13}$$

and $f(x, \sigma^2, l, u)$ is the truncated Gaussian distribution, with mean x and variance σ^2, truncated to $[l, u]$. Hyper-parameters σ_a^2, σ_b^2 and σ_c^2 are selected to be the empirical variance of the set of values of a, b and c respectively from the set of lower diagonal matrices, L, at the current iteration of the algorithm.

Proposals for a, b and c are independent, so the probability of a matrix L being perturbed to $\tilde{L}$ is given by

$$p(L \to \tilde{L}) = p(a \to \tilde{a}) \times p(b \to \tilde{b}) \times p(c \to \tilde{c}). \tag{4.14}$$

The proposed covariance matrix is then given by $\tilde{\Sigma} = \tilde{L}\tilde{L}^T$.

4.1.3 Results

We begin by looking at the results obtained when implementing Rejection ABC both with and without regression correction. Figure 4.2 shows the marginal posterior means obtained through ABC plotted against the analytic posterior means for the two methods. The Pearson correlation coefficient, r, is given in each frame.

It is immediately apparent that ABC with regression correction gives significantly improved estimates of the posterior means, compared to standard Rejection ABC. The covariance, $\rho\sigma_1\sigma_2$, is poorly estimated by both algorithms, though the regression correction does improve the results, with the Pearson correlation coefficient increasing from 0.097 to 0.785 when regression correction

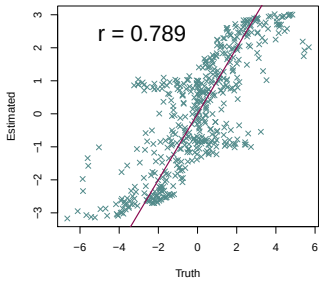
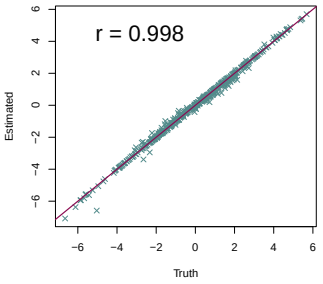
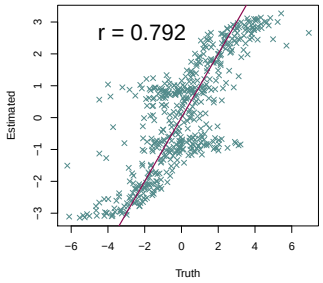
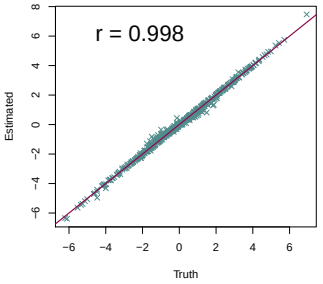
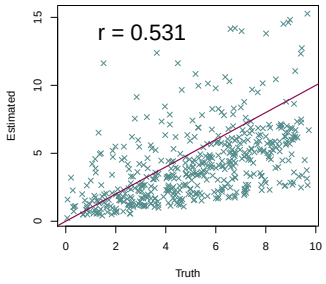
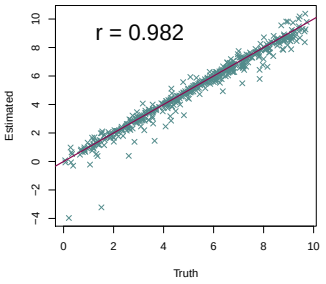
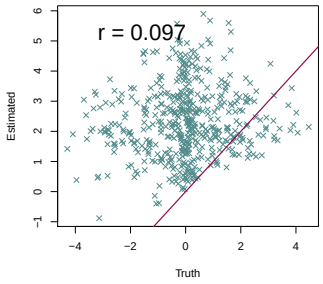
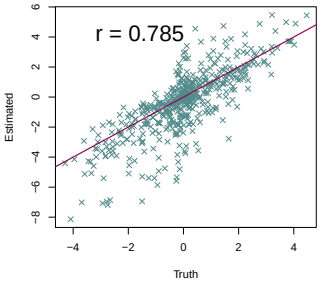
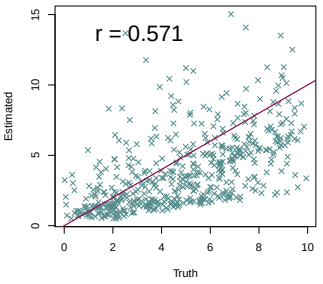
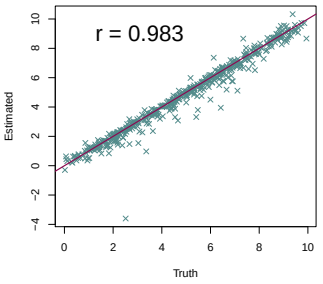
Parameter	Rejection ABC	With Regression Correction
μ_1		
μ_2		
σ_1^2		
$\rho\sigma_1\sigma_2$		
σ_2^2		

FIGURE 4.2: Analytic posterior means, plotted against ABC posterior means obtained through Rejection ABC (left hand column) and Rejection ABC with Regression Correction (right hand column).

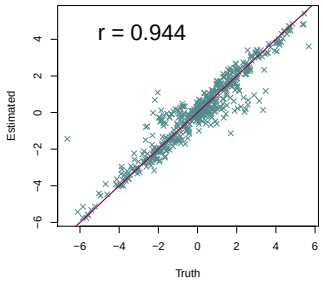
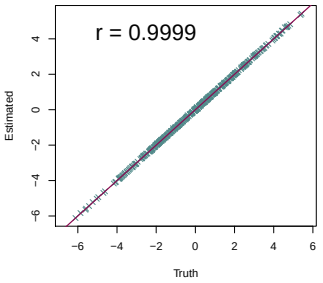
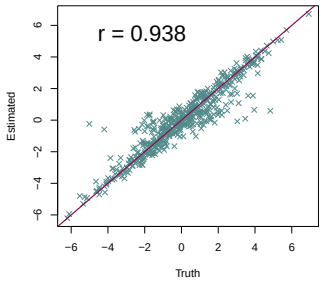
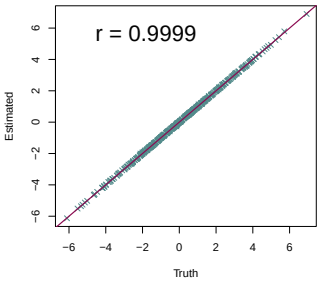
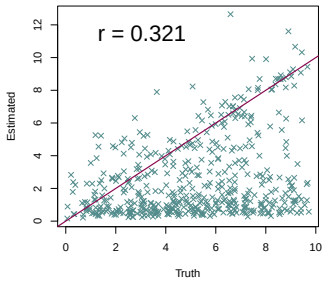
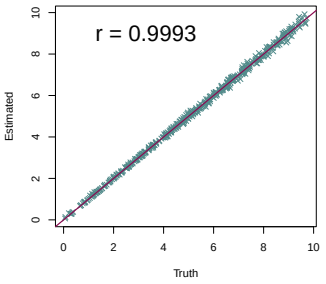
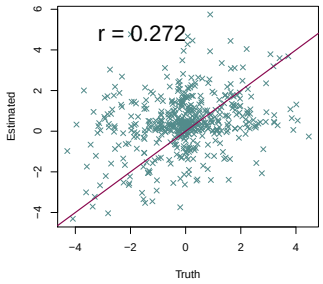
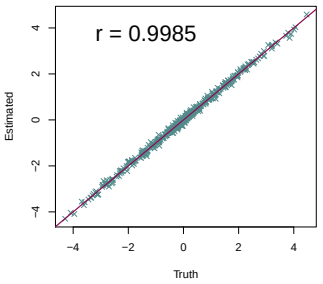
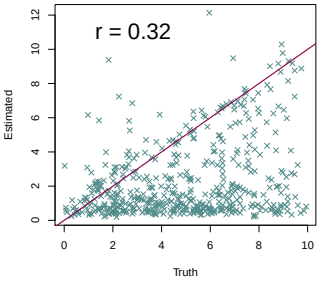
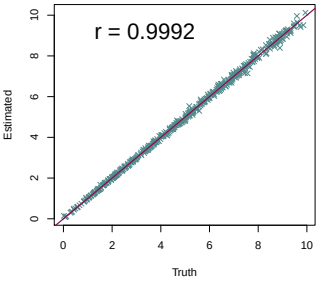
Parameter	SMC-ABC	Auto-SS SMC-ABC
μ_1		
μ_2		
σ_1^2		
$\rho\sigma_1\sigma_2$		
σ_2^2		

FIGURE 4.3: Analytic posterior means, plotted against ABC posterior means obtained through SMC-ABC (left hand column) and Auto-SS SMC-ABC.

Method	μ_1	μ_2	$\sigma^2_{1,1}$	$\sigma_1\sigma_2\rho$	$\sigma^2_{2,2}$
SMC-ABC with Sufficient Statistics	0	0	0	0	0
Rejection ABC, tol ≈ 0.0013	346,651	340,437	112,664	253,399	106,645
Regression Correction	4,151	3,236	3,808	59,020	3,857
SMC-ABC with 13 none-sufficient statistics	99,561	109,834	225,160	93,164	231,709
SMC-ABC with SS selection at $t = 0$	8,857	10,367	7,406	20,440	8,701
SMC-ABC with Iterative SS Selection	**80**	**86**	**113**	**209**	**138**

TABLE 4.2: Relative mean squared error for various ABC implementations, given in terms of percentage difference from the MSE obtained when implementing SMC-ABC with sufficient Statistics, rounded to the closest percent. The smallest (non-zero) value in each column is given in bold.

is used. This figure shows that, even for a simple model, Rejection ABC alone can lead to poor inference when implemented using non-optimal summary statistics.

In the left column of Figure 4.3 we plot the ABC posterior means against the analytic posterior means for SMC-ABC using 13 naive statistics. Comparing this to the Rejection ABC results seen in Figure 4.2 shows that better inference is obtained through SMC-ABC for both components of the mean, and for the covariance parameter, $\rho\sigma_1\sigma_2$. However the inference for σ_1^2 and σ_2^2 is worse for SMC-ABC than it is for Rejection ABC. This is reflected in both the values of the Pearson correlation coefficient, and in the relative mean squared error, given in Table 4.2.

The right hand column of Figure 4.3 shows the ABC posterior means plotted against the true posterior means, computed using the Auto-SS SMC-ABC Algorithm. It is clear that the posterior means of all parameters are extremely well approximated. The most notable improvement in inference, when compared to the other ABC algorithms, is in terms of estimating the elements of the covariance matrix.

In Table 4.2 we give the relative mean squared error of the algorithms seen previously, compared to the MSE obtained when SMC-ABC is implemented using sufficient summary statistics. The table shows that, of the methods considered, the best results for each parameter, when using non-sufficient summary statistics, are obtained by using Auto-SS SMC-ABC.

Figure 4.4 shows boxplots of the number of simulations from the model required to give the posteriors used to generate the results in Table 4.2, and Figures 4.2 and 4.3. The implementation of SMC-ABC with non-sufficient statistics required, on average, the fewest simulations from the model. However, this implementation also gave the worst results, in terms of mean squared error,

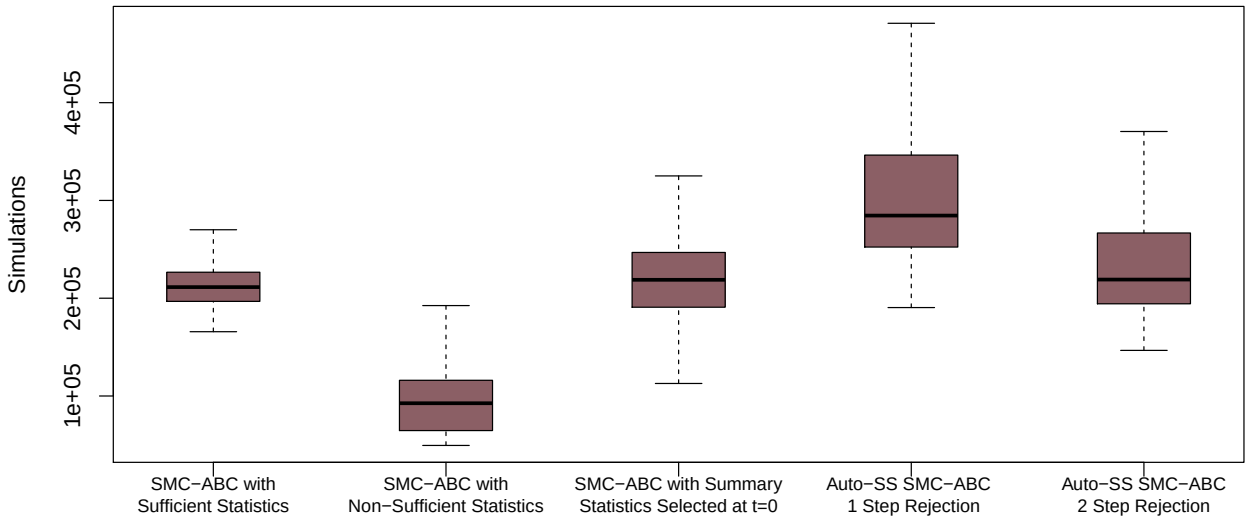


FIGURE 4.4: Boxplot showing the number of simulations from the model needed for each method.

of the methods compared in the figure. The most model simulations were needed to implement the Auto-SS SMC-ABC algorithm, using the standard Metropolis-Hastings acceptance decision. However, in practice we implemented the Two-Stage Metropolis-Hastings acceptance decision, as seen in Algorithm 6, there is a clear reduction in computational cost, with the median number of model simulations required being just over $219,000$, compared to $211,400$ required for the implementation of SMC-ABC with sufficient statistics.

Figure 4.5 shows the number of simulations from the model required to give posterior distributions for SMC-ABC with non-sufficient summary statistics, both with and without the Two-Stage Metropolis-Hastings acceptance decision, presented in Algorithm 7. The data shown in the boxplot is generated by implementing the two algorithms on 500 pseudo-observed data sets. The stopping rule used for all simulations is given in Algorithm 5. From Figure 4.5 we see that the median number of simulations from the model required is reduced under the Two-Stage method. In Table 4.3 we give the mean squared error for the posterior distributions. The results indicate that implementing SMC-ABC with the Two-Stage Metropolis-Hastings acceptance method does not lead to reduced accuracy.

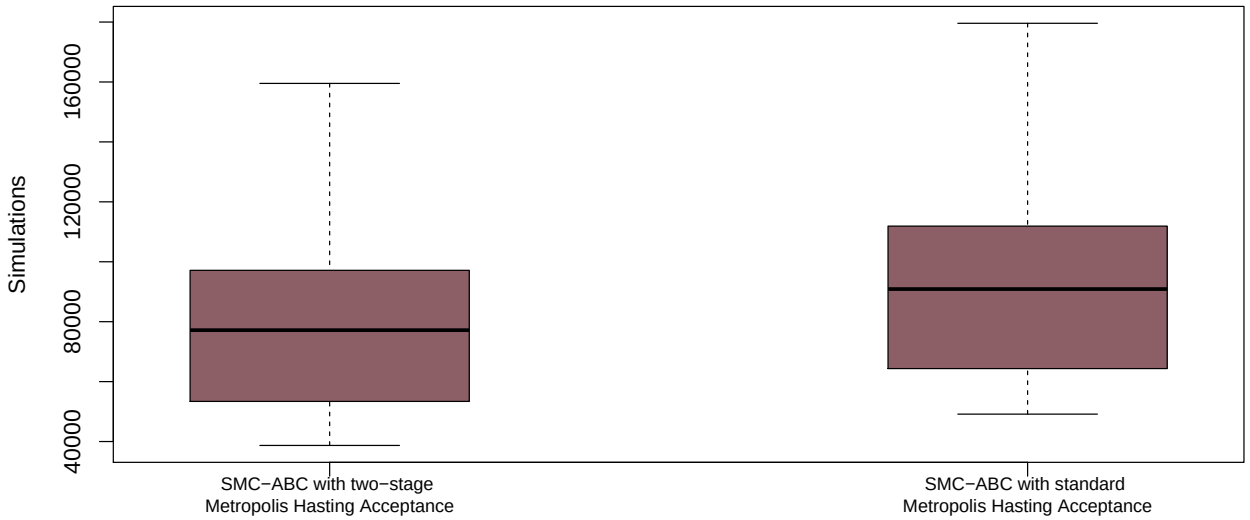


FIGURE 4.5: Boxplots showing the number of simulations from the model needed for SMC-ABC both with and without the Two-Stage Metropolis-Hastings acceptance method.

Method	μ_1	μ_2	$\sigma^2_{1,1}$	$\sigma_1\sigma_2\rho$	$\sigma^2_{2,2}$
With Two-Stage Metropolis-Hastings Acceptance	0.5337	0.6611	14.70	3.817	14.25
Without Two-Stage Metropolis-Hastings Acceptance	0.5269	0.5842	15.04	3.168	14.52

TABLE 4.3: Mean squared error for SMC-ABC with and without the Two-Stage Metropolis-Hastings acceptance method. Results are based on 500 pseudo-observed data sets.

The example shown in this section illustrates the importance of summary statistic selection methods and shows that, even with summary statistics that are not sufficient for the data, it is possible to obtain accurate estimates of the posterior mean. There is a notable reduction in the number of model simulations needed when implementing the Two-Stage Metropolis-Hastings algorithm, and all ABC algorithms that use a Metropolis-Hastings kernel can benefit from this alternative method.

4.2 g-and-k distribution

The g-and-k distribution [31] is defined only through its quantile function. Specifically, it does not have a tractable likelihood. Thus, in order to make posterior inference, likelihood free methods such as ABC must be used.

The distribution is specified in terms of five parameters, A, B, c, g and k. Following the lead of earlier work ([22] [31] [32] [33]), we fix $c = 0.8$ for the remainder of this section.

The quantile function, which defines the distribution, is given by

$$F^{-1}(x; A, B, c, g, k) = A + B\left(1 + c\frac{1 - \exp(-gz(x))}{1 + \exp(-gz(x))}\right)(1 + z(x)^2)^k z(x), \tag{4.15}$$

where $z(x)$ is the standard Gaussian quantile, $B > 0$ and $k > 0$.

The g-and-k distribution provides an interesting challenge for ABC: it is easy to sample from the distribution using the inverse method (Algorithm 9), but it is non-trivial to select summary statistics which are appropriate for inference. There is no obvious low dimensional summary of the data which retains information about the parameters, and for this reason high dimensional summaries would be sensible, but result in poor outcome due to the curse of dimensionality.

Allingham et al. [33] applied MCMC-ABC [9] to one data set simulated from the g-and-k distribution. They selected summary statistics to be the order statistics of a data sample of size $10,000$ and showed that, when MCMC-ABC was iterated 10^6 times, the resultant ABC posterior distributions for A, B and k were centred around the true parameter values and had narrow support. The inference for parameter g was poor, with the posterior distribution having a much larger support than the posterior distribution for the other parameters.

Fearnhead and Prangle [22] considered the impact of summary statistic selection methods and the regression correction of Beaumont et al. [6] on the ABC posterior distribution. By using summary statistics given by the order statistics of the data, as in Allingham et al. [33], as well as powers of the order statistics, they showed that the semi-automatic summary statistic selection method (described in Section 3.4.1) leads to a reduction in mean squared error of the resultant posterior distributions, compared to just using 100 order statistics.

In this section we obtain ABC posterior distributions using a range of SMC-ABC and Rejection ABC methods. We compare the results in terms of computational cost and accuracy.

Algorithm 9 Inverse Sampling

To sample from any distribution $F(\cdot)$, provided one has access to the quantile function, $F^{-1}(\cdot)$, the following steps should be iterated:

1: Sample $u \sim \mathcal{U}(0,1)$.
2: Compute x such that $F^{-1}(x) = u$.

4.2.1 Summary Statistics and Implementation Details

ABC methods are applied to 50 pseudo-observed data sets. Following Allingham et al. [33] and Fearnhead and Prangle [22], independent parameter values are sampled from a uniform prior on $[0, 10]$. Data sets of size $10,000$ are sampled from the distribution, and as in Fearnhead and Prangle [22], summary statistics are 100 evenly spaced order statistics, and their second, third and fourth powers. Thus each data set is summarised by 400 summary statsitics.

The following ABC methods are implemented on 50 pseudo-observed data sets, simulated from parameters drawn from the prior distribution given above.

- **Rejection ABC** $200,000$ data sets are sampled from the prior predictive distribution. Again we use the 'abc' R package [23]. Summary statistics are scaled by the median absolute deviation (MAD) from the median, and the distance metric used is Euclidean distance. Posterior distributions are taken to be the parameters which simulated the $1,000$ data sets that lie closest to the observed data. This corresponds to an acceptance rate of 0.005.

- **Rejection ABC with Regression Correction** The posteriors obtained by Rejection ABC, as explained above, are then treated with the regression correction method of Beaumont et al. [6]. Again, we use the 'abc' package in R [23] to implement this.

- **SMC-ABC** The SMC-ABC algorithm of Del Moral et al. [1] is ran, using the stopping rule given in Algorithm 5, with $N = 1000$ and $\alpha = 0.95$.

- **SMC-ABC with Summary Statistic Selection at $t = 0$** We implement the SMC-ABC algorithm, as above. However, summary statistics are selected at time $t = 0$, based on a sample of size $2,000$ using linear regression, using semi-automatic summary statistic selection [22].
- **Auto-SS SMC-ABC** Algorithm 8 is implemented with $N = 1000$ and $\alpha = 0.9$.

All simulations from the g-and-k distribution are done using the 'gk' package in R [34].

The accuracy of the resultant posterior distributions is evaluated quantitatively through mean root summed squared error (MRSSE), and through the Pearson correlation coefficient between the true parameter value and the ABC posterior means.

Let $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n$ be n parameter vectors from the ABC posterior distribution, and let $\boldsymbol{\theta}_{\text{obs}}$ be the true parameter vector, from which one pseudo-observed data set was simulated. Then the root summed squared error (RSSE) is defined as

$$RSSE = \sqrt{\sum_{i=1}^{n} \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_{obs}\|^2}. \tag{4.16}$$

When considering m pseudo-data sets (in our case $m = 50$) each with root summed squared error given by $RSSE_j$, for $j = 1, \ldots, m$, the mean root summed squared error (MRSSE) is given by

$$\frac{1}{m} \sum_{j=1}^{m} RSSE_j. \tag{4.17}$$

4.2.2 Results

Table 4.4 gives the mean root summed squared error (Equation (4.17)) for the five methods implemented, based on 50 pseudo-observed data sets. For all methods, the parameter for which inference was worst was g. This was also the case in the results of Fearnhead and Prangle [22] and Allingham et al. [33]. For A, B and g, the smallest MRSSE was obtained using Auto-SS SMC-ABC, whilst Rejection ABC followed by Regression Correction lead to the best inference for k. For parameters g and k, a smaller MRSSE was obtained using SMC-ABC without summary

statistic selection, compared to SMC-ABC with summary statistic selection at $t = 0$. This indicates that the training set, on which the summary statistic selection was based, was not large enough.

The left hand column of Figure 4.6 shows the true parameter value plotted against the ABC posterior mean for Rejection ABC, both with regression correction (pink crosses) and without (blue dots). The plot shows that a huge improvement in accuracy of the posterior can be made through regression correction. For parameters g and k, the Rejection ABC posterior means (blue) appear to tend towards the prior mean, five. However once regression correction is applied (pink) the posterior means show a strong positive correlation with the true parameter values.

The right hand column of Figure 4.6 shows the posterior means obtained through SMC-ABC, plotted against the true parameter value. There appears to be a range of runs for which the posterior mean lies on, or very close to, the true parameter value. However, there is also a moderate number of points which lie further from the line. This is reflected in Figure 4.8 in which the value of ϵ at which the ABC-SMC runs terminated are plotted, in ascending order. The figure shows that around 40 of the 50 runs terminate at a tolerance less than 2. The remaining iterations terminate at much higher tolerance values, suggesting that the SMC chains became 'stuck'. Such behaviour may be avoided by increasing the number of particles, N, from $1,000$.

Figure 4.7 shows the posterior means obtained by selecting summary statistics at $t = 0$ using the semi-automatic summary statistic selection method, then SMC-ABC, as well as the posterior means obtained through Auto-SS SMC-ABC. The Auto-SS SMC-ABC show three outliers, suggesting that these particular implementations did not converge on the posterior distribution. However, the other runs gave accurate estimates of the true parameter values. The results obtained from SMC-ABC with summary statistics selected at $t = 0$ appear to be recovering the posterior mean, five, for parameters A, B and g, however inference for k is reasonable, with a Pearson correlation coefficient of over 0.7.

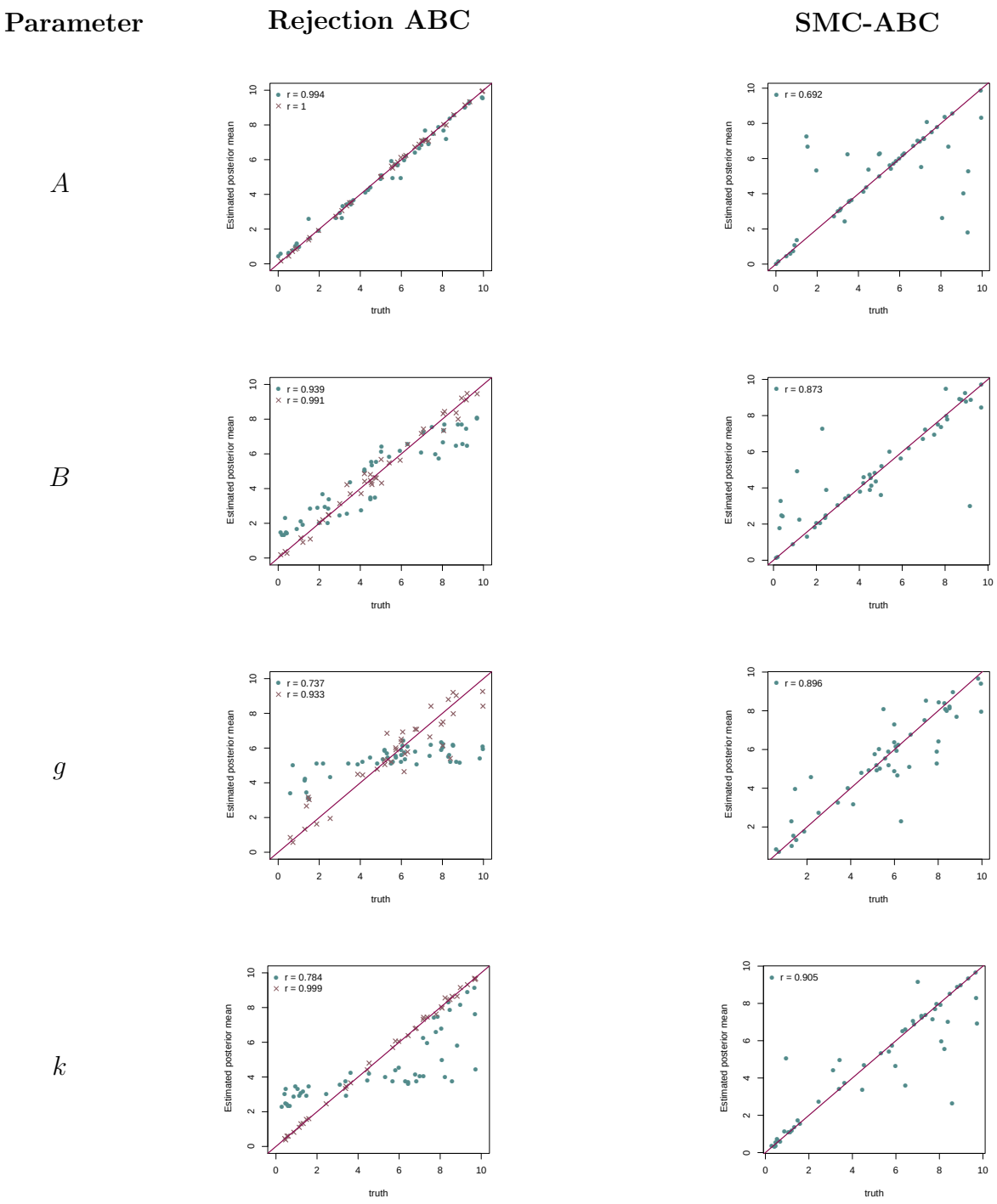


FIGURE 4.6: Analytic posterior means, plotted against ABC posterior means obtained through Rejection ABC (left hand column) and Rejection ABC with Regression Correction (right hand column).

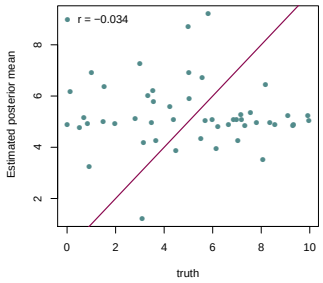
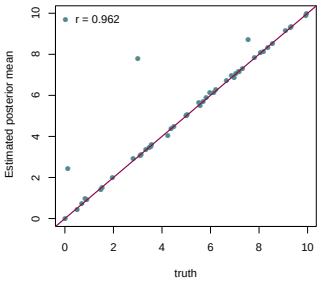
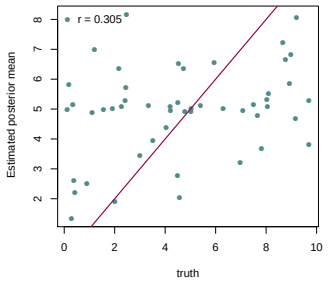
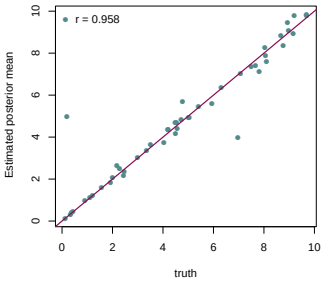
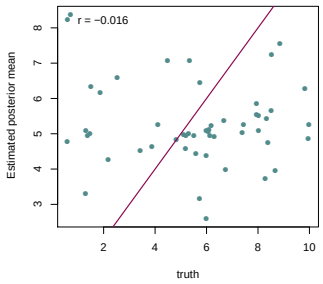
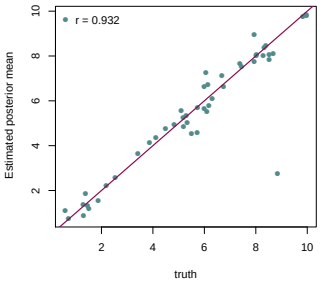
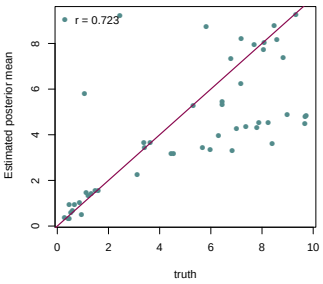
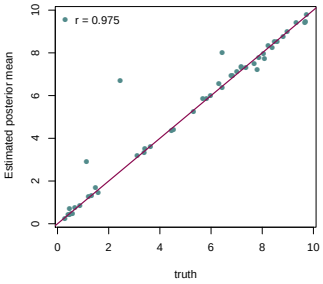
Parameter	Semi-Automatic	Auto-SS SMC-ABC
A		
B		
g		
k		

FIGURE 4.7: Analytic posterior means, plotted against ABC posterior means obtained through SMC-ABC with summary statistics selected using Semi-automatic summary statistic selection at $t = 0$ (left hand column) and Auto-SS SMC-ABC (right hand column).

Method	A	B	g	k
Rejection ABC	21.40	62.51	102.69	86.07
Rejection ABC with Regression Correction	16.16	60.31	72.45	**12.18**
SMC-ABC	45.16	36.58	42.86	32.43
SMC-ABC with summary statistic selection at $t = 0$	19.70	36.08	63.24	39.43
Auto-SS SMC-ABC	**9.40**	**14.72**	**19.99**	12.88

TABLE 4.4: Root summed squared error of the ABC posterior distributions for the g-and-k distribution. The smallest value in each column is given in bold.

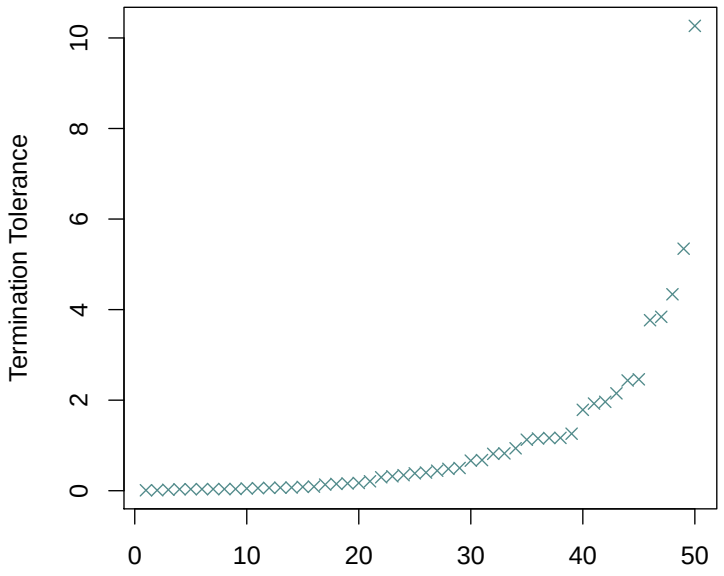


FIGURE 4.8: Ordered termination tolerance of SMC-ABC, applied to 50 g-and-k data sets.

Figure 4.9 shows box plots of the number of simulations from the g-and-k distribution required to obtain the ABC posterior distributions, for the SMC-ABC methods considered. Rejection ABC results in this section are based on 200,000 simulations from the model, and we see here that all SMC-ABC implementations required fewer than 200,000 model simulations. Auto-SS SMC-ABC appears to be the most computationally expensive method, requiring the most model iterations. However the final box plot illustrates that computational cost can be saved using the 2-Stage rejection method, as described in Algorithm 7.

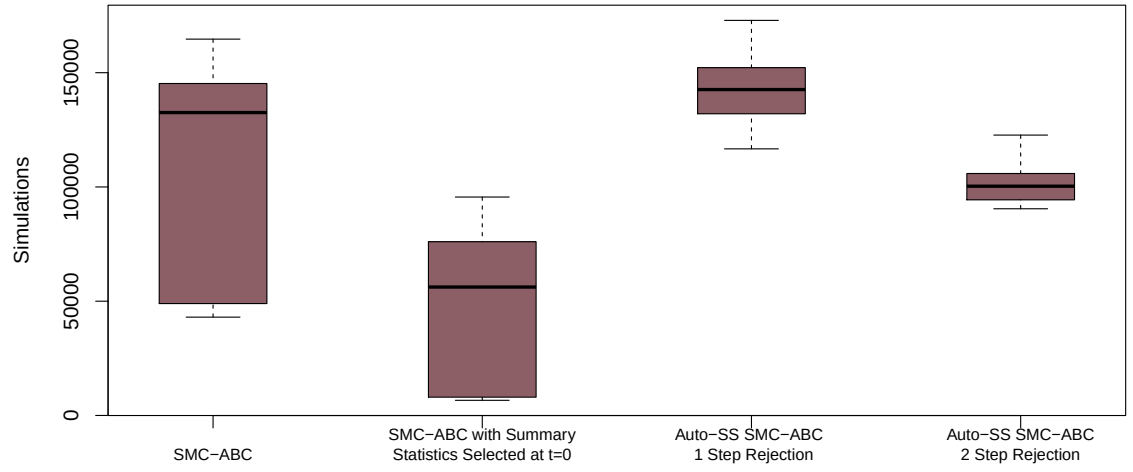


FIGURE 4.9: Box plots showing the number of simulations from the model required for each of the ABC methods, when applied to the g-and-k distribution.

4.3 An Individual Based Model for Earthworms

Individual based models (IBMs), also known as agent based models (ABMs), are models of one or more individuals from the same species. IBMs are commonly used in the field of ecology to aid the understanding of a species. By modelling the interaction between individuals, as well as between individuals and the surrounding environment, predictions can be made about a species as a whole. For example, a model can predict how a species will react to changes in its local climate, or changes in the availability of food sources. IBMs are stochastic, either due to randomness introduced through decisions made by individual agents, or through randomness in the surrounding environment.

Once an IBM has been developed, it must be calibrated so that it can be used to make inference about the species. An accurate model can, in many cases, produce output much faster than the time it takes to carry out a field experiment, and can be run multiple times with many different tuning parameters, thus giving wider insight into the process of interest. The calibration process requires that the model is fitted to observed data collected during field experiments. This step aims to ensure that the model is run at parameter values which may give rise to the observed data, meaning that the model can then be used to make predictions and inferences about the population being studied. This calibration process can be implemented through ABC: the aim is to determine the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{y})$, where $\boldsymbol{y}$ is the experimental data.

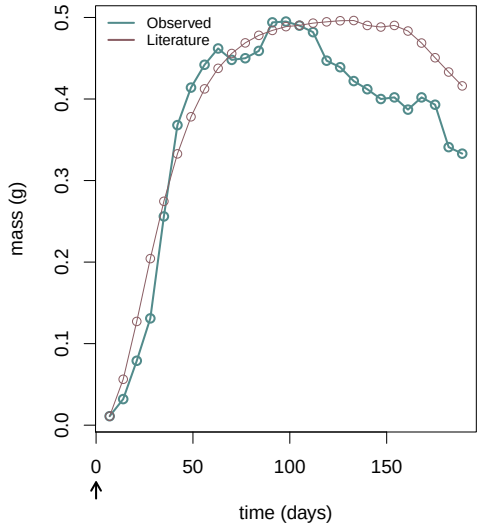
In this example, we consider an IBM of *Eisenia fetida*, a species of Earthworm. The model was developed by Johnston et al. [35] in Netlogo [36], a program for building IBMs. The model can be thought of as a sequence of processes carried out by the worms on an individual level with their actions being determined based on their current energy levels. Such models are known as energy budget models and are motivated by the understanding that an animal's behaviour is strongly governed by its energy levels at that particular time. (See Sibly et al. [37] for an overview of how such energy budget models are constructed.) In the *Eisenia fetida* model, if an earthworm has immediate access to food it will reproduce, grow, regenerate any damaged cells and store any extra energy in its reserves. If an earthworm does not have access to food it will use any energy reserves it has stored to carry out the above processes. Once the energy reserve

becomes empty, the earthworm will lose weight and eventually die if it does not gain access to new food.

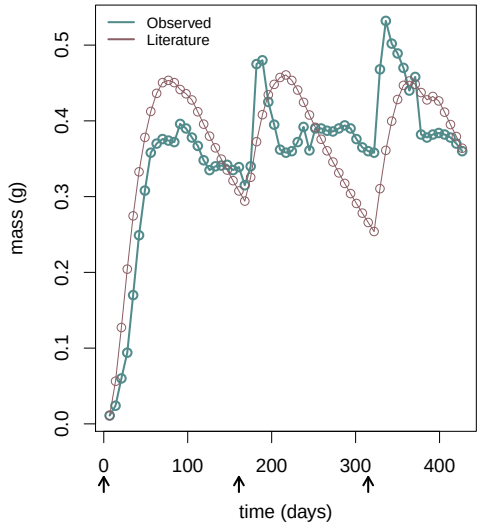
This IBM has 14 parameters, each relating to a property of the energy budget. These parameters are given in Table 4.5, along with assumed values for each of the parameters, taken from the literature. (Full details of the model and the values given in the literature can be found in Johnston et al. [35].) The model mimics four experiments that are presented in Gunadi et al. [38], Gunadi and Edwards [39] and Reinecke and Viljoen [40]. In the experiments, the feeding schedule of juvenile earthworms was controlled and the average mass of the earthworms, and the number of cocoons in the enclosure was recorded throughout. The observed data thus consists of 160 summary statistics, each of which is either an average mass or a number of cocoons - this observed data is plotted in Figure 4.10. This figure also shows the output of the model when it is run at the literature values (see Table 4.5) and, at least for these values it is clear that the model does a poor job of recreating the experimental data.

Symbol	Description	Literature Value
B_0	Taxon-specific normalisation constant (kJ/day)	967.0
E	Activation energy (eV)	0.25
E_c	Energy cost from tissue (kJ/g)	7.0
E_f	Energy from food (kJ/g)	10.6
E_s	Energy cost of synthesis (kJ/g)	3.6
h	Half saturation coefficient (g/0.001g)	3.5
IG_m	Maximum ingestion rate (g/day/g)	0.7
M_b	Mass at birth (g)	0.011
M_c	Mass of cocoon (g)	0.015
M_m	Maximum asymptotic mass (g)	0.5
M_p	Mass at sexual maturity (g)	0.25
r_B	Growth constant(1/day)	0.177
r_m	Maximum energy to reproduction (kJ/g/day)	0.182
s	Movement speed (m/day)	0.004

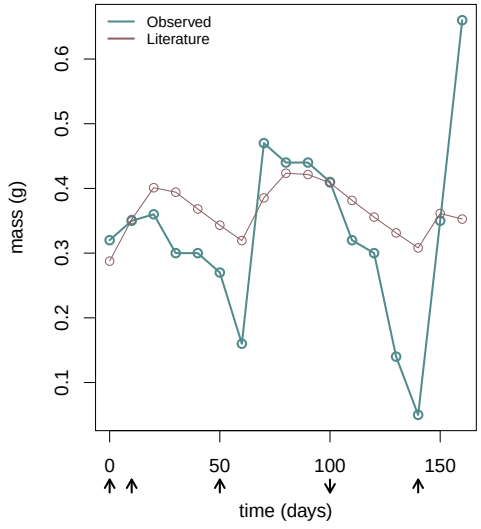
TABLE 4.5: Parameters of the *Eisenia fetida* model and their literature values.



(A) Data taken from Gunadi et al. [38]. The average mass of five juvenile earthworms was recorded weekly, over a 6-month period. The earthworms were placed in containers which contained cattle manure. The manure was not topped up.

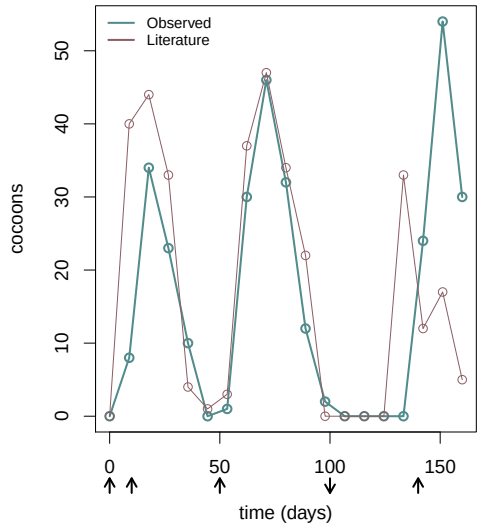


(B) Data taken from Gunadi and Edwards [39]. The average mass of eight juvenile earthworms was recoded weekly for 60 weeks. Manure was placed in the worms' container on weeks 0, 23 and 45.

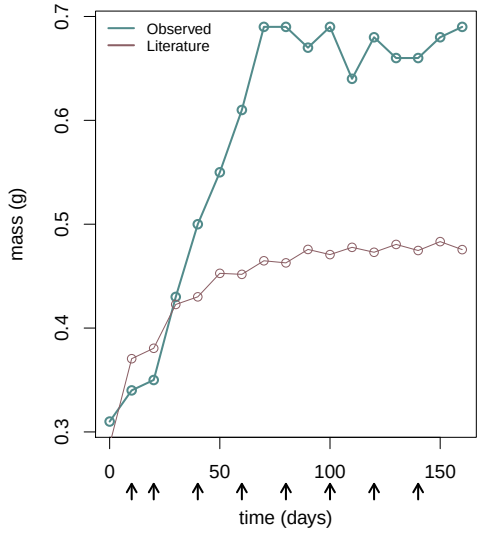


(C) Data taken from Reinecke and Viljoen [40]. The average weight of ten juvenile earthworms was recorded every ten days, over a period of 160 days. Manure was added to the container on days 0, 10, 60 and 140. On day 100 manure was removed from the container.

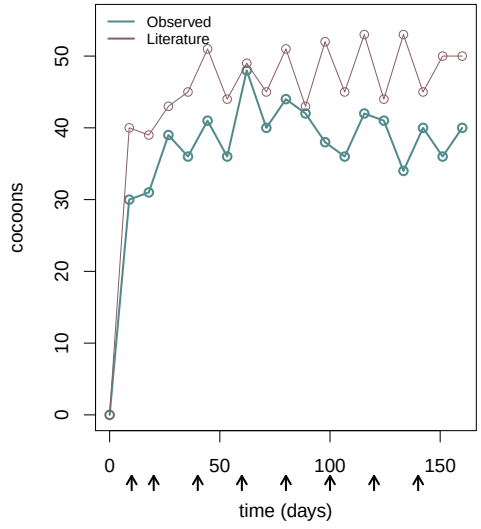
FIGURE 4.10: Continued overleaf



(D) Data taken from Reinecke and Viljoen [40]. Cocoons were counted, recorded and removed from the container every ten days. Manure was added to the container on days 0, 10, 60 and 140. On day 100 manure was removed from the container.



(E) Data taken from Reinecke and Viljoen [40]. The average mass of 10 juvenile earthworms was recorded every ten days for 160 days. Manure was added to the container on days 10, 20, 40, 60, 80, 100, 120 and 140.



(E) Data taken from Reinecke and Viljoen [40]. Cocoons were counted, recorded and removed from the container every ten days. Manure was added to the container on days 10, 20, 40, 60, 80, 100, 120 and 140.

FIGURE 4.10: The observed data, recorded in the field experiments is shown in blue. The output of the model when run at literature values is given in pink. The arrows on the x axis denote food being added (up) or removed (down) from the container.

In practice, model calibration is not straightforward. IBMs and other models of interest are often high dimensional and highly stochastic which poses problems for ABC. An additional hurdle which is commonly faced when using models of complex systems is that the experimental data is rarely reproducible by the model. Indeed this is the case for the *Eisenia Fetida* model considered here. It is a difficult task to perfectly model a complex system, and such a task is rarely completed due to a lack of time, or a lack of knowledge about the underlying processes of the system. A proposed solution to dealing with the disparity between the model output and the observed data is given by Goldstein and Rougier [41], who suggest the introduction of an additional probabilistic model, which maps from the model output to the experimental data, and thus produces data which fits the model. Here we simply consider the use of ABC to calibrate the model at hand, and do not account for the lack of fit of the experimental data to the model output, though it is discussed in our results.

This model has previously been analysed using ABC in van der Vaart et al. [42]. The paper aimed to give insights into how ABC can be used for the calibration and evaluation of IBMs, with the specific application to the *Eisenia fetida* model. The data and code used to generate the results given in the paper are available from figshare repositories for the paper [43].

Upon carrying out the work presented in this section, I found that the data in the figshare, and thus the data used to obtain the results in the paper, is not sampled from the correct prior distribution as stated in the paper. The file containing sampled parameter values contains 1×10^6 rows, with each row containing 14 values, sampled from independent log-normal priors. However, the parameter values have been truncated to between four and six significant figures before being used to run the model. Because the scales of some of the parameters are so small (e.g. M_b is of order 10^{-2}), this truncation causes many of the sampled values for a specific parameter to be rounded to the same value. Consequently, the sample from the prior distribution does not contain as many independent samples as expected. For example, for parameters E, M_b, M_c and s, the model was run at only 926, 447, 598 and 176 unique values respectively. This lack of variation in the parameter values leads to a lack of variation in the prior predictive distribution: the resultant summary statistics are also limited to a smaller number of outputs than would be expected under the correct, continuous prior distribution.

Because of this reduction in the number of independent samples in the parameter values, we cannot be certain that the posterior distributions given in the paper and the results of subsequent analysis carried out is reflective of the true posterior distributions. The authors state that of the 14 parameters, 7 had posteriors which, marginally, were significantly narrowed at a level of $\alpha = 0.01$. In Section 4.3.3 we show that, for correctly sampled data, no marginal posteriors are significantly narrowed at a level of $\alpha = 0.01$, when posteriors are obtained using Rejection ABC.

The impact of sample size on p-values is also considered in Section 4.3.3. We show that, by changing the size of the ABC posterior distribution, it is possible to obtain p-values which indicate that 10 of the 14 parameters have ABC posterior distributions which are marginally narrowed, compared to the prior distribution.

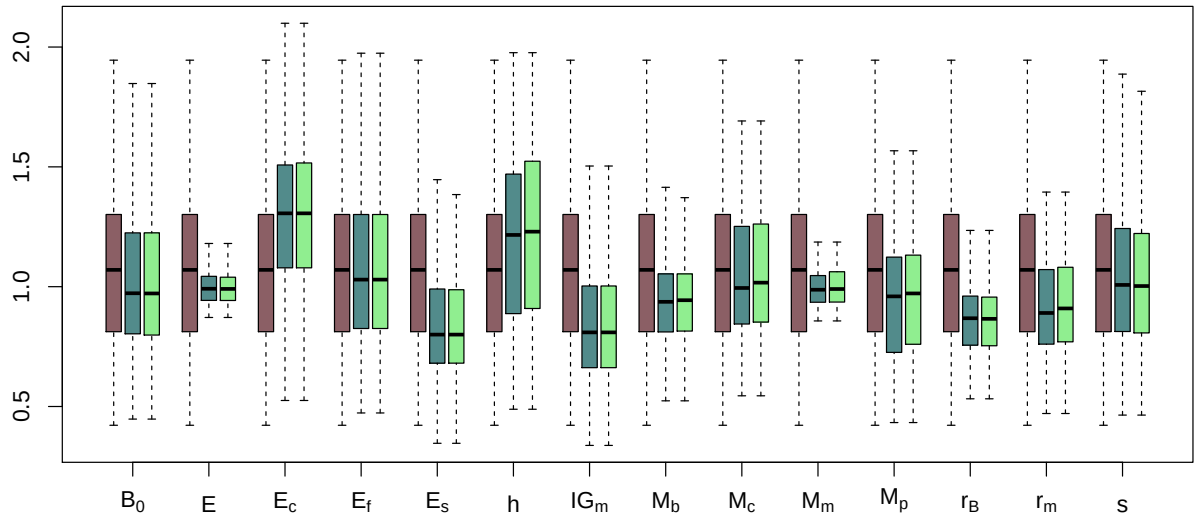


FIGURE 4.11: Boxplots of prior distributions (pink, left) and posterior distributions for Rejection ABC (blue, middle) using non-truncated parameter values, and Rejection ABC (green, right) using truncated parameter values in the *Eisenia fetida* model. Parameter values have been scaled by the literature value.

In Figure 4.11 we plot the marginal posterior distributions obtained under Rejection ABC on an experimental data set. The pink bars represent a sample from the prior distribution, whilst the blue and green bars denote ABC posterior distributions. Both the blue and green bars are based on the same set of 1×10^6 parameters drawn from the prior distribution. However, the

green bars correspond to the posteriors obtained when the parameter values are truncated and the corresponding summary statistics are truncated, as was the implementation in van der Vaart et al. [42]. It is clear from the plot that the posterior distributions for the truncated and non-truncated methods look similar, but are not identical. This similarity is not surprising, since there is little difference between the range of parameter values in the two samples.

This work begins by applying ABC to pseudo-observed data, which has been simulated under the IBM. Because we have access to the true parameter values which simulated the data, we can quantitatively measure the accuracy of the posterior distributions obtained. We compare three algorithms: Rejection ABC, SMC-ABC and Auto-SS SMC-ABC as described in Section 3.4.6.

Because the experimental data does not fit the model (see Figure 4.10), none of the regression based methods, including Auto-SS SMC-ABC, will work well on the experimental data. Many observed summary statistics lie outside of the range of the posterior predictive distribution, and hence any regression method leads to extrapolation. Thus only SMC-ABC and Rejection ABC are implemented on the experimental data. The methods are compared in terms of computational cost.

4.3.1 Pseudo-Data

For all ABC algorithms implemented in this section, parameters are drawn independently from log-normal distributions, with a mean equal to the literature values (given in Table 4.5) and a standard deviation of 0.359. This prior distribution was also used in van der Vaart et al. [42] and was selected by the authors since such a prior ensures that 95% of the simulated parameter values lie between half and twice the literature values.

We simulated 50 pseudo-observed data sets to assess the merits of different ABC algorithms. These are generated by sampling parameters from the prior distribution, then simulating data from the model.

We apply the following three ABC algorithms to the pseudo-data:

1. **Rejection ABC** Following the methodology of van der Vaart et al. [42], simulated data is compared to the observed data by taking the scaled Euclidean distance between the

two samples, where scaling is by the marginal standard deviation of the prior predictive distribution. The ABC posterior distribution is taken to be the 100 points that are closest to the observed data. The method is implemented in the following two ways:

(a) 1×10^6 parameter values are sampled from the prior distribution.

(b) For each pseudo-observed data set the number of simulations from the model required to carry out inference using SMC-ABC, as described below, is counted. This number of parameters is then sampled from the prior distribution. Implementing the algorithm in this way enables a direct comparison to be made between SMC-ABC and Rejection ABC, for the same computational cost.

2. **SMC-ABC** The SMC-ABC algorithm of Del Moral et al. [1] is implemented, using $N = 1000$ particles, and $\alpha = 0.99$. The algorithm gives a posterior distribution containing 1,000 particles, however, all results reported here are based on the 100 of these 1000 parameter values which produced simulated data that was closest to the observed data. Such a choice enables a direct comparison to be made with the output of Rejection ABC. SMC-ABC was not run with $N = 100$ particles because it is known that the behaviour of such particle methods is poor for a small number of particles. The algorithm is ran twice on each pseudo-observed data set, once using each of the following stopping rules:

   (a) The stopping rule given in Algorithm 5 is used.

   (b) The algorithm is stopped after the minimum of (i) 200,000 simulations of the model and (ii) the number of iterations until the algorithm stopped using the stopping rule given in Algorithm 5.

3. **SMC-ABC with Summary Statistic Selection at** $t = 0$ At time $t = 0$, summary statistics are selected using the Automatic Selection method of Fearnhead and Prangle [22], based on a sample of $10,000$ draws from the prior predictive distribution. SMC-ABC is then implemented as above, using these summary statistics, with stopping rule (a).

4. **Auto-SS SMC-ABC** Algorithm 8 is implemented, with $N = 1000$ particles and $\alpha = 0.9$. Again results are reported based on the 100 particles which are closest to the observed data, so that a direct comparison with Rejection ABC can be made.

4.3.2 Results

Figure 4.12 shows the posterior means for each of the parameter values against the true parameter value which simulated the data, for SMC-ABC with stopping rule (a), Rejection ABC based on 1×10^6 simulations, and Auto-SS SMC-ABC. The values of the Pearson correlation coefficient, r, for each of the two methods, which is a measure of the correlation between the posterior means and the truth, is also given in the figure.

The correlation coefficient for E, IG_m, M_b, M_m, M_p and r_B are all greater than 0.8 (under the three ABC methods), meaning that there is a strong positive correlation between the true parameter value and the estimated posterior mean. This indicates that the summary statistics of the model hold information about these parameters. Of these six parameters, Auto-SS SMC-ABC gives a larger r value for parameters E, M_p and r_b, SMC-ABC gives the largest r values for IG_m and M_m, and Rejection ABC outperforms the other two methods, in terms of r, when inferring M_b. This implies that, when a parameter is inferable, both Auto-SS SMC-ABC and SMC-ABC perform better than Rejection ABC for all of the parameters except M_b.

Parameters E_c, h, M_c and r_m have r values in the range of 0.35 to 0.75, suggesting that there is a small amount of information about these parameter values held in the summary statistics. The final four parameters, B_0, E_f, E_s and s, have values of $r < 0.35$, suggesting that these parameters are not inferable from the model's summary statistics. However, for parameter B_0, Auto-SS SMC-ABC out performs the other methods and, with an r value of 0.326, indicates that there may be some information about B_0 held in the summary statistics.

Table 4.6 gives the percentage difference in MRSSE for the 14 parameters of the *Eisenia fetida* model. Values given are relative to the MRSSE of Rejection ABC using 1×10^6 simulations. For all but two parameters, the smallest MRSSE was obtained by either SMC-ABC with the stopping rule given in Algorithm 5, or by Auto-SS SMC-ABC. Surprisingly, for r_m and s, the best results were obtained using the SMC-ABC (b) algorithm which uses fewer simulations from the model than SMC-ABC (a).

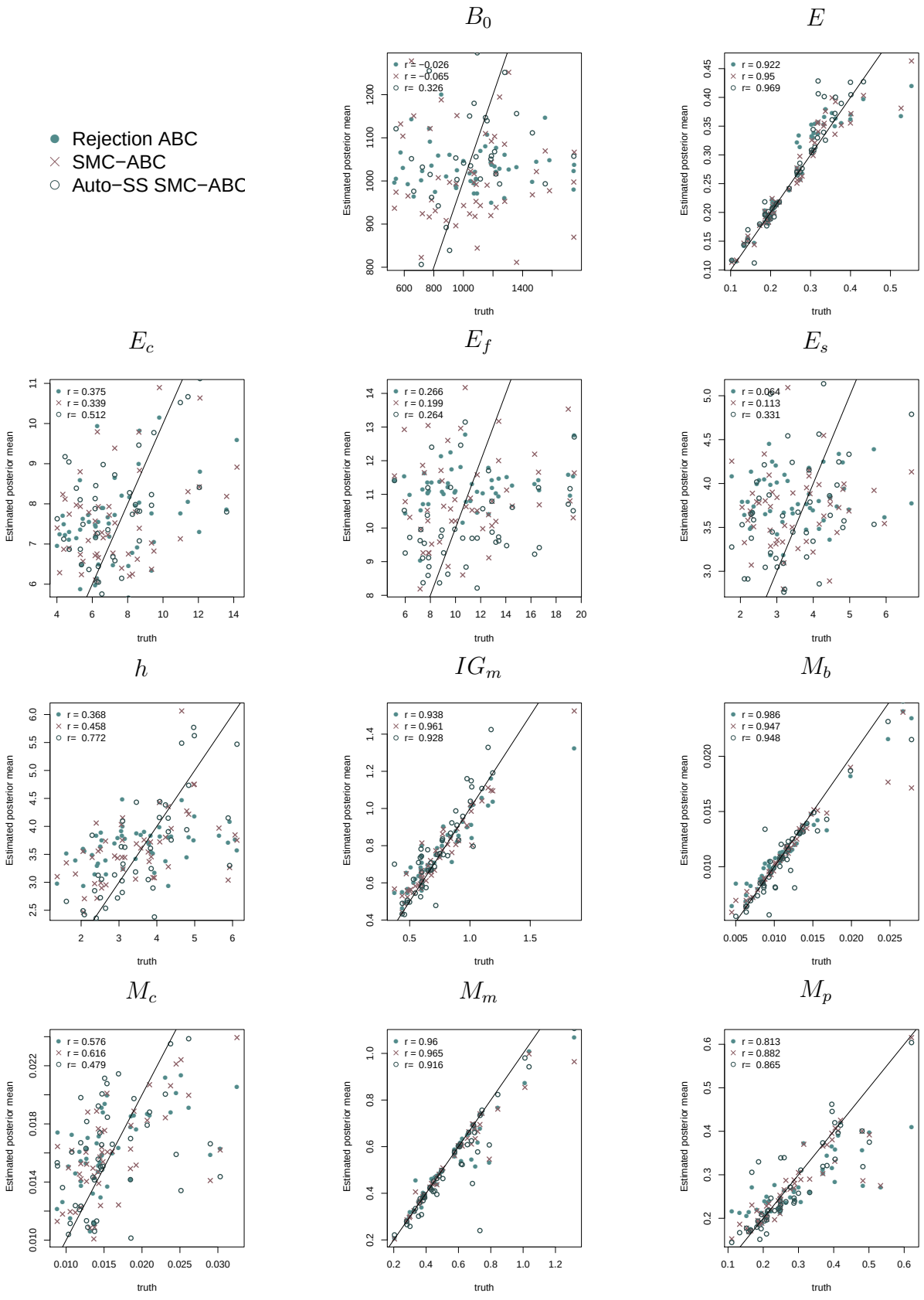


FIGURE 4.12: (continued overleaf)

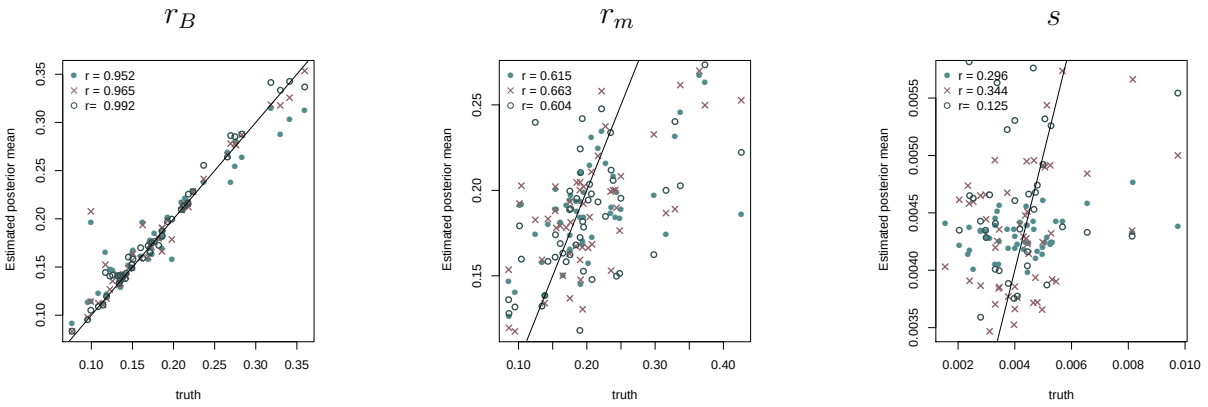


FIGURE 4.12: Estimated posterior mean plotted against true parameter value for 50 pseudo-data sets. Results are given for Rejection ABC (green dots), and SMC-ABC (pink crosses). Posteriors are based on the 100 acceptances, and r is the Pearson correlation coefficient for the posterior distributions.

Parameter	Rejection ABC *number of simulations*		SMC-ABC *stopping rule*		Auto-SS SMC-ABC	SMC-ABC *ss selection at* $t = 0$
	1×10^6	as in SMC	(a)	(b)		
B_0	0	2.28	5.09	−3.25	**-16.15**	3.65
E	0	5.04	−42.15	−28.60	**-52.06**	40.04
E_c	0	0.83	**-14.36**	−4.10	-12.69	10.64
E_f	0	−0.40	−13.29	−8.63	**-15.68**	−11.39
E_s	0	−0.96	**-10.36**	−6.40	-8.79	4.417
h	0	3.59	−21.68	−11.50	**-33.97**	−17.23
IG_m	0	10.16	**-47.30**	−27.95	-37.19	13.63
M_b	0	17.96	**-56.64**	−34.34	-23.23	36.89
M_c	0	0.97	**-20.17**	−9.44	-1.41	9.52
M_m	0	13.63	**-49.26**	−28.00	-12.17	148.55
M_p	0	7.22	**-46.61**	−41.33	-25.85	18.91
r_B	0	12.84	−59.91	−39.12	**-66.27**	64.41
r_m	0	−0.69	−3.23	**-6.33**	-5.01	−0.69
s	0	0.08	−1.76	**-1.95**	35.82	4.11

TABLE 4.6: Relative MRSSE of posterior distributions, given in terms of percentage difference from the Rejection ABC result obtained using 1×10^6 samples. All posterior distributions are based on the closest 100 data sets. The results in the *as in SMC* column are obtained using the same number simulations from the model as was used in SMC-ABC. Note that this changes for each pseudo-data set. The smallest value in each row is given in bold.

However, Figure 4.12 shows that these parameters are poorly estimated by all methods. SMC-ABC with summary statistic selection at time $t = 0$ did not give rise to the lowest MRSSE for any parameter value. This suggests that the initial sample of size $N = 10,000$, which we used to guide the summary statistic selection, was not large enough.

This example is extremely high dimensional, with 160 summary statistics and 14 parameters. Linear regression on such a high dimensional space is likely to perform poorly. This is a possible cause of the unfavourable results given by the SMC-ABC algorithm with summary statistic selection at iteration $t = 0$ performs poorly.

The Auto-SS SMC-ABC algorithm performs well for some parameters (e.g. B_0, E, E_f), but badly for others (e.g. M_m, M_c, M_b), and inference for s is extremely poor. Again this may be caused as a result of carrying out linear regression on a high dimensional space. Furthermore the summary statistics used for inference in this algorithm are estimates of the posterior mean. The posterior distributions for the parameters of the Earthworms model are on very varied scales. To account for this, we scale the projected summary statistics by the marginal standard deviation of the projections, based on a preliminary sample from the prior predictive distribution. However, it is likely that these initial projections are poor, and thus the scaling is not reflective of the true posterior standard deviation, which is the desired scaling factor, as noted by Prangle [24]. This non-optimal scaling leads to the distance metric being more influenced by certain parameters, hence explaining the range of goodness of results.

For the parameters E, IG_m, M_b, M_m, M_p and r_B, which Figure 4.12 suggest are recoverable from the summary statistics, SMC-ABC with stopping rule (a) gives rise to a MRSSE that is at least 40% smaller than that obtained through standard Rejection ABC with 1×10^6 samples. This illustrates that there is often a huge gain in accuracy of inference when SMC-ABC is used.

We compare the computational cost of the SMC-ABC algorithm to that of Rejection ABC in two ways. First, we implement Rejection-ABC with the same number of model simulations as was needed in the SMC-ABC (a) algorithm. The number of simulations used in the SMC-ABC (a) algorithm varied across the pseudo-data sets, and we varied the number of samples used in the Rejection ABC implementation accordingly. By comparing the relative MRSSE values given in column 3 and 4 of Table 4.6, we see that Rejection ABC with the limited number of samples performs much worse than SMC-ABC with the same number of samples. This is expected, since the Rejection ABC samples are drawn from the full prior, whereas SMC-ABC moves over time to sample particles from distributions which are closer to the posterior distribution.

The second comparison of computational cost we make is to run SMC-ABC, but cap the number of model simulations at 200,000. Denoted SMC-ABC (b), we use the distribution of parameters after 200,000 model simulations as the posterior distribution. For two pseudo-observed data sets, the conditions for the stopping rule in SMC-ABC (a) were met after fewer than 200,000 model simulations. Thus the smaller number of simulations was used in these cases. Column 5 of Table 4.6 shows that restricting simulations to 200,000 produces posterior distributions with much lower MRSSE than implementing Rejection ABC with 1,000,000 simulations, and hence is more efficient.

Figure 4.13 gives boxplots showing the number of model simulations required to implement the different SMC-ABC methods. The figure shows that the implementation of SMC-ABC with summary statistics selected at time $t = 0$ requires the fewest model simulations, though this method gave the worst results of the methods presented here. This was caused by the algorithm terminating early, due to poorly selected summary statistics through the initial regression. The Auto-SS SMC-ABC algorithm with 1 step rejection required the most simulations, however the 2 step rejection (Algorithm 6) reduces the number of model simulations to the same order as the number required for SMC-ABC with stopping rule (a).

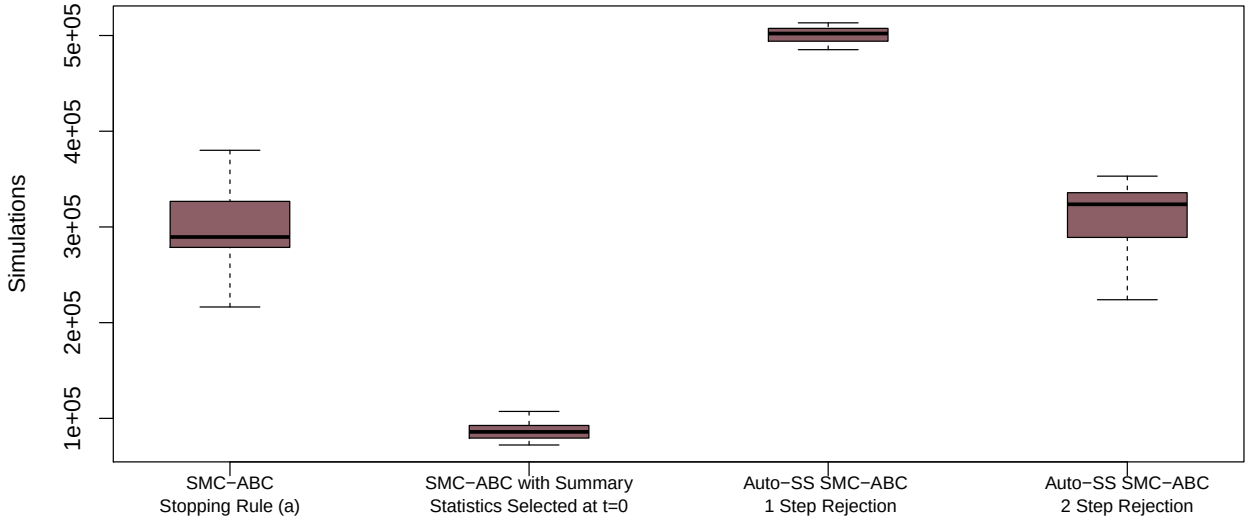


FIGURE 4.13: Boxplots Number of simulations from the model needed for a range of ABC implementations on the Earthworms model.

4.3.3 Implementing ABC on Experimental Data

In this section ABC methods are applied to the experimental data, which is plotted in Figure 4.10. Regression based methods are not applied to this data set, since the model does not fit the data, and thus they lead to extrapolation.

Rejection ABC, using the same prior distributions as in Section 4.3.1 is implemented on the experimental data. Again results are based on 1×10^6 simulations from the prior distribution and the posterior distribution is taken as the 100 parameters which simulated the closest data to the data recorded in the experiment. Figure 4.14 shows the 10 simulated data sets which were closest to the observed data, in terms of mean squared error. The Figure shows that there is a disparity between the simulated and experimental data. Specifically, the simulated data does not capture the rates of increase or decrease in the observed data.

SMC-ABC is also implemented, using the stopping rule in Algorithm 5. As with the pseudo-observed data, SMC was run with $N = 1000$ particles, but reported posterior distributions are based on the 100 particles from the SMC posterior which lie closest to the experimental data. This allows us to make direct comparisons with the Rejection ABC posteriors.

The quantity R^2 is used to assess how well the empirical data is replicated by the ABC posterior predictive distributions. It does this by providing a measure of the proportion of variance of the output which is explained by a model. As in van der Vaart et al. [42], here the mean R^2 is computed for the posterior distributions from Rejection ABC and SMC-ABC across all 6 experiments.

Let $s_1, s_2, \ldots, s_n$, be the observed outcomes of an experiment, and let $d_{1,j}, d_{2,j,} \ldots, d_{n,j}$ be the jth closest simulated data set, where $j \in 1, \ldots, 100$. Then the value of R^2 is given by

$$R^2 = 100^{-1} \sum_{j=1}^{100} \left(1 - \frac{\sum_{k=1}^{n} (s_k - d_{k,j})^2}{\sum_{k=1}^{n} (d_{k,j} - \bar{d}_j)^2} \right), \tag{4.18}$$

where

$$\bar{d}_j = n^{-1} \sum_{i=1}^{n} d_{i,j}. \tag{4.19}$$

In the event that all of the 100 simulated data sets match the experimental data exactly, the value of R^2 will equal 1. A value of R^2 near 1 suggests that the simulation results are good estimates of the observed data. In our application, this implies that the samples in the posterior predictive distribution are similar to the observed data. An R^2 equal to 0 suggests that the model does not explain any of the variation in the output and a negative value of R^2 suggests that the model fits the data very poorly and that a better fit, in terms of minimising summed squared error, would have been achieved by replacing the model by a horizontal line.

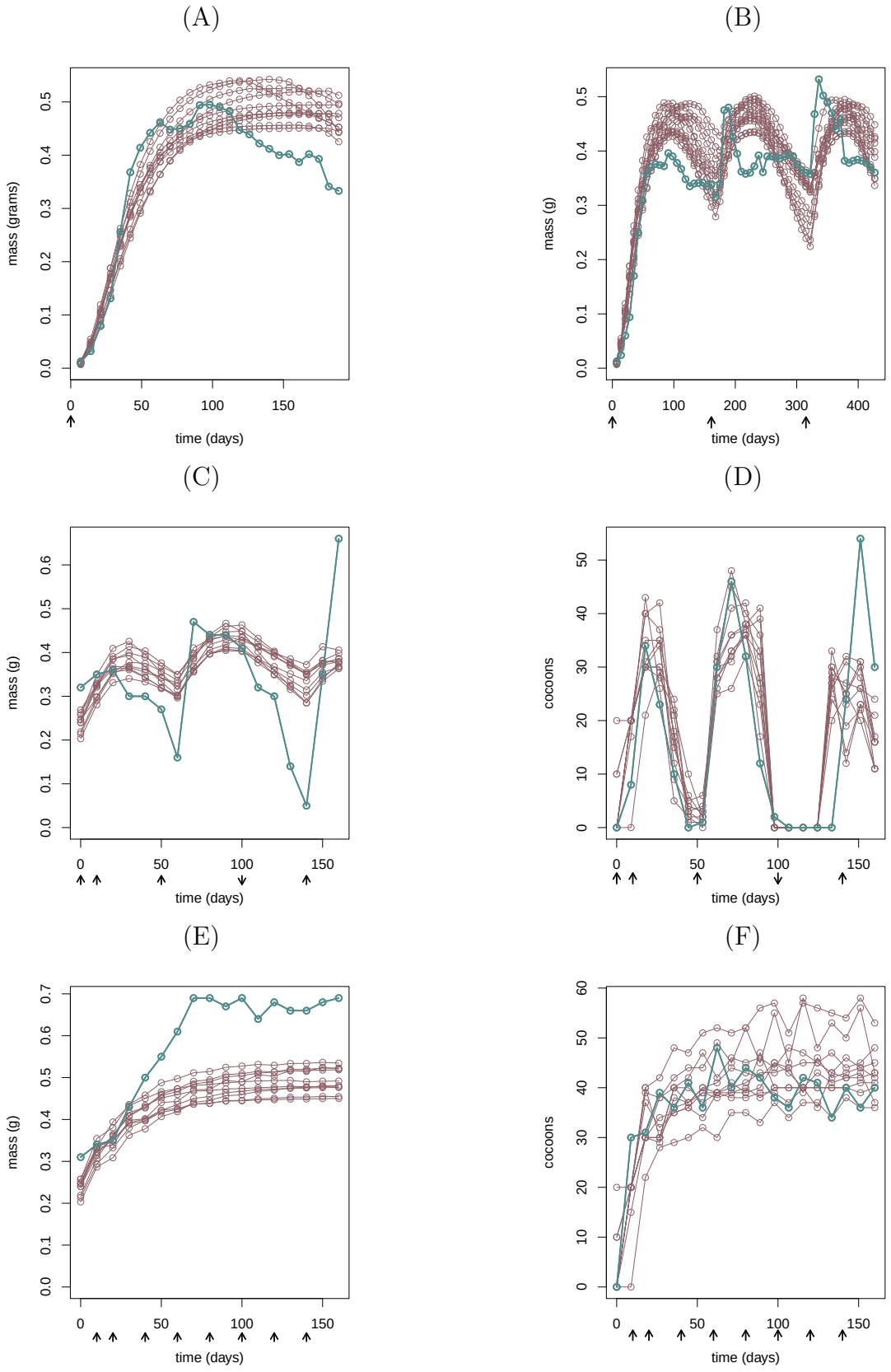


FIGURE 4.14: Experimental data (blue), and the ten closest runs from Rejection ABC (pink), where closeness is measured in terms of mean squared error.

Experiment	Rejection ABC	SMC-ABC	Literature Values
(A)	0.73	0.91	0.86
(B)	0.43	0.56	0.58
(C)	0.87	0.83	0.82
(D)	-0.40	-0.43	-0.53
(E)	0.43	0.30	0.04
(F)	0.06	0.24	0.17

TABLE 4.7: Mean R^2 values for the 6 experiments in the *Eisenia fetida* model, based on Rejection ABC, SMC-ABC and running the model at the literature values. All ABC results are based on 100 samples from the posterior distribution.

Table 4.7 gives the mean R^2 for the 100 samples in the Rejection ABC posterior, and SMC-ABC posterior, as well as the value of R^2 obtained when the model is run at the values reported in the literature. These values of mean R^2 suggest that the posterior distribution obtained through Rejection ABC fits the experimental data better than the literature values in experiments (B), (E), and (F). SMC-ABC fits better than the literature for experiments (A), (E) and (F). Comparing Rejection ABC and SMC-ABC we see that the SMC-ABC posteriors better explain the results for experiments (C), (D), (E) and (F), but worse for (A) and (B). The R^2 values for experiment (D) are negative for both ABC algorithms, and for the literature values. This emphasises the poor fit of the simulated data to the model data for this experiment.

Figure 4.15 shows the experimental data, plotted against the mean of the posterior predictive distribution obtained through Rejection ABC and through SMC-ABC. For both algorithms the mean over the 10 closest simulations in the posterior predictive distribution are plotted. The figure suggests that there is little difference between the posterior predictive mean obtained from SMC-ABC and from Rejection ABC. The main noticeable difference is in panel (F), where Rejection-ABC gives a stable incline in number of cocoons, while SMC-ABC fluctuates more, as is the pattern in the experimental data. However, the peaks and troughs of the experimental data and the SMC-ABC posterior do not appear to coincide, so evaluating the methods based on panel (F) is not straightforward.

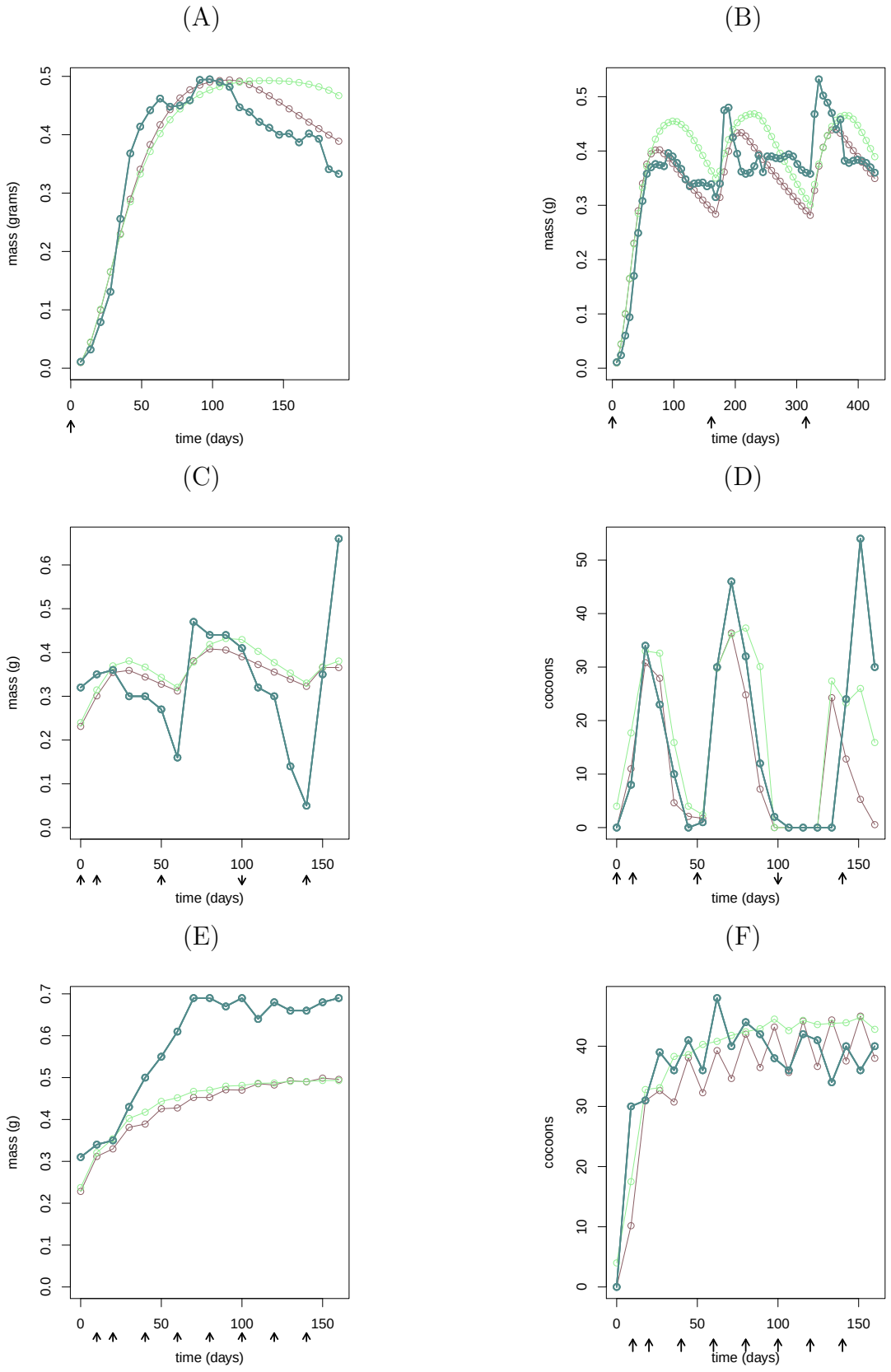


FIGURE 4.15: Experimental data (blue), the average of the ten closest runs from Rejection ABC (green), and average of the ten closest runs from SMC-ABC (pink), where closeness is measured in terms of mean squared error.

Figure 4.16 shows boxplots of the marginal posterior distributions obtained using the different ABC methods. As in van der Vaart et al. [42] the parameter values are scaled by their literature values (given in Table 4.5) so that all parameters can be viewed on the same scale. Figure 4.16 shows that the parameters that show greatly narrowed posterior distributions, compared to the prior distributions (shown in pink), are E, M_b, M_m, M_p and r_B. Note that our analysis of the pseudo-data earlier in the chapter showed that these parameters were all inferable from the summary statistics. In Figure 4.16 the posterior distributions for Rejection ABC (blue) appear wider than those for SMC-ABC (green) for these parameters in particular.

The Figure also shows that the posterior distributions for B_0, E_c, E_f, E_s and s do not appear to be narrowed from the prior distributions. This is in fitting with the results in Figure 4.12 which showed that these parameters were poorly estimated for pseudo-data.

The parameter with the biggest discrepancy between the posterior means under Rejection ABC and SMC-ABC is h. Table 4.6 shows that the MRSSE was 21.68% smaller for h when the posterior was obtained by SMC-ABC, as compared to Rejection ABC. Figure 4.12 indicates that h is recoverable. Thus we have no reason to disbelieve the larger posterior mean for h, obtained under SMC-ABC shown in Figure 4.16.

The next analysis carried out in this section is hypothesis testing, to test for significant narrowing between the prior and the posterior distribution. Again, this was implemented in van der Vaart et al. [42], using Levene's Test [44], and correcting for multiple testing using the Holm-Bonferroni method [45]. The p-values for each of the parameters, and each of the methods are given in Table 4.8. We give two sets of p-values for SMC-ABC. The first is obtained using a sample of size 100 from the posterior distribution, with the 100 parameters chosen to be those which simulated data which lies closest to the experimental data, as was done for all other SMC-ABC results previously reported in this section. The second column contains the p-values obtained when we use the full posterior sample of size 1,000.

After accounting for multiple testing using the Holm-Bonferonni method, none of these p-values based on the posterior distribution obtained from the Rejection ABC algorithm is significant at the $\alpha = 0.01$ level. This is contrary to the results given in van der Vaart et al. [42], where 7 parameters were deemed to have posteriors which were significantly narrowed at the $\alpha = 0.01$

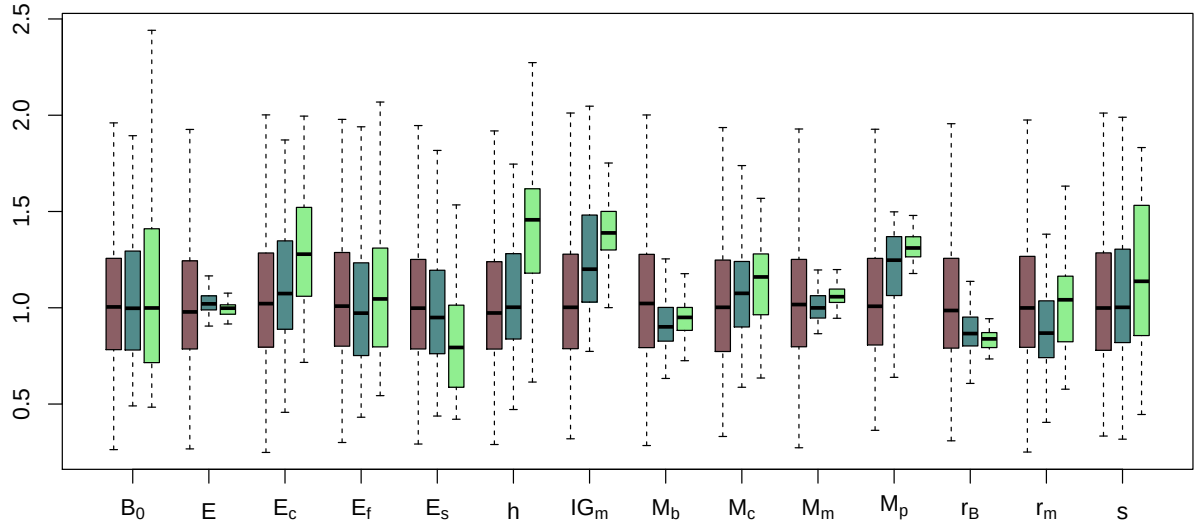


FIGURE 4.16: Box plots of prior distributions (pink) and posterior distributions for Rejection ABC (blue), and SMC-ABC (green) for the 14 parameters of the *Eisenia fetida* model. Posterior distributions are generated using Rejection ABC. Parameter values have been scaled by the literature value.

level. The SMC-ABC posteriors also shows no significant narrowing. However, when we carry out the same hypothesis test on the same SMC-ABC posterior, but with a sample size of 1,000, as opposed to 100, we see that 6 of the parameters have posterior distributions which narrow significantly from the prior distribution. The reason that more posterior distributions are deemed significantly narrowed from the prior when using more samples from the same distribution is due to an increase in power: The significance test is more certain that the posteriors are narrowed since it has more data points to base this decision on.

It is well known that p-values vary with the size of the sample on which they are being tested. With increasing sample size comes increasing confidence in the decision to reject or accept the alternative hypothesis. In ABC the choice of sample size is quite a conundrum. Ideally we would use a large sample size to test any hypothesis. However, suppose we are carrying out Rejection ABC using a fixed, finite number of samples from the prior distribution. In order to increase the number of samples in the ABC posterior distribution, the level of approximation in the posterior must also be increased, since the tolerance level ϵ must be increased. Thus there is a trade-off between the sample size, and the accuracy of the ABC posterior distribution.

Parameter	Rejection ABC	SMC-ABC (100 samples)	SMC-ABC (1000 samples)
B_0	0.882	0.963	0.002
E	0.010	0.001	1.301×10^{-23} *
E_c	0.999	0.094	0.105
E_f	0.870	0.639	0.390
E_s	0.293	0.957	0.075
h	0.444	0.564	0.323
IG_m	0.287	0.011	2.657×10^{-11} *
M_b	0.054	0.005	6.855×10^{-18} *
M_c	0.404	0.071	2.718×10^{-4}
M_m	0.009	0.002	1.136×10^{-21} *
M_p	0.277	0.004	5.062×10^{-19} *
r_B	0.025	0.002	2.279×10^{-21} *
r_m	0.126	0.077	0.003
s	0.722	0.483	0.426

TABLE 4.8: The p-values for the testing of the narrowing of the posterior distributions across Rejection ABC and SMC-ABC, based on both 100 and 1,000 samples from the posterior distribution. Values marked with an asterisk are deemed significant at the $\alpha = 0.01\%$ level.

In the Rejection ABC example given above, with a sample of size 100 from the ABC posterior, none of the 14 parameters were significantly narrowed from the prior distribution at the level $\alpha = 0.01$. In Figure 4.17 we plot the relationship between the posterior sample size and number of p values which are deemed significant at the level $\alpha = 0.01$. Note that no extra simulations are run to obtain these results: a sample size of n is obtained by taking the n points from our sample of 1×10^6 which are closest to the observed data. Thus in the limit as we allow the posterior sample size to be increased to 1×10^6, the posterior distribution is identical to the prior distribution, and it is not possible for the posterior distribution to be narrowed, compared to the prior. Figure 4.17 shows that, as the posterior sample size is increased 0 to 300,000, the number of significantly narrowed posteriors increased from 0 to 10. From this point, further increases in number of samples that are included in the ABC posterior leads to an increased tolerance thus the number of significantly narrowed parameters decreases. Surprisingly, for a posterior sample size of $975,000$ the parameter M_m is still deemed to have a posterior distribution which is significantly narrower than the prior. This is despite the prior containing only the same samples as the posterior as well as 25,000 additional points.

For SMC-ABC we saw in Table 4.8 that increasing the sample size from 100 to 1000 resulted in more posterior distributions being deemed significantly narrowed. If the number of samples

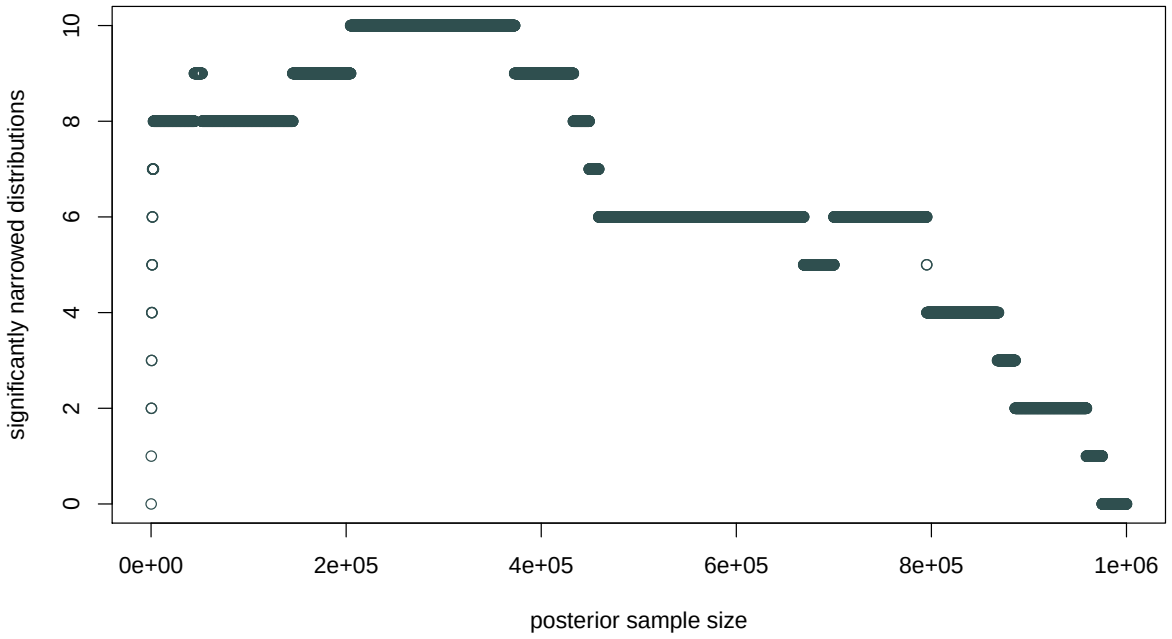


FIGURE 4.17: Number of parameters deemed to have posteriors that are significantly narrowed at the level $\alpha = 0.01$, over varying posterior sample sizes. These results are based on Rejection ABC, with a prior sample size of 1×10^6.

taken from the SMC-ABC posterior is increased, through resampling the particles, we expect this number of significant p values to increase further.

The sensitivity of p-values to sample size is such that they should not be solely relied on to draw conclusions. However, combining this information with Figure 4.16, we are confident that the posterior distributions for E, M_b, M_m, M_p and r_b are notably narrowed, compared to the prior distributions.

The results of the analysis on pseudo-observed data, given in in Section 4.3.1, showed that SMC-ABC performed more accurately than Rejection ABC, when trying to infer parameter values. For this reason, we believe that the posterior distribution obtained when SMC-ABC is applied to the experimental data is our 'best approximation' of the true posterior distribution.

4.4 Population Growth Model

The first practical application of Rubin's ABC thought experiment [2] was given in Pritchard et al. [3], where ABC was used to infer coalescence times from DNA sequence data. Many papers which apply ABC to population genetics models have been published since [46] [47].

In this example we use ABC methods to infer parameter values for data which is simulated under a model of population growth. We begin by introducing the data and summary statistics, before going on to discuss the population growth model and implementing ABC methods.

We use an abundance of summary statistics to summarise the data, with the aim of emulating the actions a scientist who is unsure of which statistics are informative for the parameters of interest. In the next section we give a sample data set and show how we compute the summary statistics for this data.

4.4.1 Haplotype Data and Summary Statistics

As is common in population genetics applications, our data is of a form known as haplotype data. Haplotype data is generated by first taking a sample from a section of a genome, then encoding this data as a string of 0s and 1s. It is this coded string that we refer to as a haplotype. By sampling haplotypes from the same position on the chromosome across multiple cells, or members of a population, the individuals can be compared and contrasted, and the haplotypes can be used to gain an understanding of the population as a whole.

Figure 4.18 shows five samples of haplotype data. Each haplotype is of length four, and thus holds the value 0 or 1 in each of the four positions. These numbers represent information about the state of the gene at a particular location, compared to the state of a sample from the ancestral

```
h1: 0000
h2: 0110
h3: 0001
h4: 0001
h5: 1100
```

FIGURE 4.18: Example of Haplotype Data, simulated using Hudson's ms. The data shows 5 haplotypes with 4 segregating sites.

population at the same location: A '0' denotes that the gene is in the ancestral state at that given position, whereas a '1' is used to represent a gene in the derived state, meaning that a mutation has occurred. We assume an infinite sites model, meaning that mutations can occur at any location along a genome, and no two mutations happen at the same position. These positions at which a mutation has occurred are known as *segregating sites*. Thus the length of a haplotype string is equal to the number of *segregating sites* in the sampled section of the gene.

With these new definitions in hand, we are able to say that the sample in Figure 4.18 shows five haplotypes and contains four segregating sites. The first haplotype is in the ancestral state at each segregating site, whereas the second is in the ancestral state at the first and last segregating site, and in the derived state at the second and third segregating sites.

Table 4.10 gives the 19 summary statistics which we will use in this population genetics example, including the number of segregating sites, and number of distinct haplotypes.

To aid the computation of the summary statistics numbered 3 to 6 in Table 4.10 we create a frequency table for the haplotype data. This is given in Table 4.9. From Table 4.9 we see that the most common haplotype appears twice and the second most common haplotype appears once. Thus $F_{[1]} = 2$ and $F_{[2]} = 1/2$. The median frequency of haplotypes, $F_{[med]}$, is 1, and the rarest haplotype has frequency of 1, hence $F_{[min]} = 1$.

haplotype	frequency
0000	1
0110	1
1100	1
0001	2

TABLE 4.9: Frequency Table for the haplotype data given in Figure 4.18.

	Symbol	Description
1	n	number of distinct haplotypes
2	$\mathcal{S}$	number of segregating sites
3	$F_{[1]}$	Frequency of the most common haplotype
4	$F_{[2]}$	Frequency of second commonest haplotype divided by frequency of most common haplotype (0 if population is monomorphic)
5	$F_{[med]}$	median frequency of haplotypes
6	$F_{[min]}$	frequency of the rarest haplotype
7	v_1	Number of sites with one variant
8	v_2	Number of sites with two variants
9	v_3	Number of sites with three variants
10	v_4	Number of sites with four variants
11	v_5	Number of sites with five variants
12	v_6	Number of sites with six variants
13	v_7	Number of sites with seven variants
14	v_8	Number of sites with eight variants
15	v_9	Number of sites with nine variants
16	v_{10}	Number of sites with ten variants
17	π	mean pairwise difference across haplotypes
18	H	Fay and Wu's H.
19	D	Tajima's d

TABLE 4.10: Summary Statistics used in the Exponential Growth Example.

The summary statistics numbered 7 to 16 in Table 4.10 all relate to the segregating sites in the haplotypes. The number of sites with i variants is defined as the number of segregating sites at which there are exactly i haplotypes in the derived state. This corresponds exactly to the number of segregating sites at which $n - i$ haplotypes are in the ancestral state, coded '0'. In the sample in Figure 4.18 there are four segregating sites, and the number of variants at each of these sites are given, from left to right, by 1, 2, 1 and 2. Thus two sites have one variant, two sites have two variants, and no sites have three or more variants.

Haplotype Pairs, h_i, h_j	Pairwise Difference, $\pi_{i,j}$
h_1, h_2	2
h_1, h_3	1
h_1, h_4	1
h_1, h_5	2
h_2, h_3	3
h_2, h_4	3
h_2, h_5	2
h_3, h_4	0
h_3, h_5	3
h_4, h_5	3
total	20

TABLE 4.11: Pairwise difference, $\pi_{i,j}$ for all pairs of haplotypes h_i, h_j given in Table 4.11.

Also known as *Tajima's estimator of* θ, summary statistic π denotes the mean number of pairwise differences between haplotypes in the sample. Each pair of haplotypes is compared at every segregating site in turn. Let $h_{i,j}$ be the state of the ith haplotype at the jth segregating site. For data with $\mathcal{S}$ segregating sites, and n haplotypes, the mean pairwise difference is given by

$$\pi = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \sum_{k=1}^{\mathcal{S}} |h_{i,k} - h_{j,k}| = \frac{2}{n(n-1)} \sum_{i,j} \pi_{i,j}, \tag{4.20}$$

where $\pi_{i,j}$ is defined as the pairwise difference between haplotypes i and j. To compute π for the data in Figure 4.18, 10 pairwise comparisons must be made. Table 4.11 gives the mean pairwise difference for each of the pairs of haplotypes. From Table 4.11 we deduce that, for this sample, π is equal to $20 \times 2/20 = 2$.

Summary statistic 18, Fay and Wu's H, is given by the difference between two quantities, both of which are estimates of θ:

$$H = \pi - \hat{\theta}_H. \tag{4.21}$$

The value of π is equal to the mean pairwise difference, as given in Equation (4.20), and

$$\hat{\theta}_H = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} i^2 \, v_i, \tag{4.22}$$

where v_i is the number of segregating sites with i variants.

For the sample of data given in Figure 4.18 we saw that $\pi = 2$, and that $v_1 = 2$, $v_2 = 2$ and $v_i = 0$ for all $i > 2$. Thus

$$\hat{\theta}_H = \frac{2}{5(5-1)}(1^2 \times 2 + 2^2 \times 2) = \frac{2 \times 10}{20} = 1, \tag{4.23}$$

and so it follows that

$$H = 2 - 1 = 1. \tag{4.24}$$

The final summary statistic in Table 4.10 is Tajima's D, which is computed by taking the scaled difference between two estimators of θ:

$$D = \frac{\pi - S/a_n}{\sqrt{e_1 \mathcal{S} + e_2 \mathcal{S}(\mathcal{S}-1)}} \quad for \quad H > 3, \tag{4.25}$$

where

$$a_n = \sum_{i=1}^{n-1} i^{-1}, \tag{4.26}$$

$$b_n = \sum_{i=1}^{n-1} i^{-2}, \tag{4.27}$$

$$e_1 = \frac{n+1}{3a_n(n+1)} - \frac{1}{a_n^2} \tag{4.28}$$

and

$$e_2 = \frac{1}{a_n^2 + b_n}\left(\frac{2(n^2+n+3)}{9n(n-1)} - \frac{n+2}{na_n} + \frac{b_n}{a_n^2}\right). \tag{4.29}$$

For $n = 5$, $a_n = 25/12$, and $b_n = 205/144$. Thus $e_1 = -44/625$ and $e_2 = 15/160$ and it follows that

$$D = \frac{2 - 4 \times 12/25}{\sqrt{\frac{-44}{625} \times 4 + \frac{15}{160} \times 4 \times 3}} = \frac{2/25}{\sqrt{0.8434}} = 0.0871. \tag{4.30}$$

We now introduce the model of population growth which is assumed for the rest of this example.

4.4.2 Population Growth Model

The simulated data for this example is generated using Hudson's ms [48], and takes the form of haplotype strings, as introduced in Section 4.4.1.

We simulate pseudo-observed data sets, under three types of population; one undergoing exponential growth, one undergoing exponential decay and one which is stable.

We take samples of 20 haplotypes from three loci and summary statistics are averaged across these. The small number of loci means that it is possible to use the Importance Sampling algorithm of Stephens and Donnelly [49] to obtain an exact approximation of the true posterior distributions. Thus we can compare ABC posterior distributions to this approximation.

The population growth model is defined by the following properties:

- at time the current time, the population is assumed to be of size N_c.
- t units of time ago, the population was of size N_0, then underwent exponential growth at rate α.
- time is measured in units of $2N_0$ generations throughout the process.
- an infinite sites model is assumed, meaning that at most one genetic mutation can occur at any position on the genome.

For this example, the model is parametrised by (θ, r, t_f), where

- $\theta = 4N_0\mu$ and μ is the mutation rate for the loci,
- t_f is the number of units, backwards in time, at which the population growth began, measured in $2 \times N_0$ generations,
- $r = \alpha \times t_f$ is the scaled growth rate.

θ	r	t_f	State of Population
20	20	0.05	expansion
1	0.05	1	contraction
10	1	1	stable

TABLE 4.12: Parameters used to simulate the three pseudo-observed data sets.

The following prior distributions are used:

$$\log \theta \sim \mathcal{U}(-5, 5) \tag{4.31}$$

$$\log r \sim \mathcal{U}(-5, 5) \tag{4.32}$$

$$\log t_f \sim \mathcal{U}(-5, 5). \tag{4.33}$$

4.4.3 Implementation Details

The parameter values used to simulate the pseudo-observed data are given in Table 4.12.

The following ABC algorithms are implemented on the three data sets:

- **Rejection ABC** $1,000,000$ data sets are simulated from the prior predictive distribution. We scale summary statistics by the median absolute deviation from the observed summary statistics, and use the Euclidean distance to measure the distance between a simulated and observed set of summary statistics. ABC posterior distributions are taken to be the $1,000$ parameter values which simulated data that lies closest to the observed data. This corresponds to an acceptance rate of 0.001.
- **SMC-ABC** We implement the SMC-ABC algorithm of Del Moral et al. [1] and use the stopping rule given in Algorithm 5. The algorithm is implemented with $N = 1000$, and $\alpha = 0.9$.
- **Auto-SS SMC-ABC** We implement Algorithm 8, with $N = 1,000$ and $\alpha = 0.9$.

4.4.4 Results

We now compare the ABC algorithms on each of the three data sets, simulated from the parameters given in Table 4.12.

4.4.4.1 Population under Expansion

Our first pseudo-observed data is from a population which is undergoing Exponential Growth. The growth rate at any time is proportional to the current population size. The posterior distributions obtained by the Monte Carlo particle method are given in Figure 4.19.

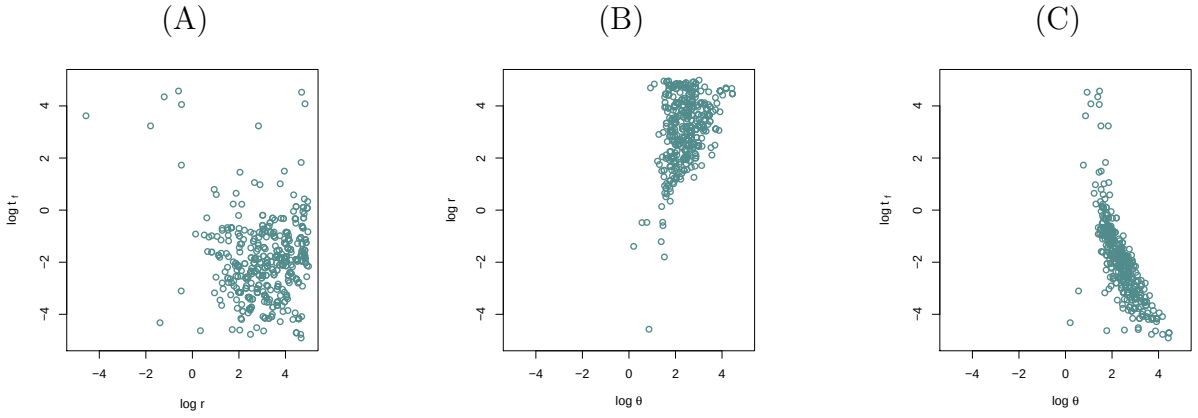


FIGURE 4.19: Joint Posterior distribution for the population under expansion, for $\log r$ and $\log t_f$ (A), and $\log \theta$ and $\log r$ (B), and $\log \theta$ and $\log t_f$ (C).

Figure 4.19 shows that, for a population under expansion, the natural logarithm of the growth rate r is positive, corresponding to a large growth rate. The number of generations since the population growth began are generally small, and the values of θ are relatively high, indicating a large mutation rate μ.

Figures 4.20, 4.21 and 4.22 show the ABC posterior distributions for the data from the population under expansion for the three ABC methods given in Section 4.4.3.

Of the three sets of ABC posterior distributions, the set that most resembles the posterior distributions given in Figure 4.19 is that produced by SMC-ABC. The Rejection ABC output also captures the same relationships in the posterior distributions. However, the distributions obtained by Auto-SS SMC-ABC bear little resemblance to those in Figure 4.19.

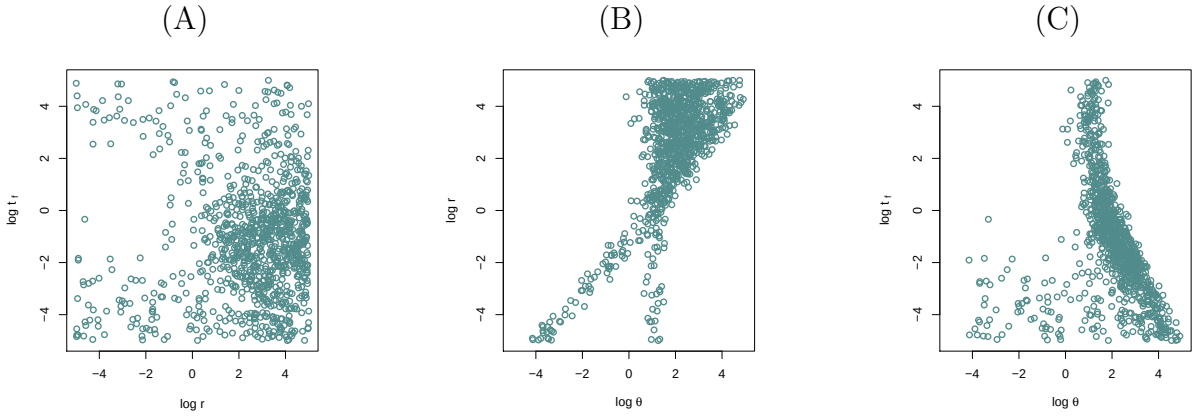


FIGURE 4.20: Joint ABC posterior distribution for the population under expansion, for $\log r$ and $\log t_f$ (A), $\log \theta$ and $\log r$ (B), and $\log \theta$ and $\log t_f$ (C), obtained using Rejection ABC.

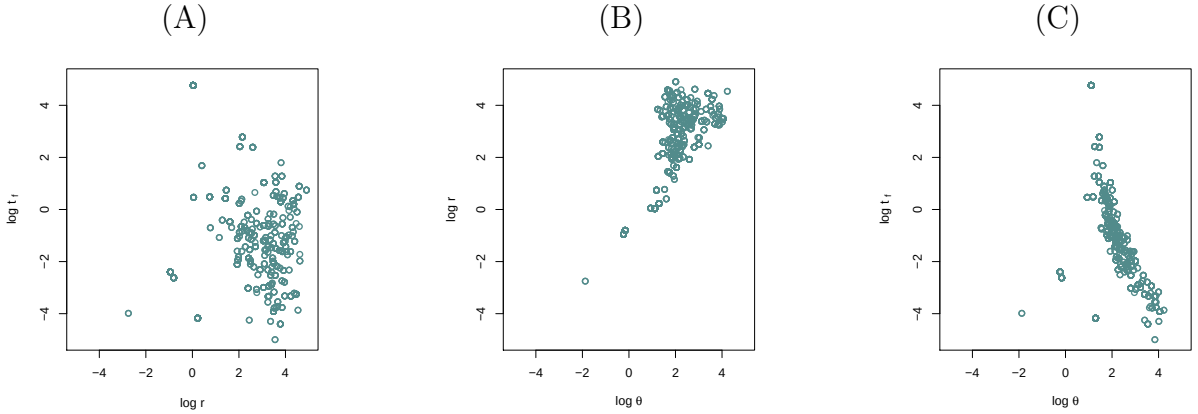


FIGURE 4.21: Joint ABC posterior distribution for the population under expansion, for $\log r$ and $\log t_f$ (A), $\log \theta$ and $\log r$ (B), and $\log \theta$ and $\log t_f$ (C), obtained using SMC-ABC.

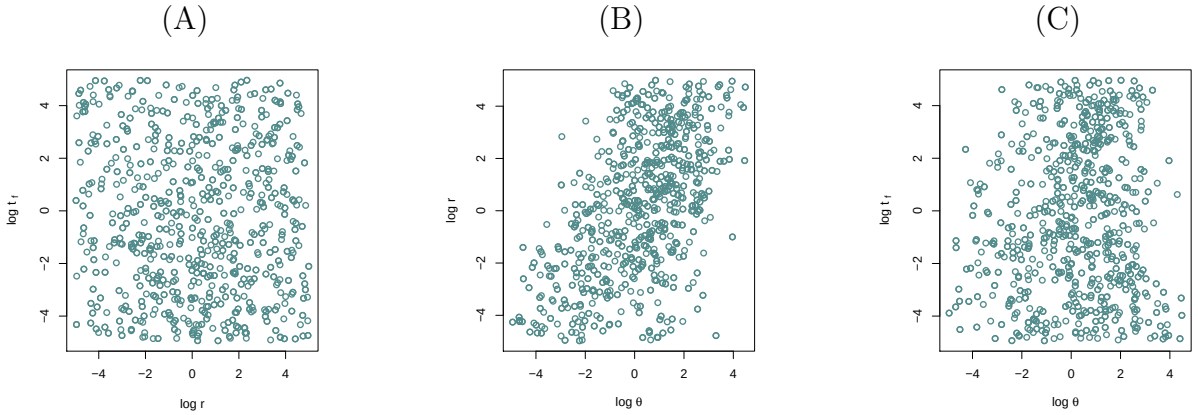


FIGURE 4.22: Joint ABC posterior distribution for the population under expansion, for $\log r$ and $\log t_f$ (A), $\log \theta$ and $\log r$ (B), and $\log \theta$ and $\log t_f$ (C), obtained using Auto-SS SMC-ABC.

4.4.4.2 Population under Contraction

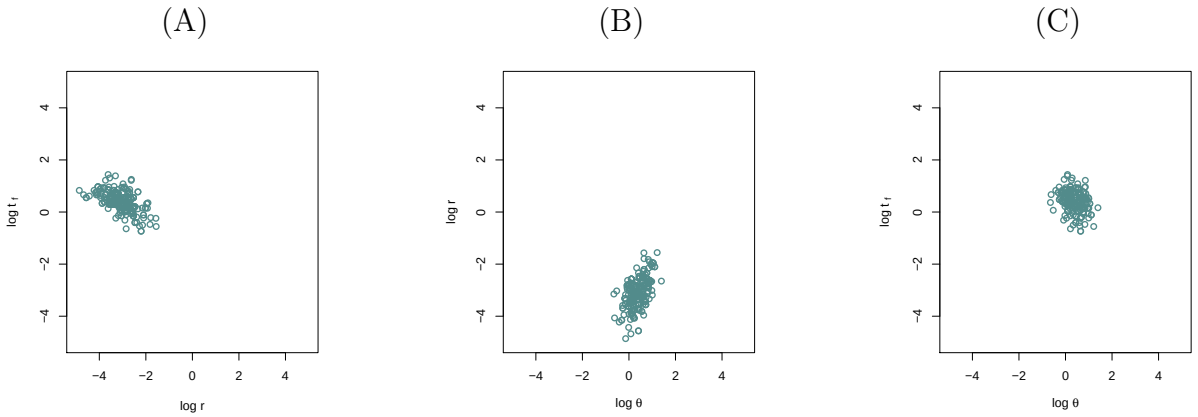


FIGURE 4.23: Joint Posterior distribution for the population under contraction, for $logr$ and $logt_f$ (A), and $log\theta$ and $logr$ (B), and $\log\theta$ and $\log t_f$ (C).

Figure 4.23 shows that, for a population under contraction, the marginal posterior distributions for $\log\theta$ and $\log t_f$ are centred around 0, and the marginal posterior for $\log r$ is centred around -3. All posteriors are very narrow, when compared to the prior distribution. Like for the data from the population under expansion, the ABC method which best reproduces the posteriors in Figure 4.23 is SMC-ABC. Again we see that the Auto-SS SMC-ABC posterior distributions bear little resemblance to those in Figure 4.23.

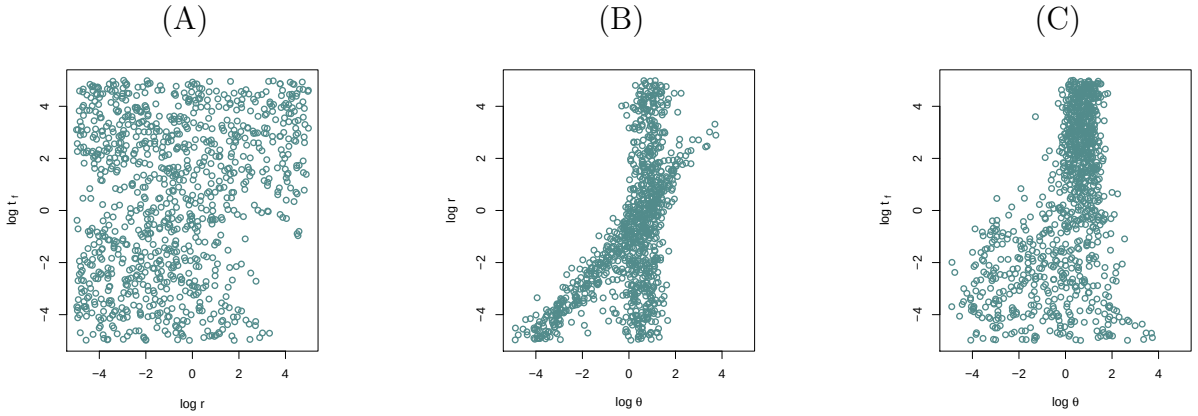


FIGURE 4.24: Joint Posterior distribution for the population under contraction, for $\log r$ and $\log t_f$ (A), $\log \theta$ and $\log r$ (B), and $\log \theta$ and $\log t_f$ (C), obtained using Rejection ABC.

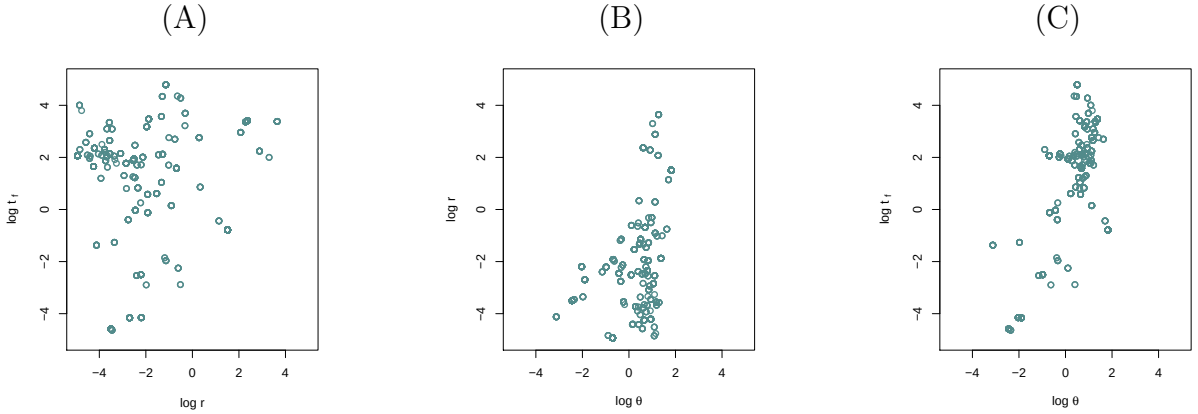


FIGURE 4.25: Joint Posterior distribution for the population under contraction, for $\log r$ and $\log t_f$ (A), $\log \theta$ and $\log r$ (B), and $\log \theta$ and $\log t_f$ (C), obtained using SMC-ABC.

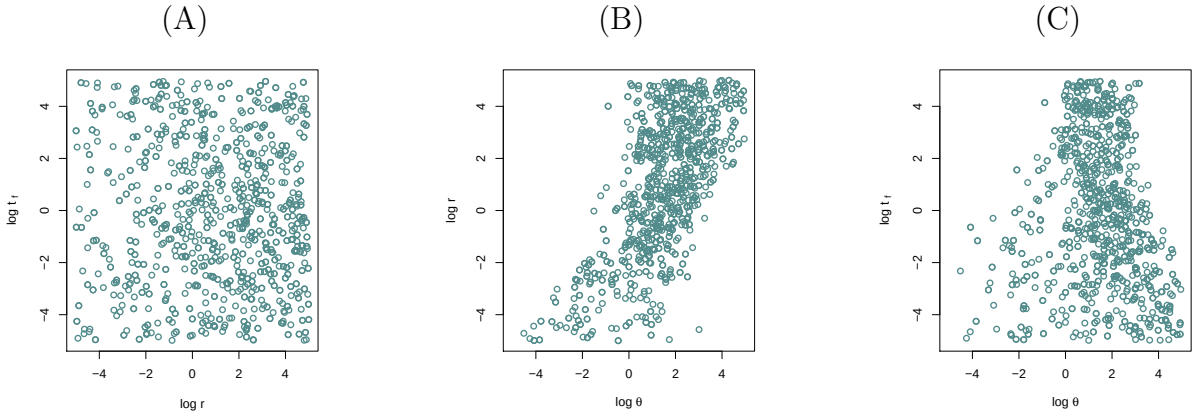


FIGURE 4.26: Joint Posterior distribution for the population under contraction, for $\log r$ and $\log t_f$ (A), $\log \theta$ and $\log r$ (B), and $\log \theta$ and $\log t_f$ (C), obtained using Auto-SS SMC-ABC.

4.4.4.3 Stable Population

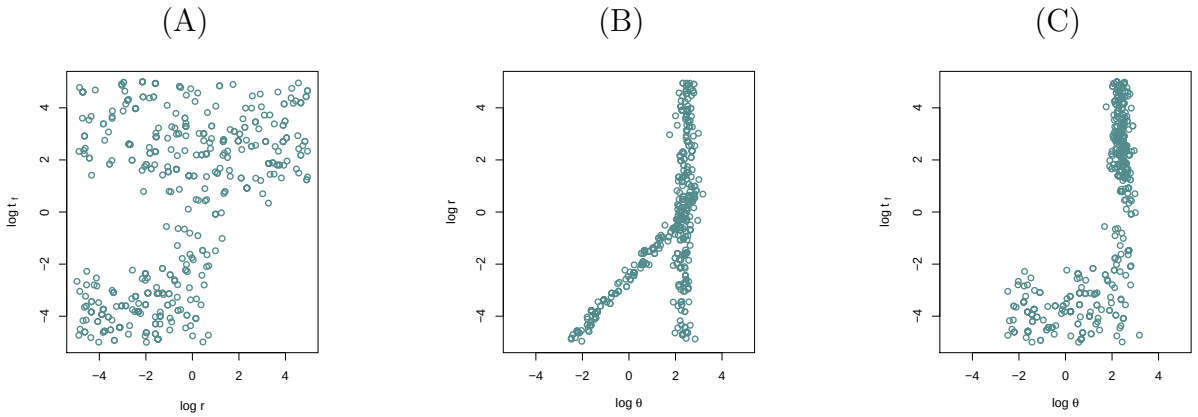


FIGURE 4.27: Joint Posterior distribution for the population of stable size, for $logr$ and $logt_f$ (A), and $log\theta$ and $logr$ (B).

The posterior distribution in Figure 4.27 shows that the posterior distributions for the parameters in a stable distribution are non-linear. For parameters t_f and r, the marginal posterior distributions are supported in the full range of the prior distribution. The regions of low density in frame (A) correspond to the unison of the high density regions in the posterior distributions for populations under expansion and contraction.

For this data set, the posterior distributions obtained by both SMC-ABC and rejection ABC capture the trends shown in Figure 4.27. The Auto-SS SMC-ABC algorithm gives posterior distributions which appear to be extremely noisy representations of the posterior distributions in Figure 4.27. However, the Auto-SS SMC-ABC results are again much worse than those produced by the other methods.

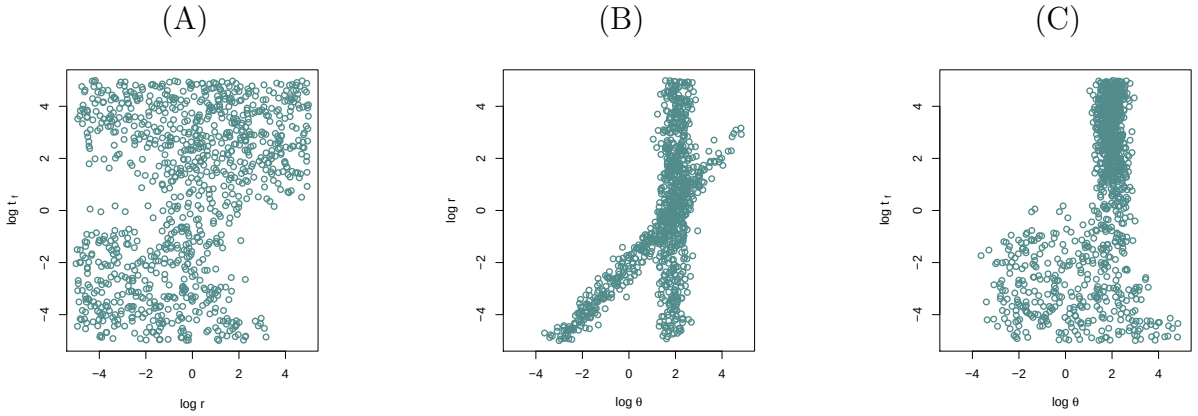


FIGURE 4.28: Joint Posterior distribution for the population of stable size, for $\log r$ and $\log t_f$ (A), $\log \theta$ and $\log r$ (B), and $\log \theta$ and $\log t_f$ (C), obtained using Rejection ABC.

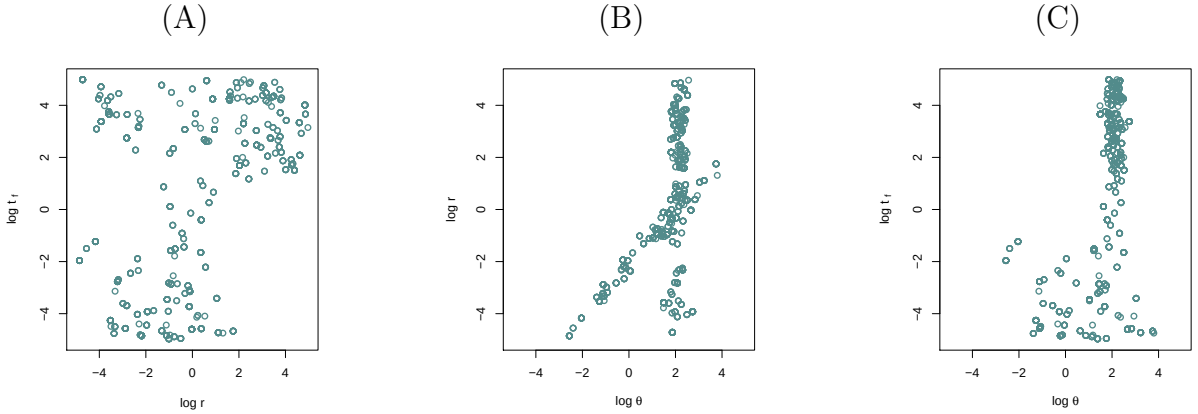


FIGURE 4.29: Joint Posterior distribution for the population of stable size, for $\log r$ and $\log t_f$ (A), $\log \theta$ and $\log r$ (B), and $\log \theta$ and $\log t_f$ (C), obtained using SMC-ABC.

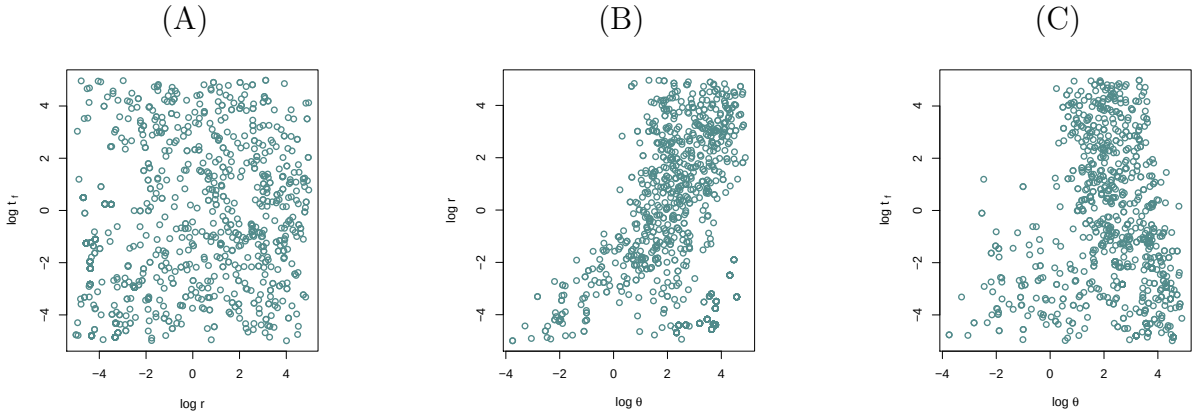


FIGURE 4.30: Joint Posterior distribution for the population of stable size, for $\log r$ and $\log t_f$ (A), $\log \theta$ and $\log r$ (B), and $\log \theta$ and $\log t_f$ (C), obtained using Auto-SS SMC-ABC.

For all of the data sets considered in this section, the worst posterior inference was produced by Auto-SS SMC-ABC. The Auto-SS SMC-ABC algorithm uses linear regression to select summary statistics at each iteration of the algorithm. However, for this example it is not the case that $\mathbb{E}(\theta|S)$ is linear. Thus the algorithm works poorly.

4.5 Discussion

In this chapter we saw a range of SMC-ABC methods applied to four example data sets. In the first three examples the Auto-SS SMC-ABC algorithm, presented in Algorithm 8, gave inference which was at least as good as that given by standard SMC-ABC, with notably better results being obtained by standard Auto-SS SMC-ABC on the first two examples in this chapter. However, such favourable results were not achieved when Algorithm 8 was applied to the population growth model in Section 4.4. This signifies that Auto-SS SMC-ABC is not a silver bullet and should be used with caution.

Algorithm 8 assumes that $\mathbb{E}(\theta|S)$ is linear for summary statistics S close to observed data $S(y)$. For the first two examples, namely the Bivariate Gaussian Model and the g-and-k distribution, this strong linearity between the parameters and the summary statistics existed. However, relationships between parameters and summary statistics were more spurious in the Earthworms model, leading to poor inference for some parameters. In the population growth model it is clear that this linearity does not exist, and as such the regression based methods perform poorly. The subject of population growth has been well studied, so it is known which summaries are informative for the parameters of simple models. Thus, in this case, this information should be used to select summaries to use within ABC methods which do not require adaptive summary statistic selection.

Chapter 5

Modelling the Likelihood Function

The ABC algorithms we have considered up to this chapter all have some inherent tuning parameter ϵ, on which acceptance or rejection decisions about parameters are made. In Rejection ABC, seen earlier in Algorithm 1, a parameter $\boldsymbol{\theta}$ is accepted if it gives rise to simulated data $\boldsymbol{x}$, such that

$$\rho(S(\boldsymbol{x}), S(\boldsymbol{y})) < \epsilon, \tag{5.1}$$

where $\boldsymbol{y}$ denotes the observed data, for which we wish to compute the posterior distribution, $S(\cdot)$ denotes the summary statistics, and $\rho(\cdot, \cdot)$ is a distance metric. Such ABC algorithms that include a tolerance ϵ are derivatives of the thought experiment of Rubin [2], which was equivalent to setting ϵ in Equation (5.1) to 0. It was in the paper of Pritchard et al. [3] that the first example of implementing such a rejection algorithm, using a non-zero tolerance, was seen.

Alongside the work of Rubin [2] and Pritchard et al. [3], the paper of Tavaré et al. [4] is also frequently listed as a pioneering ABC paper. However, Tavaré et al. [4] does not include a tolerance ϵ. Instead, parameters are accepted or rejected based on the value of the likelihood function. Specifically, for a parameter $\boldsymbol{\theta}$, observed data $\boldsymbol{y}$ and summary statistics $S(\cdot)$, parameter $\boldsymbol{\theta}$ is accepted with probability equal to

$$\frac{p(S(\boldsymbol{y})|\boldsymbol{\theta})}{\max_{\boldsymbol{\theta}} p(S(\boldsymbol{y})|\boldsymbol{\theta})}. \tag{5.2}$$

For the population genetics example considered in Tavaré et al. [4] the likelihood function was tractable, and thus the ratio in Equation (5.2) could be evaluated exactly. However when implementing ABC, it is assumed that the likelihood is intractable, or unknown, hence such a method could not be implemented directly.

The Synthetic Likelihood algorithm of Wood [50] uses an empirical estimate of the likelihood, $p(S(\boldsymbol{y})|\theta)$ at each iteration to make accept or reject decisions. The method of Wood [50] was shown to work well, though is fairly inefficient since, at each iteration, a large number of model simulations are required to construct the empirical likelihood function.

This motivated the work of Wilkinson [51], and Meeds and Welling [52], which both aim to model the likelihood function, but in a more efficient manner, using information from historical model simulations to guide the estimate, thereby reducing the total number of model simulations needed.

In this chapter, our novel contribution is to implement a Rejection ABC algorithm, which uses estimates of the joint density of parameters and summary statistics, $p(\boldsymbol{\theta}, S(\boldsymbol{x}))$, to make accept or reject decisions. We use a nearest neighbour algorithm to estimate the joint density efficiently, without the need for vast user tuning of the algorithm.

We then develop the algorithm further so that it works within a sequential Monte Carlo framework. This removes the need for computing the maximum likelihood, as is needed in the rejection algorithm. We found this step to be inefficient in practice, and thus the SMC algorithm with likelihood estimation gives more accurate results, for the same computational cost.

We begin this chapter with an overview of the algorithms of Wood [50], Wilkinson [51], and Meeds and Welling [52]. We then introduce our Iterative Synthetic Likelihood Estimation algorithm (ISLE), before discussing the k-Nearest Neighbour density estimation method which we use to estimate the joint density throughout the algorithm. Finally we present the Sequential Monte Carlo adaptation of our iterative Likelihood estimation algorithm (SMC-ISLE), and show that it performs well for a number of examples.

5.1 The Synthetic Likelihood Method

The Synthetic Likelihood method of Wood [50] was developed with the focus of improving inference for '*noisy non linear ecological dynamical systems*'. Such models are highly stochastic and so likelihood estimates based on only one simulation from the model give inaccurate results.

The Synthetic Likelihood method hinges on modelling the unknown likelihood $f(S(\boldsymbol{x})|\boldsymbol{\theta})$ as a multivariate Gaussian for a given parameter vector $\boldsymbol{\theta}$. Unbiased estimates of the mean, $\mu_{\boldsymbol{\theta}}$, and covariance matrix, $\Sigma_{\boldsymbol{\theta}}$ for the Gaussian are computed through standard moment estimations, based on data which is sampled from the model.

Formally, for a parameter $\boldsymbol{\theta}$, the likelihood function is modelled as

$$f(\cdot|\boldsymbol{\theta}) \sim N(\mu_{\boldsymbol{\theta}}, \Sigma_{\boldsymbol{\theta}}). \tag{5.3}$$

The unknown mean vector and covariance matrix in Equation (5.3) are computed by first simulating M data sets, $x_1, \ldots, x_M$ at parameter value $\boldsymbol{\theta}$. From these summaries the unknown parameters $\mu_{\boldsymbol{\theta}}$ and $\Sigma_{\boldsymbol{\theta}}$ are unbiasedly estimated as

$$\widehat{\mu}_{\boldsymbol{\theta}} = \bar{S} = \frac{1}{M}\sum_{i=1}^{M} S(\boldsymbol{x}_j) \qquad \text{and} \tag{5.4}$$

$$\widehat{\Sigma}_{\boldsymbol{\theta}} = \frac{1}{M-1}\sum_{j=1}^{M}(S(\boldsymbol{x}_j) - \bar{S})(S(\boldsymbol{x}_j) - \bar{S})^T. \tag{5.5}$$

Thus for any given data $\boldsymbol{x}$, with summary $S(\boldsymbol{x})$, we are able to compute an estimate of the likelihood $f(S(\boldsymbol{x})|\boldsymbol{\theta})$ as

$$\widehat{f}(S(\boldsymbol{x})|\boldsymbol{\theta}) \propto |\widehat{\Sigma}_{\boldsymbol{\theta}}|^{-1/2} \exp\left(-\frac{1}{2}(S(\boldsymbol{x}) - \hat{\mu}_{\boldsymbol{\theta}})^T \widehat{\Sigma}_{\boldsymbol{\theta}}(S(\boldsymbol{x}) - \hat{\mu}_{\boldsymbol{\theta}})\right), \tag{5.6}$$

where the proportionality constant depends only on the dimension of $S(\boldsymbol{x})$.

In practice, we work with an estimate of the log likelihood function, denoted by

$$\widehat{l}_{S(\boldsymbol{x})}(\boldsymbol{\theta}) := \log \widehat{f}(S(\boldsymbol{x})|\boldsymbol{\theta}), \tag{5.7}$$

rather than the likelihood function itself, thereby removing the need to compute the exponential in Equation (5.6) and avoiding arithmetic underflow when dealing with points in the tails of the distribution. The full Synthetic Likelihood method is presented in Algorithm 10.

Algorithm 10 Synthetic Likelihood, Wood [50]

Let $q(\cdot|\theta)$ be a symmetric proposal distribution for θ. Fix $M \in \mathbb{N}^+$, set $t = 1$ and initialise θ_1. Let $\rho(\cdot, \cdot)$ be a distance metric on the space of summary statistics. Set $t = 0$ and select $\boldsymbol{\theta}^{(0)}$.

1: Simulate M data sets

$$\boldsymbol{x}_j^{(t)} \sim f(\cdot|\boldsymbol{\theta}^{(t)}) \text{ for } j \in 1, \ldots, M \tag{5.8}$$

and compute summaries $S(\boldsymbol{x}_j)$

2: Compute $\widehat{\boldsymbol{\theta}}^{(t)}$ and $\widehat{\Sigma}^{(t)}$, using Equations (5.4) and (5.5).

3: Propose $\boldsymbol{\theta}' \sim q(\cdot|\boldsymbol{\theta}^{(t)})$.

4: Simulate M data sets at $\boldsymbol{\theta}'$:

$$\boldsymbol{x}'_j \sim f(\cdot|\boldsymbol{\theta}'), \text{ for } j \in 1, \ldots, M \tag{5.9}$$

and compute summaries $S(\boldsymbol{x}'_j)$.

5: Compute $\widehat{\boldsymbol{\theta}}'$ and $\widehat{\Sigma}'$, using summaries $S(\boldsymbol{x}'_j)$ with Equations (5.4) and (5.5).

6: With probability

$$\min\left(1, \exp\left(\widehat{l}_{S(\boldsymbol{y})}(\boldsymbol{\theta}') - \widehat{l}_{S(\boldsymbol{y})}(\boldsymbol{\theta}^{(t)})\right)\right). \tag{5.10}$$

set $\theta^{(t+1)} = \theta'$, else set $\theta^{(t+1)} = \theta^{(t)}$.

7: Set $t = t + 1$

Repeat steps 3 to 7 until convergence.

As it is presented in Wood [50], the Synthetic Likelihood algorithm is classical, rather than Bayesian, as an improper uniform prior is assumed for the parameter $\boldsymbol{\theta}$. In practice, implementing the Synthetic Likelihood method in a Bayesian way involves only a minor amendment to Algorithm 10: A ratio of the prior densities should be included in the Metropolis-Hastings ratio. Explicitly, Equation (5.10) should be replaced by

$$\min\left(1, \exp\left(\widehat{l}_{S(\boldsymbol{y})}(\theta') - \widehat{l}_{S(\boldsymbol{y})}(\theta^{(t)}) + \log(\pi(\theta') - \log(\pi(\theta^{(t)})\right)\right). \tag{5.11}$$

Furthermore, Algorithm 10 assumes a symmetric perturbation kernel is used, however an asymmetric kernel can be used, provided the ratio in Equation (5.11) is amended appropriately. The Bayesian Synthetic Likelihood (BSL) algorithm which uses Equation(5.11) has been implemented in many papers since the original algorithm was suggested in Wood [50] ([53], [54], [55]).

The Synthetic Likelihood method makes assumptions on the distribution of the underlying likelihood function. It is certainly not the case that, for any given summary statistics, the likelihood will be Gaussian. However, Wood [50] notes that, in the limit as $M \to \infty$, where M is the number of simulated data sets at a particular parameter value, we should expect the Gaussian approximation of the likelihood to tend to the true likelihood due to the central limit theorem. As well as this, the paper gives suggestions for summary statistics which should to lead to likelihoods which follow a normal distribution, even for a small M. Price et al. [56] show that, for a range of examples, the value of M selected has little impact on the accuracy of the resultant posterior distributions produced by the algorithms, though a larger value of M leads to a higher acceptance rate for the algorithm.

5.2 Gaussian Processes in ABC

The Synthetic Likelihood algorithm sparked an interest in other methods of modelling the likelihood function. Both Meeds and Welling [52] and Wilkinson [51] use Gaussian processes to model the likelihood of summary statistics marginally. The main advantage to using Gaussian processes, compared to the methodology of the Synthetic Likelihood algorithm, is that the Gaussian process is able to use information about the estimated likelihood function at parameter value $\boldsymbol{\theta}$ to guide the estimate of the likelihood function at parameter value $\boldsymbol{\theta} + \boldsymbol{h}$, for small $\boldsymbol{h}$.

Meeds and Welling [52] use Gaussian processes to model the likelihood of each summary statistic marginally. The paper gives guidance on when it is necessary to simulate data from the model at a newly proposed $\boldsymbol{\theta}'$ versus when there are enough samples drawn near by to ensure that an accurate estimate of the likelihood can be made without the need for more simulations. The method considers the expected error of making an incorrect reject or accept decision. If this error exceeds some threshold more model simulations are run.

The algorithm of Wilkinson [51] models the joint log-likelihood for all summary statistics by a Gaussian process. An initial training ensemble of parameter values are selected using a Sobol sequence, which ensures that the training points for the Gaussian Process are evenly distributed across the support of the prior distribution. The Gaussian processes are used to determine *implausible regions* of parameter space, which have negligible posterior mass. Such implausible

regions are identified, new training points are sampled from plausible regions, and a new Gaussian process is fitted to the samples. Once an unspecified number of iterations of the fitting and sampling process has occurred, and the Gaussian process model is deemed to accurately model the log likelihood function, MCMC can be run, without the need for any further sampling from the model.

In practice, the method of Wilkinson [51] was shown to work well on a population growth example, and on the Ricker model. However, a large amount of user guidance was needed to implement the algorithm in both these cases, specifically in selecting the number of iterations of the fitting and sampling process and in picking a functional form for the mean and the covariance for the Gaussian Process.

5.3 Iterative Likelihood Estimation

The Synthetic Likelihood method [50], presented in Section 5.1 has been shown to deliver good inference on a range of applications, but relies on the assumption that the underlying likelihood function is Gaussian.

The methods of both Wilkinson [51] and Meeds and Welling [52] require non trivial amounts of user input to select tuning parameters for the Gaussian process, with the method of Wilkinson [51] requiring extensive user guided tuning in the initial stages of the algorithm. With IBMs in mind, we wish to develop a method that, in a similar vein to those presented earlier in this chapter, successfully models the likelihood function, to obtain improved inference for the posterior distribution, but requires minimal user tuning, and does not make strong assumptions about the underlying distribution of the likelihood function. In a similar vein, Papamakarios and Murray [57] and Bonassi et al. [58] present ABC methods which are based on estimating conditional densities.

We begin by presenting an algorithm which follows on from the Synthetic Likelihood algorithm of Wood [50], in the sense that, at every proposed parameter value, $\boldsymbol{\theta}_j^{(t)}$, an accept or reject decision is made based on an estimate of the likelihood of the observed summary statistics $S(\boldsymbol{y})$, at that parameter value. The likelihood at the observed data is compared to the maximum

likelihood for that parameter value, given by

$$\max_{\boldsymbol{x}} \frac{\widehat{p}_t(\boldsymbol{\theta}_j^{(t)}, S(\boldsymbol{x}))}{\pi(\boldsymbol{\theta}_j^{(t)})}, \tag{5.12}$$

where $\widehat{p}_t(\cdot, \cdot)$ denotes the estimate of the joint density at iteration t. The overall structure of the algorithm follows that of a standard Importance Sampling algorithm, an example of which we saw in Algorithm 3. At each iteration, parameters are given an importance sampling weight which accounts for the difference between the sampling distribution used at that iteration, and the prior distribution. Parameters are then resampled with probability proportional this weight.

Unlike the Algorithms of Wood [50], Wilkinson [51] and Meeds and Welling [52], we do not estimate the likelihood function directly, but instead estimate the joint density of parameters and summary statistics. Likelihood estimates can then be obtained by dividing the estimate of the joint density by the prior density. Specifically, let $\widehat{p}(\boldsymbol{\theta}, S(\boldsymbol{x}))$ be an estimate of the joint density. Then we obtain an estimate of the likelihood, which we denote $\widehat{f}(S(\boldsymbol{x})|\boldsymbol{\theta})$, by

$$\widehat{f}(S(\boldsymbol{x})|\boldsymbol{\theta}) = \frac{\widehat{p}(\boldsymbol{\theta}, S(\boldsymbol{x}))}{\pi(\boldsymbol{\theta})}. \tag{5.13}$$

In Section 5.3.1 we develop a density estimation method, which we use to compute $\widehat{p}_t(\boldsymbol{\theta}, S(\boldsymbol{y}))$, where the subscript t denotes the iteration at which the estimate is made. For now we assume we have access to this density estimate for any pair $(\boldsymbol{\theta}, S(\boldsymbol{x}))$. With this density estimate in hand, we present the Importance Sampling Likelihood Estimation algorithm (ISLE) in Algorithm 11 and we describe the main processes below.

Algorithm 11 Importance Sampling with Likelihood Estimation (ISLE)

Fix N and N_{init}. Let $\widehat{p}_t(\cdot,\cdot)$ denote the estimate of the joint distribution of parameters and summary statistics, computed using historical samples $\{\boldsymbol{\theta}^{(\tau)}, S(\boldsymbol{x}^{(\tau)})\}$, for all $\tau < t$.

1: **Obtain Initial Sample for Joint Density Estimation** At iteration $t = 0$,

For $j = 1, \ldots, N_{\text{init}}$:

a. Sample $\boldsymbol{\theta}_j^{(t)} \sim \pi(\cdot)$.

b. Simulate $\boldsymbol{x}_j^{(t)} \sim f(\cdot|\boldsymbol{\theta}_j^{(t)})$ and compute $S(\boldsymbol{x}_j^{(t)})$.

2: **Simulating First Estimate of Posterior** At iteration $t = 1$,

For $j = 1, \ldots, N$:

a. Sample $\boldsymbol{\theta}_j^{(t)} \sim \pi(\cdot)$.

b. Simulate $\boldsymbol{x}_j^{(t)} \sim f(\cdot|\boldsymbol{\theta}_j^{(t)})$ and compute $S(\boldsymbol{x}_j^{(t)})$.

c. Compute

$$\alpha_j = \max_{\boldsymbol{x}} \frac{\widehat{p}_t(\boldsymbol{\theta}_j^{(t)}, S(\boldsymbol{x}))}{\pi(\boldsymbol{\theta}_j^{(t)})}. \tag{5.14}$$

Sample $u \sim \mathcal{U}(0, \alpha_j)$.

If

$$u \leq \frac{\widehat{p}_t(\boldsymbol{\theta}_j^{(t)}, S(\boldsymbol{y}))}{\pi(\boldsymbol{\theta}_j^{(t)})}, \tag{5.15}$$

set $w_j^{(t)} \propto 1$.

Else,

set $w_j^{(t)} = 0$.

3: **Iterating** At iteration $t > 1$, set σ_t^2 equal to twice the weighted empirical variance of the set of $\boldsymbol{\theta}_i^{(t-1)}$s.

For $j = 1, \ldots, N$:

a. Sample $\boldsymbol{\theta}_j^{(\prime)}$ from the $\boldsymbol{\theta}_i^{(t-1)}$s, with probability $w_i^{(t-1)}$.

b. Simulate $\boldsymbol{\theta}_j^{(t)} \sim \mathcal{N}(\boldsymbol{\theta}_j^{(\prime)}, \sigma = \sigma_t^2)$

c. Simulate $\boldsymbol{x}_j^{(t)} \sim f(\cdot|\boldsymbol{\theta}_j^{(t)})$ and compute $S(\boldsymbol{x}_j^{(t)})$.

d. Compute

$$\alpha_j = \max_{\boldsymbol{x}} \frac{\widehat{p}_t(\boldsymbol{\theta}_j^{(t)}, S(\boldsymbol{x}))}{\pi(\boldsymbol{\theta}_j^{(t)})}. \tag{5.16}$$

e. Sample $u \sim \mathcal{U}(0, \alpha_j)$.

f. If

$$u \leq \frac{\widehat{p}_t(\boldsymbol{\theta}_j, S(\boldsymbol{y}))}{\pi(\boldsymbol{\theta}_j)}, \tag{5.17}$$

accept $\boldsymbol{\theta}_j^{(t)}$ and set

$$w_j^{(t)} \propto \frac{\pi(\boldsymbol{\theta}_j^{(t)})}{\sum_{i=1}^n w_i^{(t-1)} \phi\left(\sigma_t^{-1}(\boldsymbol{\theta}_j^{(t)} - \boldsymbol{\theta}_i^{(t-1))}\right)}, \tag{5.18}$$

where $\phi(\cdot)$ is the standard normal density.

Else,

set $w_j^{(t)} = 0$ and reject $\theta_j^{(t)}$.

Iterate step 3 until estimates of $\widehat{p}_t(\cdot, \cdot)$ converge.

We now describe the main steps of the algorithm.

- In step one an initial set of parameters and summary statistics are sampled from the joint distribution. These samples are used to estimate the joint density, $p(\boldsymbol{\theta}, S(\boldsymbol{x}))$, at iteration $t = 1$, and at all subsequent iterations of the algorithm.

- In step two of Algorithm 11, parameters are sampled from the prior distribution, and are accepted or rejected based on the estimate of the likelihood at the observed summary statistics, $S(\boldsymbol{y})$. The likelihood is estimated using the samples of $\boldsymbol{\theta}$ and $S(\boldsymbol{x})$ simulated in step one of the algorithm.

- Finally, in step three of the algorithm, parameters are drawn from the proposal distribution which is here selected to be the Gaussian Kernel Density estimation of the set of accepted parameters $\boldsymbol{\theta}_i^{(t-1)}$ from the previous iteration. This proposal distribution can be written explicitly as

$$q_t(\boldsymbol{\theta}) \propto \sum_{i=1}^{N} w_i^{(t-1)} \phi \left(\frac{\boldsymbol{\theta} - \boldsymbol{\theta}_i^{(t-1)}}{\sigma_t} \right), \tag{5.19}$$

  where $\phi(\cdot)$ is the standard normal density, and σ_t is a scalar if parameters $\boldsymbol{\theta_i}$ are one dimensional. For higher dimensional $\boldsymbol{\theta_i}$, σ_t^2 is a covariance matrix. This proposal distribution is also used in the PMC-ABC algorithm of Beaumont et al. [15], who note that the Gaussian Kernel can be replaced by any general Kernel.

- In Step 3d, the likelihood is estimated using all samples $(\boldsymbol{\theta}, S(\boldsymbol{x}))$ which have been simulated at previous iterations of the algorithm. Again, parameters are accepted or rejected based on the estimate of the likelihood, evaluated at the observed summary statistics. Accepted particles are then weighted with the standard Importance Sampling weight, given in Equation (5.18).

Step 3 of Algorithm 11 is then repeated.

At the end of step 2 the set of accepted parameters, $\boldsymbol{\theta}_i^{(1)}$, are drawn from the approximate posterior distribution. In theory we could terminate the algorithm here. However, when the joint space of summary statistics and parameters is high dimensional, it is unlikely that the initial density estimate is accurate, particularly for N_{init} relatively small.We wish to build up

a better estimation of the joint density $p(\boldsymbol{\theta}, S(\boldsymbol{x}))$ in the region of parameter space close to the posterior distribution, thus we iterate the algorithm, obtaining more samples from the joint distribution, and therefore obtain a more accurate density estimate and hence a more accurate posterior distribution.

5.3.0.1 Weighted Samples from the Joint Distribution

In steps 1 and 2 of Algorithm 11, parameters are sampled from the prior distribution $\pi(\cdot)$, and summary statistics are computed for data which has been simulated from the model. Thus for all j and $\tau \in (0, 1)$, we have that

$$(\theta_j^{(\tau)}, x_j^{(\tau)}) \sim \pi(\theta) f(x|\theta). \tag{5.20}$$

However, at iteration $t > 1$, parameters are simulated from a distribution $q_t(\cdot)$, which given in Equation (5.19). Because of this, any parameters and summary statistics sampled after iteration $t = 1$ are no longer drawn from the joint distribution, as was the case for $t = 0$ and $t = 1$. This means we must weight such particles in order to be able to use them to estimate the joint density. Let $W_j^{(t)}$ be the weight such that, when particles $(\boldsymbol{\theta}_j^{(t)}, S(\boldsymbol{x}_j^{(t)}))$ are resampled proportional to weight $W_j^{(t)}$ we obtain a sample from the joint distribution of parameters and summary statistics. Any samples $(\boldsymbol{\theta}_j^{(t)}, S(\boldsymbol{x}_j^{(t)}))$ drawn at time $t = 0$ and $t = 1$ are drawn from the true joint distribution, hence for all j, $W_j^{(0)} = 1$ and $W_j^{(1)} = 1$. For $t > 1$, particles are sampled from the proposal distribution $q_t(\cdot)$, rather than the prior distribution $\pi(\cdot)$. Hence we give samples the weight

$$W_j^{(t)} = \frac{\pi(\boldsymbol{\theta}_j^{(t)})}{q_t(\boldsymbol{\theta}_j^{(t)})}, \tag{5.21}$$

which coincides with the importance sampling weight given in Equation (5.18).

5.3.1 Density Estimation

Algorithm 11 assumes we have access to an estimate of the joint density $p(\boldsymbol{\theta}, S(\boldsymbol{x}))$, with the estimate at time t being denoted $\widehat{p}_t(\boldsymbol{\theta}, S(\boldsymbol{x}))$. When estimating the joint density $p(\boldsymbol{\theta}, S(\boldsymbol{x}))$, we wish to use a density estimation method which exhibits the following properties:

1. Accurate in High Dimensions: it is common for both $\boldsymbol{\theta}$ and $S(\boldsymbol{x})$ to be high dimensional. Hence we wish to use a density estimation method which is effective in high dimensions.

2. Computationally Cheap: estimates of $p(\boldsymbol{\theta}, S(\boldsymbol{x}))$ are to be computed at many points throughout the domain, so the cost must be cheap to keep running time low.

3. Require little tuning: previously discussed methods require much tuning, which is undesirable in practice.

4. Flexible Modelling: the method should be able to estimate the density for all underlying distributions of $p(\boldsymbol{\theta}, S(\boldsymbol{x}))$, and should make no strong assumptions about the distribution.

5. Able to incorporate weighted samples: the algorithm should run within the Importance Sampling framework. This means that the density estimation method must be able to estimate densities based on samples of points $(\boldsymbol{\theta}, S(\boldsymbol{x}))$, where W is the weight given to the particle $(\boldsymbol{\theta}, S(\boldsymbol{x}))$ such that, under resampling with weight W, the particles are distributed from the true joint distribution.

With these properties in mind, we proceed with k-Nearest Neighbour (kNN) Density estimation [59],[60]. As k Nearest Neighbour density estimation is non-parametric, it does not require the user to select a specific form for the density function, unlike when fitting Gaussian processes. It appeared to be the most flexible of the density estimation methods we considered (Gaussian Processes, kernel density estimation, etc.), and has only one tuning parameter, k. Furthermore, the search for the set of k Nearest Neighbours can be implemented non exhaustively, using a method known as k-d tree data structure [61]. When using a k-d tree on a set of n points, the time taken to find the k nearest neighbours to any point is bounded above by $O(kn\log(n))$, and bounded below by $O(n\log(n))$.

We could find no examples of k-Nearest Neighbour Density Estimation being used to compute densities based on weighted samples in the literature. Consequently, in Section 5.4.1 we adapt the standard k-NN density estimation such that weighted samples could be given as input to the density estimation method.

5.4 k-Nearest Neighbour Density Estimation

The k-Nearest Neighbour (kNN) density estimate (Loftsgaarden et al. [59]) is defined as follows: Let $y_1, \ldots, y_n \in \mathbb{R}^d \sim_{iid} f(\cdot)$ be a sample of n training points from distribution $f(\cdot)$. Let c_d be the measure of the unit sphere in $\mathbb{R}^d$, let $\rho(\cdot, \cdot)$ be a distance function on $\mathbb{R}^d$. Then with the distance from x to its kth nearest neighbour in the training set being given by

$$r_k(x) = inf\Big\{r : \sum_{i=1}^{n} \mathbb{I}\{\rho(y_i, x) \leq r\} \geq k\Big\}, \tag{5.22}$$

the kNN density estimate of $f(\cdot)$ at x is given by

$$\widehat{f}(x) = \frac{k}{n} \frac{1}{c_d \, r_k(x)^d}. \tag{5.23}$$

In Section 5.4.3 we consider an alternative statement of the kNN density estimate, given by Parzen [60] and show that incorporating the two methods leads to better density estimation in the tails of the distribution.

5.4.1 Incorporating Weighted Samples

Equation (5.23) enables us to compute a density estimate at any point in $\mathbb{R}^d$, however we can see that the estimate is not a true density, as the integral of Equation (5.23) is unbounded over the domain of x. Furthermore, in its standard form, the kNN density estimate is not able to compute a density estimate based on a weighted sample from the distribution of interest. Suppose now our sample points $y_1, \ldots, y_n$ are a weighted sample from $f(\cdot)$ with corresponding weights $W_1, \ldots, W_n$. Except in the trivial case where $W_i = 1$ for all i, we are unable to use Equation (5.23) without first resampling the points $y_1, \ldots, y_n$ according to their weights. This makes the method, as it stands, unusable within Algorithm (11), for time $t > 2$, training points $\theta, S(x)$ have a weight associated with them, given by Equation (5.21).

We amend the Loftsgaarden et al. [59] k-Nearest Neighbour density estimate so that we can compute the estimate from a set of weighted points (y_i, W_i), from the joint distribution. Instead

of specifying k, the number of neighbours we wish to consider, we give a value ω, which is the desired total amount of weight we wish to include in the density estimate.

Let $\{y_i, W_i\}_{i=1}^n$ be a weighted sample from the distribution $f(\cdot)$. Let $r_\omega(x)$ be given by

$$r_\omega(x) = inf\Big\{d : \sum_{i=1}^{n} W_i \, \mathbb{I}\{\rho(y_i, x) \leq d\} \geq \omega\Big\}. \tag{5.24}$$

Then the weighted k nearest neighbour density estimate of $f(x)$ is given by

$$\hat{f}_\omega(x) = \frac{\sum_{i=1}^n W_i^*}{\sum_{i=1}^n W_i} \frac{1}{c_d \, r_\omega(x)^d}, \tag{5.25}$$

where c_d is the area of the unit sphere in $\mathbb{R}$, and

$$W_i^* = \begin{cases} W_i & \text{if } \rho(y_i, x) \leq r_\omega(x), \\ 0 & \text{otherwise.} \end{cases} \tag{5.26}$$

The quantity $r_\omega(x)$ is defined as the smallest distance such that the d dimensional sphere of radius $r_\omega(x)$ contains weight no less than ω. Note that in the case where all particles have weight $W_i = 1$, setting $\omega = k$, recovers the standard kNN density estimate, as given in Equation (5.23).

5.4.2 Selecting Tuning Parameters

Ideally, the ISLE algorithm (Algorithm 11) should be implementable with minimal user tuning required. The main tuning parameter of the algorithm as it stands is ω as seen in Equation (5.24). To advise our choice of ω we return to the standard k Nearest Neighbour density estimation, and consider guidance in the literature on how to choose parameter k. The choice of k affects the level of smoothness we see in the resultant density estimate. A value of k that is too small leads to locally noisy density estimates, whereas a value that is too large results in an over-smoothed curve.

Mack and Rosenblatt [62] show that the value of k which minimises the mean squared error of a multivariate density estimate in d dimensions, with respect to a sample from the joint

distribution of the data, based on a set of n sample points is given by

$$k \approx n^{4/4+d}, \tag{5.27}$$

In practice, such a choice of k gives good estimates in central regions of the density. However, the tail behaviour of density estimates is commonly very different to the behaviour elsewhere (as the bias is greater than variance in these regions Silverman [63]). Because of this, the ideal value of k for estimating the density in the center of the distribution differs from that in the tails. Thus we wish to develop an automatic method of selecting k, (and hence ω,) which leads to good density estimates across the domain of $f(\cdot)$.

Example 5.1. *To illustrate the impact of k on the density estimation we consider the following example.*

Let $x_1, \ldots, x_{1,000}$ be a sample from the standard normal distribution in one dimension. Figure 5.1 shows the true density (black line) and the kNN density estimate for $k = 10, 50, 100, 200, 251, 300$ in green, with $k = 251$ being the nearest whole integer to $1000^{4/5}$, thus is the optimal choice of k. The density is estimated at points $-4, -3.99, -3.98, \ldots, 3.98, 3.99, 4$ using Equation (5.23).

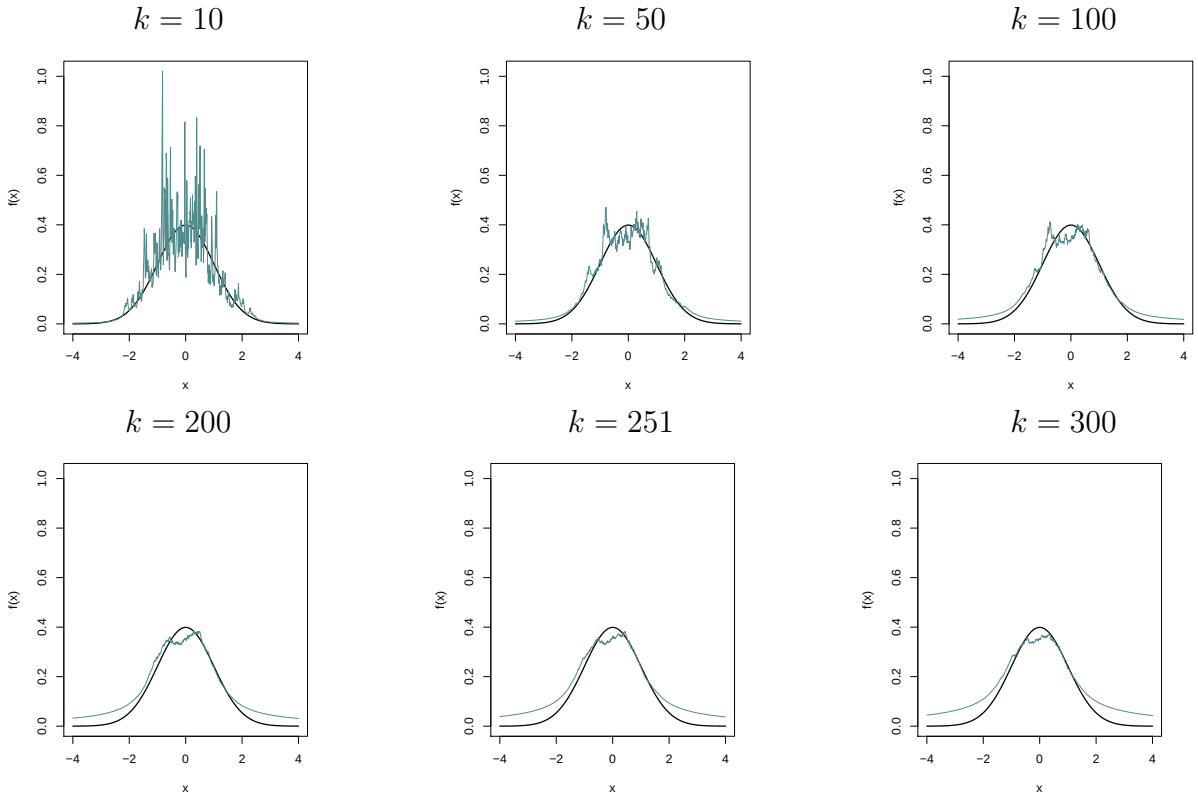


FIGURE 5.1: True density (black) and kNN density (blue), for a range fo values of k.

Figure 5.1 shows that, as k increases the density estimate becomes smoother and appears to give a better approximation of the density in the region $x \in (-2, 2)$. However, the figure also shows that the density estimate in the tails of the distribution worsens as k is increased. This is further illustrated in Table 5.1, which gives the error between the estimated and true density over the range of values of k, for $x = 0$ and $x = 3$. For $x = 3$, which sits in the tails of the density, we see that the percentage error in the density estimate increases greatly as k is increased. This is contrary to the point $x = 0$, where we see a general improvement in the percentage error, as k is increased to 251. The estimate then worsens for $k = 300$.

x	$f(x)$	$\hat{f}(x)$					
		k=10	k=50	k=100	k=200	k=251	k=300
0	0.3989	0.2229 (44.14 %)	0.3282 (17.72 %)	0.3359 (15.80 %)	0.3511 (12.00 %)	0.3624 (9.16 %)	0.3519 (11.79 %)
3	0.0044	0.0074 (67.08 %)	0.0120 (350 %)	0.0289 (550 %)	0.0454 (924 %)	0.0536 (1109 %)	0.0602 (1258 %)

TABLE 5.1: k nearest neighbour density estimate at $x = 0$ and $x = 3$, for a range of values of k. Training data is sampled from the standard normal distribution. Percentage error of the density estimates is given in parenthesis.

5.4.2.1 Improving density estimates in the tails

As we saw in Example (5.1), the kNN method leads to overestimation of density in the tails of the distribution. In essence, you can think of the kNN algorithm as "borrowing density sideways" in the tails of the joint density, as it computes estimates based on points which are significantly closer to the mean of the domain than the point we are interested in itself. One possible way to combat this would be to select a smaller value of k (or ω,) as the tuning parameter in the tails, since this would lead to less 'borrowing of density' from higher density regions. However in practice this is difficult to implement, since it is not straightforward to determine whether a point x is in the tails of the density $f(\cdot)$. A very similar density estimation method is suggested in Parzen [60]. The method differs from that of Loftsgaarden et al. [59] in that, instead of fixing k, and thus the radius $r_k(x)$ being defined as in Equation (5.22), it is the radius which is fixed. Thus the k Nearest Neighbour density estimate of [60], for a fixed radius r is given by

$$\widehat{f}(x) = \frac{1}{n} \frac{\sum_{i=1}^{n} \mathbb{I}\{\rho(y_i, x) \leq r\}}{c_d r^d}. \tag{5.28}$$

In attempt to reduce the density overestimation in the tails of the distribution, we enforce a maximum radius, $r_{\max}$, and search only for k nearest neighbours within this radius.

5.4.3 Limiting search distance for Neighbours

By enforcing an upper bound on $r_k(x)$ (or $r_\omega(x)$), we aim to prevent the"borrowing of density sideways" phenomenon. Thus when dealing with unweighted data (or when all weights W equal 1), the k Nearest Neighbour density estimate with maximum search distance is computed as

$$\hat{f}(x) = \frac{k}{n} \frac{1}{c_d \, \min\{r_{max}, r_k(x)\}^d} . \mathbb{I}\{\rho(y_i, x) \leq min\{r_{max}, r_k(x)\}\}. \tag{5.29}$$

Similarly, for a weighted training set, the k Nearest Neighbour density estimate with a maximum search distance is given by

$$\hat{f}_\omega(x) = \frac{\sum_{i=1}^{n} W_i^*}{\sum_{i=1}^{n} W_i} \frac{1}{c_d \, min\{r_{max}, r_\omega(x)\}^d}, \tag{5.30}$$

where

$$W_i^* = \begin{cases} W_i & \text{if } \rho(y_i, x) \leq min\{r_{max}, r_\omega(x)\}), \\ 0 & \text{otherwise.} \end{cases} \tag{5.31}$$

Example 5.2. *We return to the same model as seen in Example 5.1, and compare the density estimates in the tail for the case where we use Equation (5.30) to the case where we do not alter our implementation for tail behaviour. For this example, we select* $r_{max} = 1.5$*, rather arbitrarily. Later when implementing algorithms we look at an iterative automatic method for selecting this maximum radius. We select* $k = 251$*, as by the guidance seen in Equation (5.27), and use the same training set of points as were used in Example 5.1.*

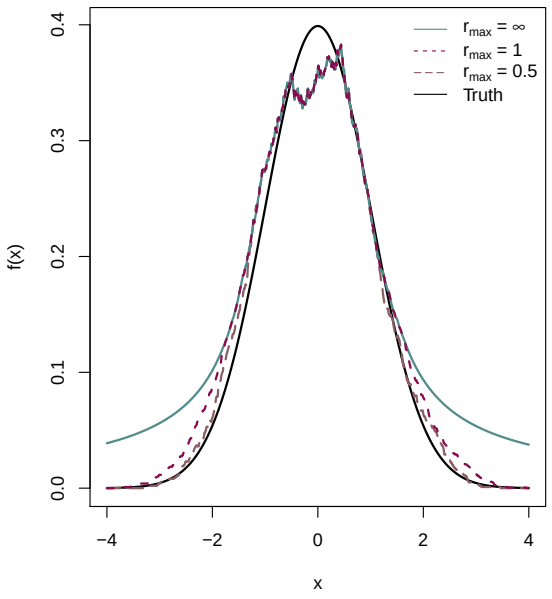


FIGURE 5.2: k nearest neighbour density estimation using a range of values of $r_{\max}$.

Figure 5.2 shows that in the centre of the distribution the knn density estimates with and without r_{max} *converge. However, in the region* $|x| > 2$*, the two estimates are no longer equal, with the method which incorporates* r_{max} *having lighter tails, and thus better fitting the true density curve. This result is echoed in Table 5.2, which shows that the two methods give equal density estimates at the point* $x = 0$*, but that the percentage error in the estimate at* $x = 3$ *when using* $r_{max} = 1$ *is less than one fifth of the error when there is no such maximum radius constraint.*

One criticism of the k Nearest Neighbour density estimation method, as given in Equation (5.23) is that, when estimating the density at x, the method only considers whether a sample point is

x	f(x)	$r_{\max} = 0.5$	$r_{\max} = 1$	$r_{\max} = \infty$
0	0.3989	0.3624 (9.16 %)	0.3624 (9.16 %)	0.3624 (9.16 %)
3	0.0044	0.003 (32.31 %)	0.0135 (205 %)	0.0536 (1109%)

TABLE 5.2: k-Nearest Neighbour density estimate for $F(x)$ at $x = 0$ and $x = 3$, using varying maximum radius. Percentage error is given in brackets.

closer or further away than the kth nearest neighbour of x. To illustrate the need for a more robust method, which takes into account the distance of each of the k closest neighbours of x, we consider the following example.

Example 5.3. *Let $x = 0$ be the point for which we wish to compute the kNN density estimate. Let $k = 5$, and consider the following two training data sets, $\mathcal{D}_1$ and $\mathcal{D}_2$ given by*

$$\mathcal{D}_1 = \{10, 10, 10, 10, 10\} \tag{5.32}$$

$$\mathcal{D}_2 = \{10, 1, 1, 0, 0\}. \tag{5.33}$$

Using either of the data sets $\mathcal{D}_1$ or $\mathcal{D}_2$ leads to the same estimate of $f(x)$ when using the kNN density estimate of Loftsgaarden et al. [59], as given in Equation (5.23). However, from looking at the two training sets we are led to believe that the density at $x = 0$ is much higher in the distribution from which the points in $\mathcal{D}_2$ are simulated, than in the distribution which gave rise to $\mathcal{D}_1$.

5.4.4 Kernel k Nearest Neighbour Density Estimation

Example 5.3 motivates a more general form of kNN density estimation. Mack and Rosenblatt [62] generalise the estimate of Loftsgaarden et al. [59] in the following way:

Let $K(\cdot) : \mathbb{R} \to \mathbb{R}^{\not\in}$ be a Kernel function, satisfying

$$\int_{-\infty}^{\infty} K(u)du = 1. \tag{5.34}$$

Then the kNN density estimate of $f(\cdot)$ at x is given by

$$\widehat{f}(x) = \frac{1}{nr_k(x)^d} \sum_{i=1}^{n} K\left(\frac{x - y_i}{r_k(x)}\right). \tag{5.35}$$

Thus we can see that the Loftsgaarden et al. [59] $k$nn density estimate is recovered in the case where $K(\cdot)$ is selected to be the uniform kernel, with bandwidth c_d.

We wish to adapt this density estimate further, to include a maximum radius, as we implemented in Section 5.4.3, and to include weighted data, as we saw in Equations (5.24) and (5.25).

To do so, we let

$$r = \min\{r_{\max}, r_\omega(x)\}, \tag{5.36}$$

where $r_\omega(x)$ is given in Equation (5.24), and $r_{\max}$ is a pre-defined maximum radius

Then the adapted k nearest neighbour density estimate of $f(\cdot)$ at x, based on a training set of data $y_1, \cdots, y_n$, with associated weights $W_1, \cdots, W_n$ is given by

$$\hat{f}_\omega(x) = \frac{\sum_{i=1}^{n} W_i^*}{\sum_{i=1}^{n} W_i} \frac{1}{r^d} \sum_{i=1}^{n} K\left(\frac{x - y_i}{r}\right), \tag{5.37}$$

where

$$W_i^* = \begin{cases} W_i & \text{if } \rho(y_i, x) \leq r\}), \\ 0 & \text{otherwise.} \end{cases} \tag{5.38}$$

For the remainder of this chapter, we select $K(\cdot)$ to be the Epanechnikov Kernel (Epanechnikov [64]), given by

$$K(u) = \begin{cases} \frac{3}{4}(1 - u^2) & \text{if } |u| < 1 \\ 0 & \text{otherwise.} \end{cases} \tag{5.39}$$

The Epanechnikov Kernel is desirable for our application since it is has bounded support, and minimises Asymptotic Mean Integrated Squared Error (Hodges et al. [65]).

Example 5.4. *We now repeat the example seen earlier in this chapter, using Equation (5.37) to compute the density estimates. Again k is selected to be 251, and the training set is the same as in the previous examples.*

x	f(x)	$r_{\max} = 0.5$	$r_{\max} = 1$	$r_{\max} = \infty$
0	0.3989	0.3870 (2.98 %)	0.3870 (2.98 %)	0.3870 (2.98 %)
3	0.0044	0.0024 (45 %)	0.0077 (75 %)	0.032 (627 %)

TABLE 5.3: k-Nearest Neighbour density estimate for $F(x)$ at $x = 0$ and $x = 3$, using varying maximum radius and an Epanechnikov Kernel. Percentage error is given in brackets.

Table 5.3 shows that incorporating the Kernel weights into the k nearest neighbour density estimate leads to improved estimates of the density, compared to the results given in Tables 5.2 and 5.1.

Example 5.5. *Univariate Gaussian with unknown mean. Consider a sample from the univariate Gaussian distribution with fixed variance $\sigma^2 = 4$. We use Algorithm 11 to obtain an estimate of the posterior distribution, $p(\mu|\boldsymbol{y})$, where $\boldsymbol{y} = y_1, \ldots, y_n$ denotes the observed data. We use a conjugate Gaussian prior, given by*

$$\mu \sim \mathcal{N}(\mu_0, \sigma_0^2) \tag{5.40}$$

where μ_0 and σ_0^2 are hyper-parameters.

The analytical posterior distribution is also Gaussian, with updated hyper-parameters given by

$$\mu_n = \frac{\frac{\mu_0}{\sigma_0^2} + \frac{\sum_{i=1}^n y_i}{\sigma^2}}{\frac{1}{\sigma_0^2} + \frac{n}{\sigma^2}} \tag{5.41}$$

$$\sigma_n^2 = \frac{\sigma_0^2 \sigma^2}{\sigma^2 + n\sigma_0^2}. \tag{5.42}$$

We select hyper-parameters $\mu_0 = 2.5$ and $\sigma_0^2 = 16$, and the observed data is such that

$$n^{-1} \sum_{i=1}^{n} y_i = 2.5. \tag{5.43}$$

We implement Algorithm 11, with $N_{init} = 1,000$ and $N = 500$. Figure 5.3 shows histograms of the estimates of the posterior distribution produced by the algorithm over a range of iterations.

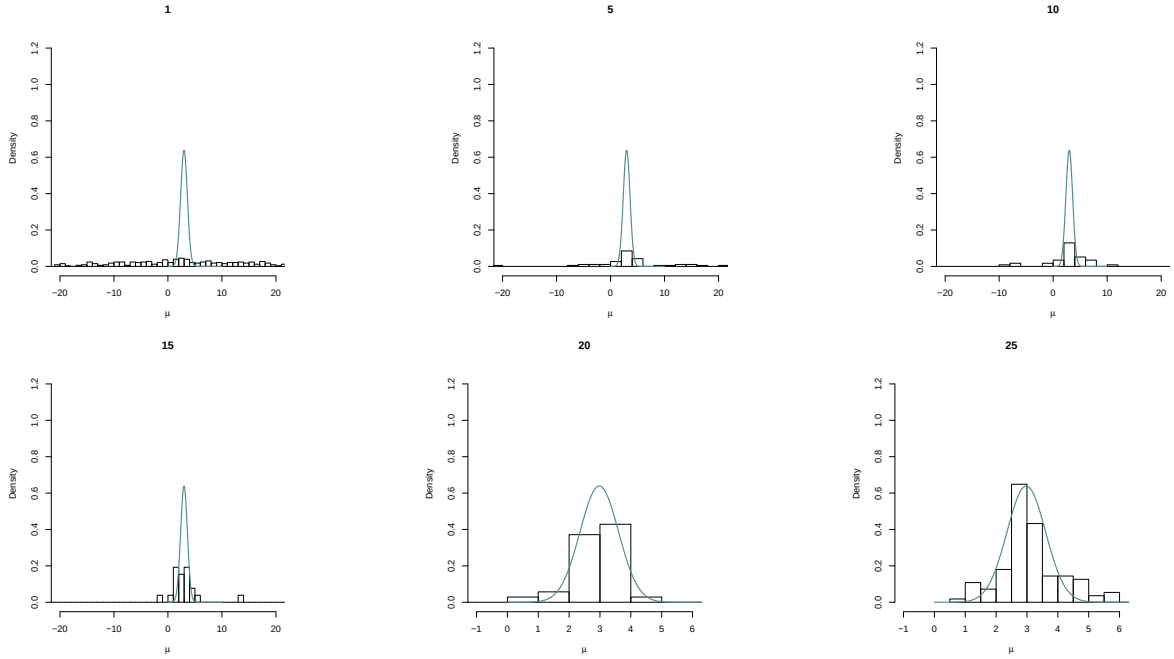


FIGURE 5.3: Analytic posterior distribution (blue) and posterior distribution given by Algorithm 11 (histogram), for the mean of the univariate Gaussian distribution.

From Figure 5.3 we see that, despite using a wide prior distribution for μ, the algorithm is able to approach the posterior distribution, within 25 iterations.

5.5 Discussion of the ISLE Algorithm

Although Algorithm 11 produced convincing results in Example 5.5, the algorithm is extremely inefficient to implement. The cause of the inefficiency is the need to evaluate α_j for each parameter, where

$$\alpha_j = \max_{\boldsymbol{x}} \frac{\widehat{p}_t(\boldsymbol{\theta}_j^{(t)}, S(\boldsymbol{x}))}{\pi(\boldsymbol{\theta}_j^{(t)})}. \tag{5.44}$$

In Example 5.5 the joint space of parameters and summary statistics is only two dimensional, yet at iteration $t = 1$, estimating this maximum, using the 'optim' function in R, takes over 0.5 seconds to run, and this time increases as the number of points in the joint sample increases, and as the number of neighbours k increases.

We developed the ISLE algorithm following the method of Tavaré et al. [4], which also accepted parameters based on the likelihood of the observed summary statistics, compared to the maximum likelihood for a given parameter value. However in the population genetics example in their paper, this maximum likelihood was tractable, and computable for a small computational cost. However, this tractability is uncommon for models on which we wish to implement ABC. With this in mind, we move forward with an alternative algorithm which does not require the computation of the maximum likelihood.

5.6 A Sequential Monte Carlo Synthetic likelihood approach, LE-SMC

The importance sampling algorithm we presented in Algorithm 11 requires knowledge of the maximum density, which we denote $\alpha_{\max}$. In practice this density is not known, so we estimate it as the maximum of all the densities we have previously estimated.

This approximation of the maximum is unlikely to be accurate, particularly in preliminary iterations and when the joint sample space $(\boldsymbol{\theta}, S(\boldsymbol{x}))$ is high dimensional. Consequently we wish to move away from this algorithm and instead implement the kNN density estimation within a Metropolis-Hastings algorithm, thus enabling us to make acceptance decisions on the comparison of two estimated likelihoods, rather than basing all acceptances on the estimated maximum, $\alpha_{\max}$.

We now present a Sequential Monte Carlo algorithm, with Metropolis Hastings updates, which bypasses the need to know a global maximum density, $\alpha_{\max}$. The algorithm follows a similar form to the SMC-ABC algorithm of Del Moral et al. [1]. Thinking of their algorithm as propagating N independent Markov Chains, we make two main amendments. The first is that we use an estimate of the likelihood in the Metropolis-Hastings algorithm, rather than a ratio of indicator functions. The standard MCMC-ABC Metropolis-Hastings ratio is given by

$$\alpha_j = min\left(1, \frac{\pi(\boldsymbol{\theta}'_j)}{\pi(\boldsymbol{\theta}_j^{(t)})} \frac{q_t(\boldsymbol{\theta}_j^{(t)}|\boldsymbol{\theta}'_j)}{q_t(\boldsymbol{\theta}'_j|\boldsymbol{\theta}_j^{(t)})} \mathbb{I}\{(d'_j < \epsilon_t)\}\right), \tag{5.45}$$

where $\boldsymbol{\theta}_j^{(t)}$ is the current parameter value of the jth chain, $\boldsymbol{\theta}'$ is the proposed parameter value and d'_j denotes the distance from the observed summary statistics, $S(\boldsymbol{y})$ to those simulated from parameter $\boldsymbol{\theta}'_j$. This acceptance ratio was first seen in Marjoram et al. [9].By replacing the ABC likelihood estimation, $\mathbb{I}\{(d'_j < \epsilon_t)\}$, with the nearest neighbour estimate of the likelihood, $\widehat{f}_t(\boldsymbol{\theta}, S(\boldsymbol{y}))$, the value α_j becomes

$$\alpha_j = \min\left(1, \frac{\widehat{f}_t(\boldsymbol{\theta}'_j, S(\boldsymbol{y}))}{\widehat{f}_t(\boldsymbol{\theta}_j^{(t)}, S(\boldsymbol{y}))} \frac{\pi(\boldsymbol{\theta}'_j)}{\pi(\boldsymbol{\theta}_j^{(t)})} \frac{q_t(\boldsymbol{\theta}_j^{(t)}|\boldsymbol{\theta}'_j)}{q_t(\boldsymbol{\theta}'_j|\boldsymbol{\theta}_j^{(t)})}\right) \tag{5.46}$$

$$= \min\left(1, \frac{\widehat{p}_t(S(\boldsymbol{y})|\boldsymbol{\theta}'_j)}{\widehat{p}_t(S(\boldsymbol{y})|\boldsymbol{\theta}_j^{(t)})} \frac{q_t(\boldsymbol{\theta}_j^{(t)}|\boldsymbol{\theta}'_j)}{q_t(\boldsymbol{\theta}'_j|\boldsymbol{\theta}_j^{(t)})}\right). \tag{5.47}$$

Thus α_j, as given in Equation (5.47), can be computed without estimating the maximum of the likelihood.

In Algorithm 12 we present a sequential Monte Carlo algorithm, LE-SMC, which targets the posterior distribution for observed summary statistics $S(\boldsymbol{y})$. The method uses the Metropolis-Hastings ratio from Equation (5.47), and follows the same structure as the SMC-ABC algorithm of Del Moral et al. [1], which is presented in Algorithm (4).

As for Algorithm 11, the algorithm is implemented using k nearest neighbour density estimation, as given in Equation (5.37), to estimate the joint density of parameters and summary statistics.

Algorithm 12 Likelihood Estimation Sequential Monte Carlo (LE-SMC),

Fix N . Let $\widehat{p}_t(\cdot,\cdot)$ denote the estimate of the joint distribution of parameters and summary statistics, computed using historical samples $\{\boldsymbol{\theta}^{(\tau)}, S(\boldsymbol{x}^{(\tau)})\}$, for all $\tau < t$.

1: Set $t = 0$.

For $j = 1, \ldots, N$:

a. Sample $\boldsymbol{\theta}_j^{(t)} \sim \pi(\cdot)$.

b. Simulate $\boldsymbol{x}_j^{(t)} \sim f(\cdot|\boldsymbol{\theta}_j^{(t)})$ and compute $S(\boldsymbol{x}_j^{(t)})$.

2: Use $\{\boldsymbol{\theta}_j^{(0)}, S(\boldsymbol{x}_j^{(0)})\}_{j=1}^N$ to build an estimate of the joint density, denoted $\hat{p}_0(\boldsymbol{\theta}, S(\boldsymbol{x}))$.

3: Set t=1.

For $j = 1, \ldots, N$:

a. Sample $\boldsymbol{\theta}_j^{(t)} \sim \pi(\cdot)$.

b. Simulate $\boldsymbol{x}_j^{(t)} \sim f(\cdot|\boldsymbol{\theta}_j^{(t)})$ and compute $S(\boldsymbol{x}_j^{(t)})$.

4: For $j = 1, \ldots, N$:

a. Sample $\boldsymbol{\theta}'_j \sim q_t(\cdot|\boldsymbol{\theta}_j^{(t)})$

b. Simulate $\boldsymbol{x}_j^{(t)} \sim f(\cdot|\boldsymbol{\theta}'_j)$ and compute $S(\boldsymbol{x}_j^{(t)})$.

c. Set

$$\alpha_j = \min\left(1, \frac{\widehat{p}_t(\boldsymbol{\theta}'_j, S(\boldsymbol{y}))}{\widehat{p}_t(\boldsymbol{\theta}_j^{(t)}, S(\boldsymbol{y}))} \frac{\pi(\boldsymbol{\theta}'_j)}{\pi(\boldsymbol{\theta}_j^{(t)})} \frac{q_t(\boldsymbol{\theta}_j^{(t)}|\boldsymbol{\theta}'_j)}{q_t(\boldsymbol{\theta}'_j|\boldsymbol{\theta}_j^{(t)})}\right). \tag{5.48}$$

d. Simulate $u_j \sim \mathcal{U}(0,1)$.

e. If $u_j \leq \alpha_j$,

set $\boldsymbol{\theta}_j^{(t+1)} = \boldsymbol{\theta}'_j$

Else,

set $\boldsymbol{\theta}_j^{(t+1)} = \boldsymbol{\theta}_j^{(t)}$.

5: Resample N parameters from the set of $\boldsymbol{\theta}_j^{(t)}$, with probability proportional to $\hat{f}_t(\boldsymbol{\theta}|S(\boldsymbol{y}))$.

6: While the algorithm has not converged, update $\widehat{p}_t(\cdot,\cdot)$, set $t = t + 1$ and return to step 4.

Example 5.6. *Univariate Gaussian Distribution with Unknown Mean.*

We return to Example 5.5, and consider a Gaussian distribution with unknown mean, μ. Prior and posterior distributions are selected as in the previous example. In Example 5.5, at least 25 iterations of Algorithm 11 were required to obtain a posterior distribution which fit well to the contours of the true posterior distribution. However, for this simple univariate example, we found that only two iterations of the LE-SMC algorithm were required to accurately target the true posterior distribution. Figure 5.4 shows the output of the LE-SMC algorithm, obtained at $t = 1$ and $t = 2$, using $N = 1,000$ particles. The sample at $t = 1$ is drawn directly from the prior distribution, in step 3 of Algorithm 12. The distribution for $t = 2$ is obtained after one pass through step 4 of Algorithm 12.

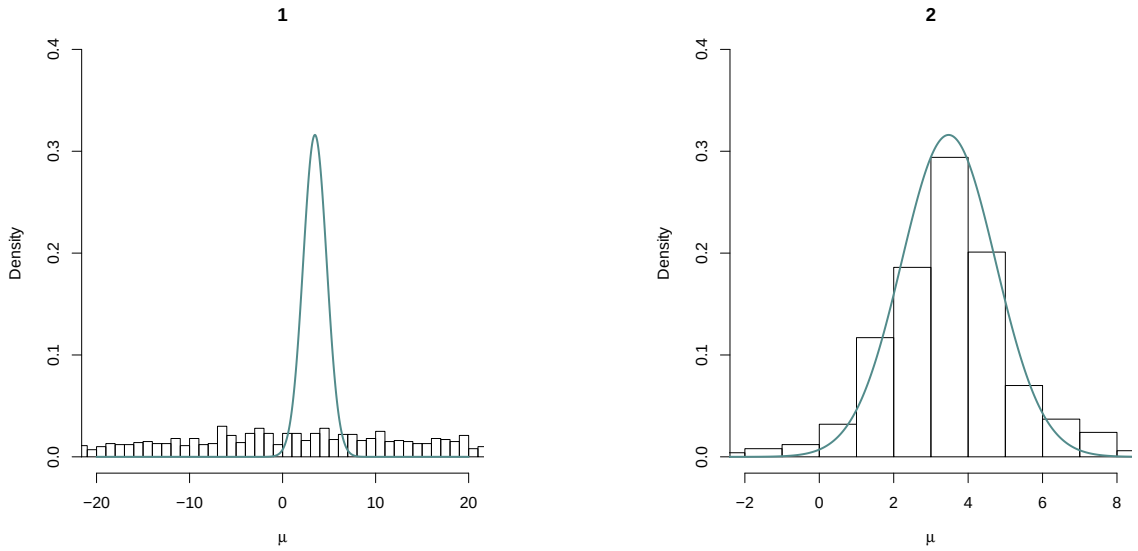


FIGURE 5.4: Analytic posterior distribution (green) and posterior distribution given by Algorithm 12 (histogram), for the mean of the univariate Gaussian distribution, with $N = 1000$.

Figure 5.5 shows that, even for $N = 100$, Algorithm 12 gives posterior distributions which are a reasonable fit to the analytic posterior distribution after $t = 5$ iterations.

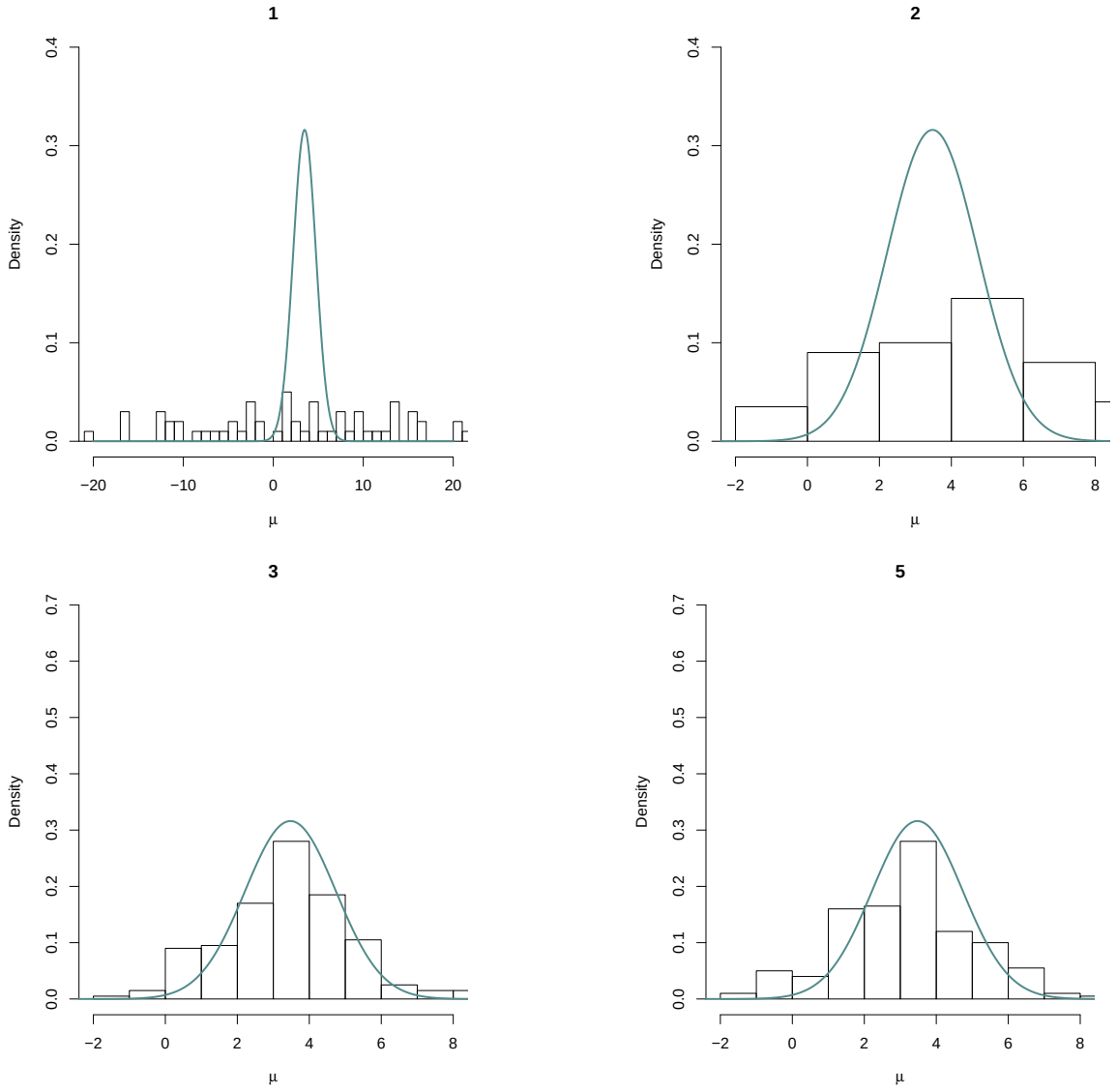


FIGURE 5.5: Analytic posterior distribution (blue) and posterior distribution given by Algorithm 12 (histogram), for the mean of the univariate Gaussian distribution, using $N = 100$.

In this simple example, where the joint space $p(\boldsymbol{\theta}, S(\boldsymbol{x}))$ *is two dimensional, it is the case that a sample of size* $1,000$*, drawn from the joint distribution, is enough to give an accurate estimate of the joint density. As such, LE-SMC requires so few (only 1) iterations to obtain an accurate estimate of the posterior distribution. This, in turn, leads to a very small overall computation time. For a sample size of* $N = 100$ *we see that the algorithm takes slightly longer to converge, since the estimate of the joint distribution is less accurate, and there are fewer samples drawn from the areas of non-negligible posterior mass. However, after 3 iterations the algorithm is producing*

accurate estimates of the posterior distribution, and thus the algorithm could be terminated, with a very low overall computational cost.

Example 5.7. *Bivariate Gaussian with unknown mean and covariance We return to the Bivariate Gaussian distribution with unknown mean* $\mu \in \mathbb{R}^2$*, and unknown covariance matrix* $\Sigma \in \mathbb{M}_+^{2\times 2}$*. This was first introduced in Example 2.1, and we use the same prior distributions and hyper-parameters as before, given by*

$$\Sigma \sim Inv - Wishart(\nu_0, \Lambda_0^{-1}) \tag{5.49}$$

$$\boldsymbol{\mu}|\Sigma \sim \mathcal{N}(\boldsymbol{\mu}_0, \Sigma/k_0), \tag{5.50}$$

where

$$\mu_0 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \quad \kappa_0 = 1, \quad \nu_0 = 4 \quad \textit{and} \quad \Lambda_0^{-1} = \begin{pmatrix} \frac{4}{3} & -\frac{2}{3} \\ -\frac{2}{3} & \frac{4}{3} \end{pmatrix}. \tag{5.51}$$

The observed data, $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{10})$ *is simulated by*

$$\boldsymbol{y}_i \sim^{iid} \mathcal{N}\left(\begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0.9 & 0.5 \\ 0.5 & 0.9 \end{pmatrix} \right). \tag{5.52}$$

The data is summarised by sufficient statistics, given by the sample mean and the sample covariance. Again the observed summary statistics are

$$\bar{y} = \begin{pmatrix} -1.0136 \\ 0.9764 \end{pmatrix} \quad \textit{and} \quad S_{\boldsymbol{y}} = \begin{pmatrix} 0.8820 & 0.1474 \\ 0.1474 & 0.8425 \end{pmatrix}. \tag{5.53}$$

This model has 5 *unique summary statistics and* 5 *unique parameters (since the diagonal elements of the covariance matrix are equal). Thus the k nearest neighbour density estimation is being carried out on the 10-dimensional space of parameters and summary statistics. To ensure that the distance metric in the k Nearest Neighbour density estimation equally weights points in each dimensions, we scale both the parameters and summary statistics by the marginal standard deviation of a sample from the joint distribution, drawn at time* $t = 0$*. We set* $N = 1,000$ *and use the proposal distributions given in Section 4.1.2.*

Figure 5.6 shows that, even for this 10 dimensional example, Algorithm 10 quickly gives a good approximation for the posterior distribution of $\boldsymbol{\mu}$. Even after 1 iteration, which required only $1,000$ simulations from the model, the approximation of the posterior distribution is fairly accurate to the contours. Figure 5.7 shows the mean squared error of the output produced by Algorithm 12, over 100 iterations. The figure shows that, for all parameters, the mean squared error decreases rapidly initially, then the decrease becomes slow. Such a diagnostic could be used to guide a stopping rule for the algorithm in further work.

Table 5.4 shows that the mean squared error of the outputs of LE-SMC decrease greatly between iteration 0 and 1, after which point the decrease is minimal.

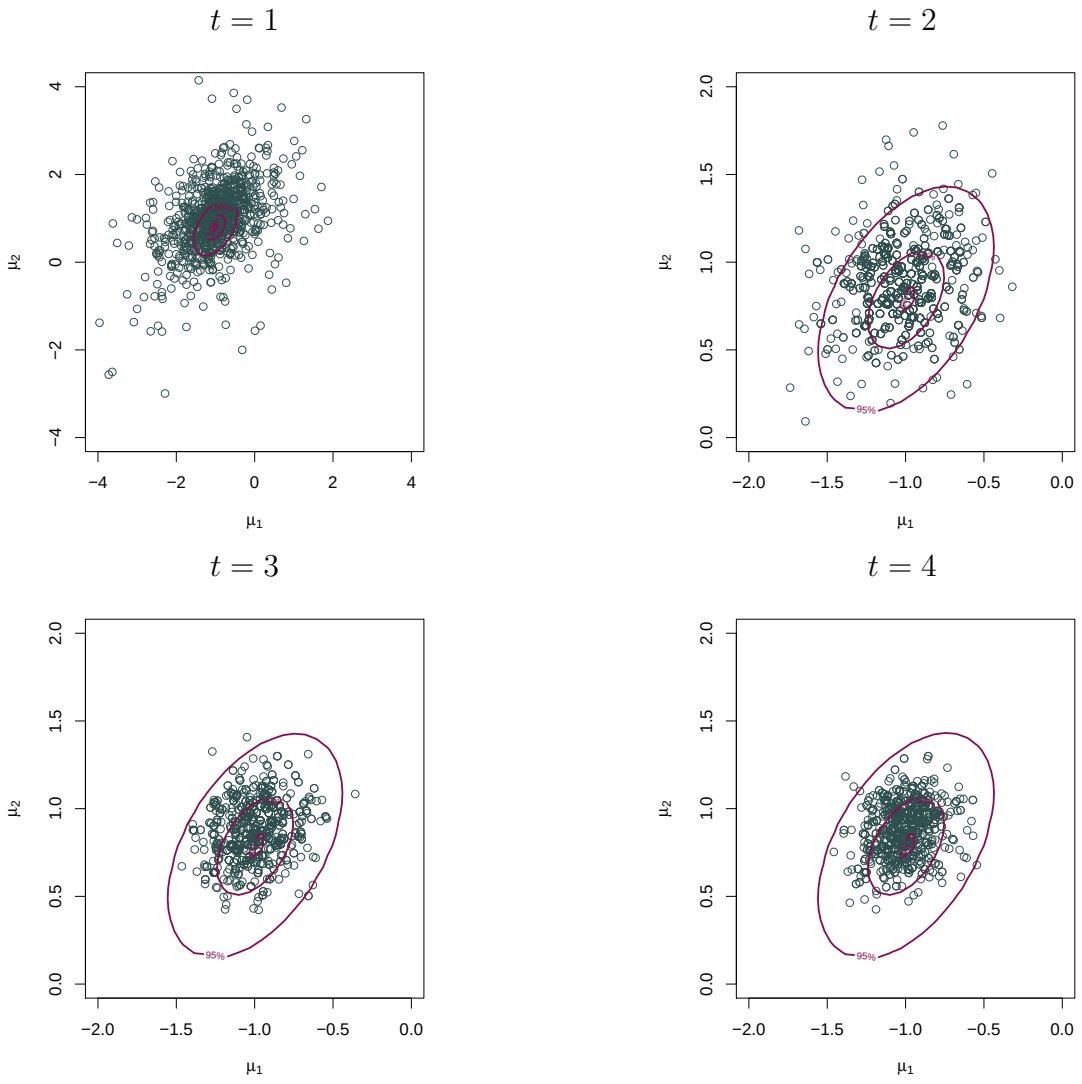


FIGURE 5.6: Samples for $\boldsymbol{\mu}$, drawn from the posterior distribution produced by Algorithm 12. The true posterior contours are shown in pink.

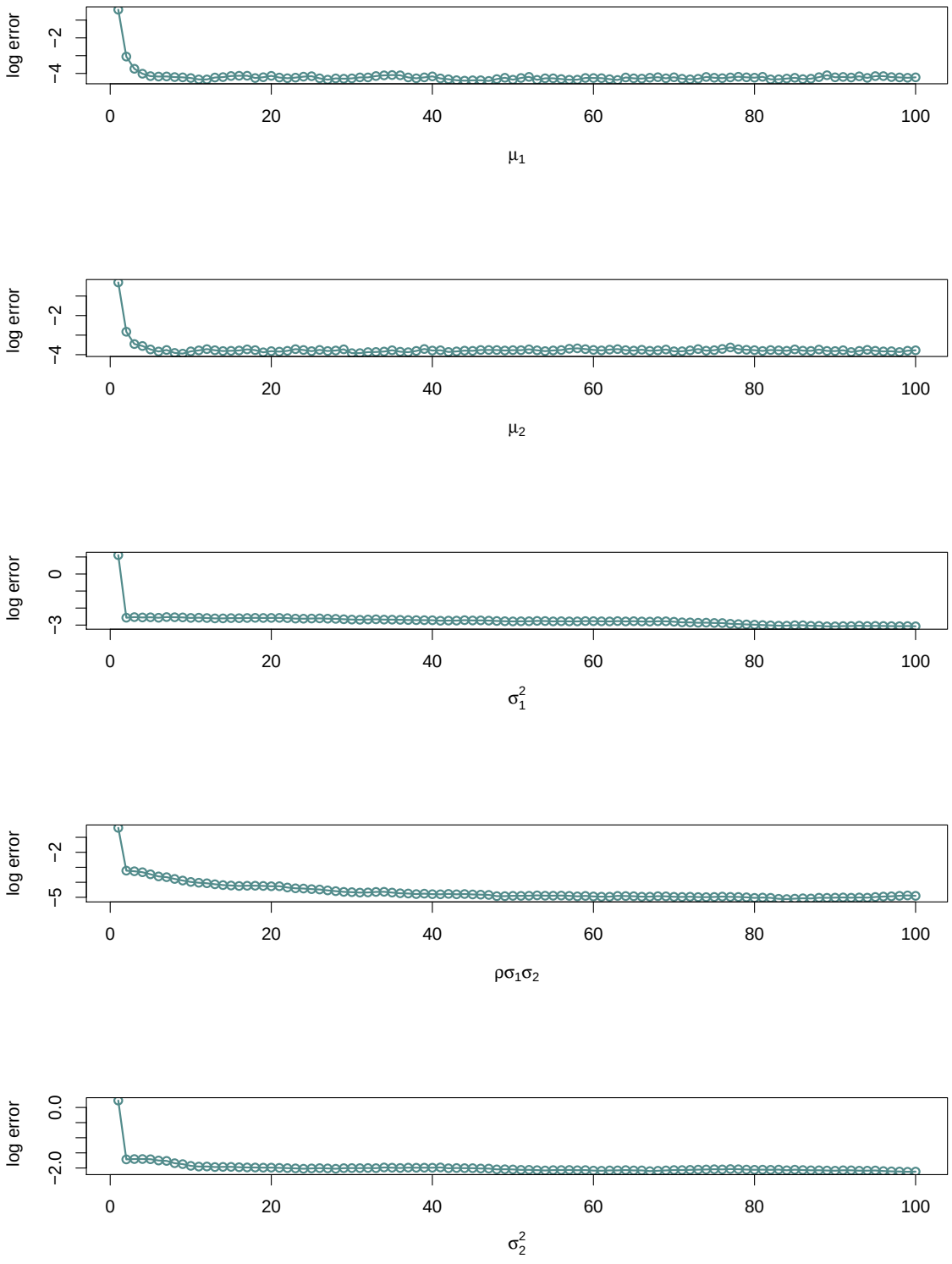


FIGURE 5.7: Mean Squared error in the output of Algorithm 12, on a log scale, compared to the true posterior mean, over 100 iterations.

Iteration	μ_1	μ_2	$\sigma^2_{1,1}$	$\sigma_1\sigma_2\rho$	$\sigma^2_{2,2}$
$t = 0$	0.661	0.738	3.029	0.692	1.257
$t = 1$	0.047	0.059	0.077	0.040	0.180
$t = 3$	0.024	0.032	0.079	0.038	0.182
$t = 4$	0.018	0.029	0.078	0.036	0.182

TABLE 5.4: Mean Squared Error for the output of the LE-SMC algorithm, applied to the Bivariate Gaussian distribution, using sufficient summary statistics.

We now repeat the same analysis, but use the 14 naive statistics, given in Table 4.1. We keep $n = 10$*, and use the same prior distribution and hyper-parameters as above. Again, we scale the parameters and summary statistics by the marginal standard deviation of an initial sample from the joint distribution. Table 5.5 shows that, as with sufficient statistics, the algorithm reaches low values of mean squared error in very few iterations.*

Despite the joint distribution now being 19 dimensional, and using a training set of only $N = 1000$ *samples to guide the initial density estimation, the LE-SMC Algorithm still gives accurate estimates of the posterior mean,* $\boldsymbol{\mu}$*, after 2 iterations, as shown by Figure 5.8. However, the posterior distribution does appear to be biased towards the value of* μ_2*. The probable cause of this is the scaling used in the example. The joint space of parameters and summary statistics is scaled by the marginal standard deviations obtained from the* 1000 *samples drawn at time* $t = 0$*. Computing the marginal standard deviations from a larger initial sample will likely lead to better inference.*

Figure 5.9 shows again that the mean squared error of the posterior distributions decreases rapidly initially, and tends towards 0 for four of the parameters, though remains at around 0.2 for σ_2^2*.*

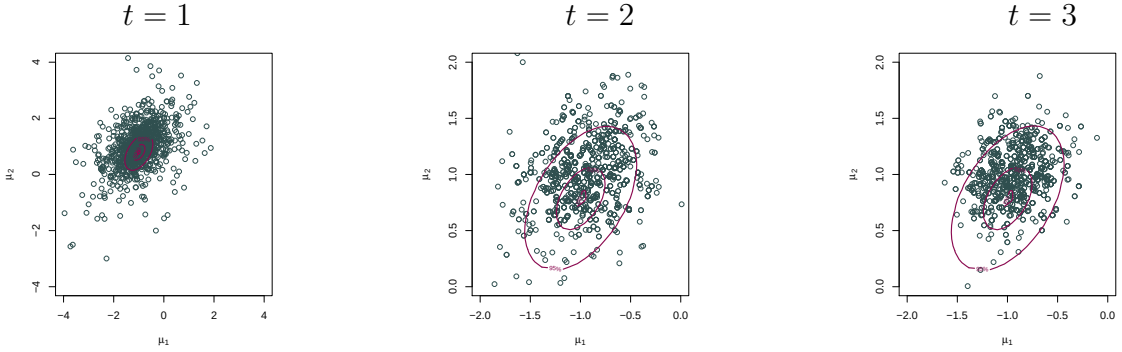


FIGURE 5.8: Samples for $\boldsymbol{\mu}$, drawn from the posterior distribution produced by Algorithm 12, using naive summary statistics. The true posterior contours are shown in pink.

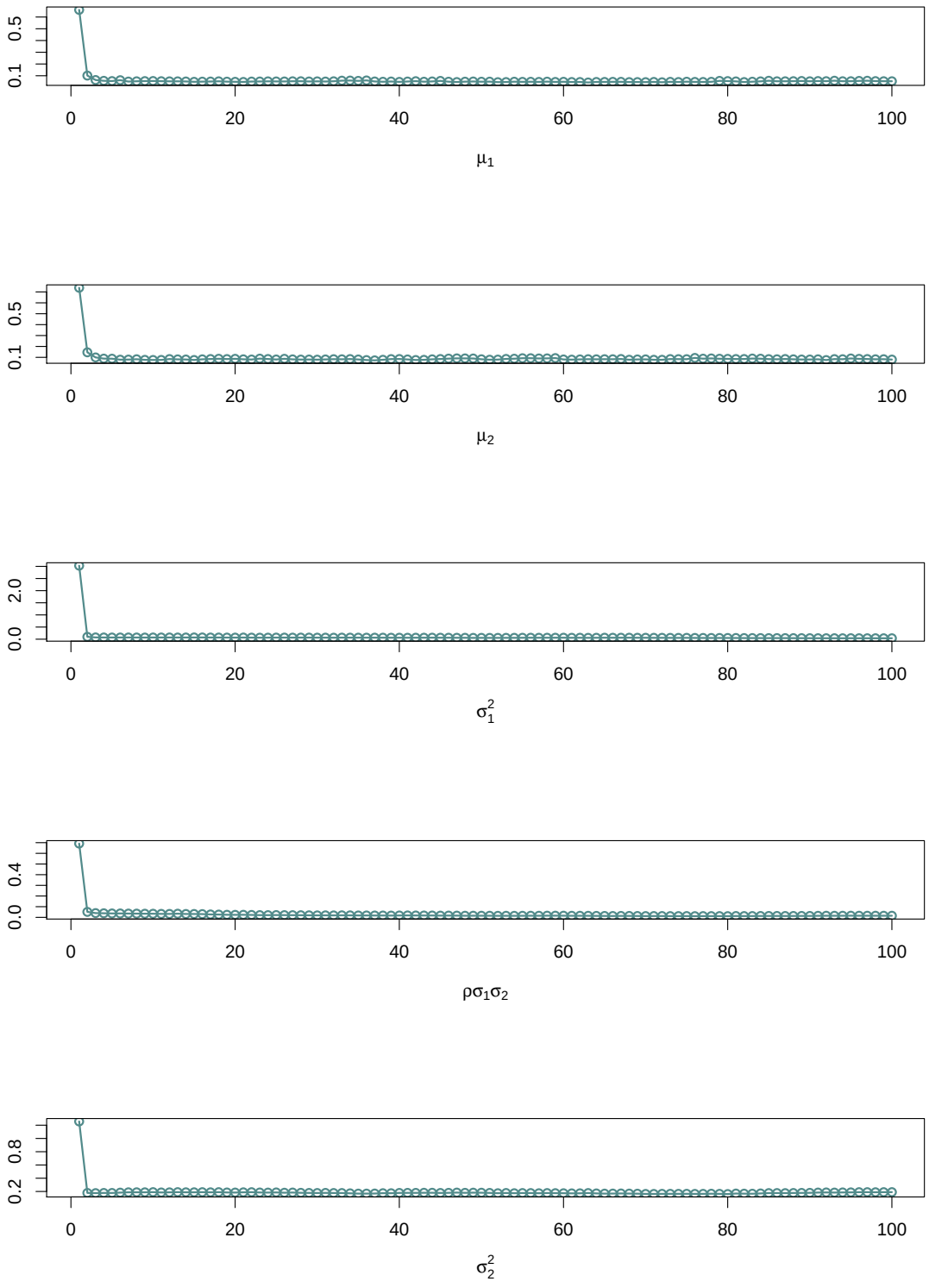


FIGURE 5.9: Mean squared error of the output of Algorithm 12, compared to the true posterior mean, over 100 iterations, using naive summary statistics

Iteration	μ_1	μ_2	$\sigma^2_{1,1}$	$\sigma_1\sigma_2\rho$	$\sigma^2_{2,2}$
$t = 0$	0.661	0.738	3.029	0.692	1.257
$t = 1$	0.101	0.146	0.097	0.051	0.178
$t = 3$	0.063	0.099	0.073	0.040	0.175
$t = 4$	0.057	0.088	0.071	0.038	0.170

TABLE 5.5: Mean Squared Error for the output of the LE-SMC algorithms, applied to the Bivariate Gaussian distribution, using naive summary statistics.

The results of the LE-SMC algorithm in Example 5.7 suggested that a large computational saving can be made through modelling the likelihood function, compared to implementing the ABC methods we saw earlier in the Thesis. The algorithm requires little user tuning, with only a choice of number of particles N, and proposal distribution $q(\cdot)$ to be made.

This chapter illustrates that there is the potential to develop and implement algorithms which model the likelihood function, efficiently and without the need for vast user tuning. We showed that, when estimating the density of a 19 dimensional space, k Nearest Neighbour density estimation leads to fairly accurate results, and is easily amendable to incorporate weighted samples from the distribution of interest. However, the density estimation method is sensitive to the scaling of the samples, and more work should be done to further investigate the impact of different scaling regimes.

Chapter 6

Conclusions and Further Work

This thesis has focused on sequential Monte Carlo methods for approximate Bayesian computation. I have developed new methods, which I have shown give more favourable results on a number of examples, requiring less user tuning and fewer model simulations.

I begin Chapter 2 with a review of existing ABC methodologies, and compare standard methods on a data set from the Bivariate Gaussian distribution with conjugate prior distributions. In this Chapter I illustrate the duality between the choice of summary statistics and the choice of distance metric. Specifically, I show that we can think of scaling summary statistics as using unscaled summary statistics with an alternative distance metric. I use this notion in Chapter 3 to develop an iterative, automatic method for selecting summary statistics.

In Chapter 3 I give an overview of the SMC-ABC algorithm of Del Moral et al. [1] and explore the impact of different tuning parameters on the resultant posterior distribution, for the Bivariate Gaussian example first seen in Chapter 2. By considering the algorithm's behaviour I was able to propose an alternative stopping rule for the algorithm, which removes one tuning parameter from the algorithm, namely ϵ_{final}, and removes the risk of stopping the algorithm too early.

In Chapter 3 I also discuss a way to reduce the number of simulations from the model within any ABC algorithm that uses a Metropolis-Hastings kernel. By splitting the Metropolis-Hastings ratio into two separate ratios, I show that, in the ABC setting, fewer simulations from the model are required, leading to more efficient implementations.

Finally, in Chapter 3 I present an iterative summary statistic selection method which works within the SMC-ABC algorithm, localising the Automatic summary statistic selection method of Fearnhead and Prangle [22] as the SMC-ABC algorithm moves towards the posterior distribution.

Our three contributions in Chapter 3 are included in Algorithm 8, Auto-SS SMC-ABC.

In Chapter four I implement the Auto-SS SMC-ABC algorithm on a range of examples, and compared the performance to existing ABC methods, namely standard Rejection ABC, Regression based methods, and SMC-ABC with a range of levels of summary statistic selection. I show that the Auto-SS SMC-ABC algorithm performs favourably in some cases.

One such model I consider is the earthworms model. This highlighted that there is much more work to be done in the case where the model is misspecified: Regression based methods, of which Auto-SS SMC-ABC is one, are not appropriate in such examples, since for many summary statistics, the observed summary statistics from the experimental data lie outside the range of summary statistics that can be simulated under the model. Because of this, regression based methods which aim to make inference at the observed summary statistics lead to extrapolation. Further work could be aimed at investigating robust and automated methods for dealing with model misspecification. Existing work in this area has been done by Wilkinson [66], and Ratmann et al. [67], and explored how to model the discrepancy between the input and the observed data. However, I feel that it would be worth investigating the merits of transforming the summary statistics of the model into broader summaries, such as through wavelet decomposition, to capture the general properties of the observed summary statistics.

For the population growth model I considered in Chapter 3, the worst results were obtained using the Auto-SS SMC-ABC algorithm. This was because the assumption of a locally linear relationship between the parameters and the parameters and summary statistics is not valid in this example. Thus I show that SMC-ABC without summary statistic selection gives the most accurate ABC posterior distributions for this example.

In Chapter 5 I explore an alternative class of ABC algorithms, which do not have a notion of distance ϵ. This work was motivated by Wood [50], and led to more efficient implementations in Wilkinson [51], and Meeds and Welling [52]. I present two original algorithms, both of which remove the need for vast amounts of user tuning. I was able to approximate the likelihood by

using nearest neighbour density estimation to approximate the joint density of parameters and summary statistics, then dividing by the prior density. By iteratively drawing more samples from the joint distribution in the region around the observed data we see that the estimate of the likelihood in this region improves over time, and thus leads to better posterior inference. Although I show that the first algorithm correctly targets the posterior distribution for a toy example, it was extremely inefficient due to the need to estimate the maximum likelihood for each sampled parameter. Therefore I developed a second algorithm that uses a Metropolis-Hastings kernel to make accept or reject decisions, and thus the estimate of the maximum likelihood is no longer required. I show that this algorithm leads to good posterior inference on medium dimensional spaces. There is an opportunity for more work to compare the results to that of the other algorithms, and implement on high dimensional spaces.

Bibliography

[1] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. An adaptive sequential monte carlo method for approximate bayesian computation. *Statistics and Computing*, 22(5):1009–1020, 2012.

[2] Donald B. Rubin. Bayesianly justifiable and relevant frequency calculations for the applied statistician. *Ann. Statist.*, 12(4):1151–1172, 12 1984. doi: 10.1214/aos/1176346785. URL http://dx.doi.org/10.1214/aos/1176346785.

[3] Jonathan K Pritchard, Mark T Seielstad, Anna Perez-Lezaun, and Marcus W Feldman. Population growth of human y chromosomes: a study of y chromosome microsatellites. *Molecular Biology and Evolution*, 16(12):1791–1798, 1999.

[4] Simon Tavaré, David J Balding, Robert C Griffiths, and Peter Donnelly. Inferring coalescence times from dna sequence data. *Genetics*, 145(2):505–518, 1997.

[5] Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*, volume 2. Chapman & Hall/CRC Boca Raton, FL, USA, 2014.

[6] Mark A Beaumont, Wenyang Zhang, and David J Balding. Approximate bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002.

[7] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[8] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[9] Paul Marjoram, John Molitor, Vincent Plagnol, and Simon Tavaré. Markov chain monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26):15324–15328, 2003.

[10] Trevelyan McKinley, Alex R Cook, Robert Deardon, et al. Inference in epidemic models without likelihoods. *The International Journal of Biostatistics*, 5(1):1–40, 2009.

[11] Luke Bornn, Natesh S. Pillai, Aaron Smith, and Dawn Woodard. The use of a single pseudo-sample in approximate bayesian computation. *Statistics and Computing*, 27(3):583–590, May 2017. ISSN 1573-1375. doi: 10.1007/s11222-016-9640-7. URL https://doi.org/10.1007/s11222-016-9640-7.

[12] Paola Bortot, Stuart G Coles, and Scott A Sisson. Inference for stereological extremes. *Journal of the American Statistical Association*, 102(477):84–92, 2007.

[13] Scott A Sisson, Yanan Fan, and Mark M Tanaka. Sequential monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6):1760–1765, 2007.

[14] Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

[15] Mark A Beaumont, Jean-Marie Cornuet, Jean-Michel Marin, and Christian P Robert. Adaptive approximate bayesian computation. *Biometrika*, 96(4):983–990, 2009.

[16] Tina Toni, David Welch, Natalja Strelkowa, Andreas Ipsen, and Michael PH Stumpf. Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31):187–202, 2009.

[17] Scott A Sisson. Correction for sisson et al., sequential monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 106(39):16889, 2009. doi: 10.1073/pnas.0908847106. URL http://www.pnas.org/content/106/39/16889.1.short.

[18] M. G. B. Blum, M. A. Nunes, D. Prangle, and S. A. Sisson. A Comparative Review of Dimension Reduction Methods in Approximate Bayesian Computation. *ArXiv e-prints*, February 2012.

[19] Paul Joyce, Paul Marjoram, et al. Approximately sufficient statistics and bayesian computation. *Statistical applications in genetics and molecular biology*, 7(1):26, 2008.

[20] Jean-Michel Marin, Pierre Pudlo, Christian P Robert, and Robin J Ryder. Approximate bayesian computational methods. *Statistics and Computing*, pages 1–14, 2012.

[21] Matthew A Nunes and David J Balding. On optimal selection of summary statistics for approximate bayesian computation. *Statistical applications in genetics and molecular biology*, 9(1), 2010.

[22] Paul Fearnhead and Dennis Prangle. Constructing summary statistics for approximate bayesian computation: semi-automatic approximate bayesian computation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(3):419–474, 2012.

[23] Katalin Csillery, Olivier Francois, and Michael G. B. Blum. abc: an r package for approximate bayesian computation (abc). *Methods in Ecology and Evolution*, 2012. doi: http://dx.doi.org/10.1111/j.2041-210X.2011.00179.x.

[24] Dennis Prangle. Adapting the abc distance function. *Bayesian Analysis*, 12(1):289–309, 2017.

[25] Michael GB Blum and Olivier François. Non-linear regression models for approximate bayesian computation. *Statistics and Computing*, 20(1):63–73, 2010.

[26] Jun S Liu. *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2008.

[27] Peter H Peskun. Optimum monte-carlo sampling using markov chains. *Biometrika*, 60(3): 607–612, 1973.

[28] Richard G Everitt and Paulina A Rowińska. Delayed acceptance abc-smc. *arXiv preprint arXiv:1708.02230*, 2017.

[29] Umberto Picchini and Julie Lyng Forman. Accelerating inference for diffusions observed with measurement error and large sample sizes using approximate bayesian computation. *Journal of Statistical Computation and Simulation*, 86(1):195–213, 2016.

[30] D. Ruppert and M. P. Wand. Multivariate locally weighted least squares regression. *The Annals of Statistics*, 22(3):1346–1370, 1994. ISSN 00905364. URL http://www.jstor.org/stable/2242229.

[31] Michele Ann Haynes. *Flexible distributions and statistical models in ranking and selection procedures with applications*. PhD thesis, Queensland University of Technology, 1998.

[32] Glen D Rayner and Helen L MacGillivray. Numerical maximum likelihood estimation for the g-and-k and generalized g-and-h distributions. *Statistics and Computing*, 12(1):57–75, 2002.

[33] David Allingham, RAR King, and Kerrie L Mengersen. Bayesian estimation of quantile distributions. *Statistics and Computing*, 19(2):189–201, 2009.

[34] Dennis Prangle. *gk: g-and-k and g-and-h Distribution Functions*, 2017. URL https://CRAN.R-project.org/package=gk. R package version 0.5.0.

[35] ASA Johnston, Mark Edward Hodson, Pernille Thorbek, Tania Alvarez, and RM Sibly. An energy budget agent-based model of earthworm populations and its application to study the effects of pesticides. *Ecological modelling*, 280:5–17, 2014.

[36] Uri Wilensky. Netlogo. 1999.

[37] Richard M. Sibly, Volker Grimm, Benjamin T. Martin, Alice S. A. Johnston, Katarzyna Kułakowska, Christopher J. Topping, Peter Calow, Jacob Nabe-Nielsen, Pernille Thorbek, and Donald L. DeAngelis. Representing the acquisition and use of energy by individuals in agent-based models of animal populations. *Methods in Ecology and Evolution*, 4(2):151–161, 2013. ISSN 2041-210X. doi: 10.1111/2041-210x.12002. URL http://dx.doi.org/10.1111/2041-210x.12002.

[38] Bintoro Gunadi, Charles Blount, and Clive A Edwards. The growth and fecundity of eisenia fetida (savigny) in cattle solids pre-composted for different periods. *Pedobiologia*, 46(1):15–23, 2002.

[39] Bintoro Gunadi and Clive A Edwards. The effects of multiple applications of different organic wastes on the growth, fecundity and survival of eisenia fetida (savigny)(lumbricidae). *Pedobiologia*, 47(4):321–329, 2003.

[40] AJ Reinecke and SA Viljoen. The influence of feeding patterns on growth and reproduction of the vermicomposting earthworm eisenia fetida (oligochaeta). *Biology and Fertility of Soils*, 10(3):184–187, 1990.

[41] Michael Goldstein and Jonathan Rougier. Probabilistic formulations for transferring inferences from mathematical models to physical systems. *SIAM journal on scientific computing*, 26(2):467–487, 2004.

[42] Elske van der Vaart, Mark A Beaumont, Alice SA Johnston, and Richard M Sibly. Calibration and evaluation of individual-based models using approximate bayesian computation. *Ecological Modelling*, 312:182–190, 2015.

[43] Elske van der Vaart, Mark Beaumont, Alice Johnston, and Richard Sibly. ABC for IBMs - Earthworm Runs & Results of Checks. 6 2015. doi: 10.6084/m9.figshare.1412720.v1. URL https://figshare.com/articles/ABC_for_IBMs_Earthworm_Runs_amp_Results_of_Checks/1412720.

[44] Howard Levene et al. Robust tests for equality of variances. *Contributions to probability and statistics*, 1:278–292, 1960.

[45] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

[46] Mark A Beaumont. Joint determination of topology, divergence time, and immigration in population trees. 2008.

[47] Jean-Marie Cornuet, Filipe Santos, Mark A Beaumont, Christian P Robert, Jean-Michel Marin, David J Balding, Thomas Guillemaud, and Arnaud Estoup. Inferring population history with diy abc: a user-friendly approach to approximate bayesian computation. *Bioinformatics*, 24(23):2713–2719, 2008.

[48] Richard R Hudson. Generating samples under a wright–fisher neutral model of genetic variation. *Bioinformatics*, 18(2):337–338, 2002.

[49] Matthew Stephens and Peter Donnelly. Inference in molecular population genetics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):605–635, 2000.

[50] Simon N Wood. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466(7310):1102–1104, 2010.

[51] Richard D Wilkinson. Accelerating abc methods using gaussian processes. *arXiv preprint arXiv:1401.1436*, 2014.

[52] Edward Meeds and Max Welling. Gps-abc: Gaussian process surrogate approximate bayesian computation. *arXiv preprint arXiv:1401.2838*, 2014.

[53] Matteo Fasiolo, Natalya Pya, Simon N Wood, et al. A comparison of inferential methods for highly nonlinear state space models in ecology and epidemiology. *Statistical Science*, 31 (1):96–118, 2016.

[54] Richard G Everitt, Adam M Johansen, Ellen Rowing, and Melina Evdemon-Hogan. Bayesian model comparison with un-normalised likelihoods. *Statistics and Computing*, 27 (2):403–422, 2017.

[55] Florian Hartig, Claudia Dislich, Thorsten Wiegand, and Andreas Huth. Approximate bayesian parameterization of a process-based tropical forest model. *arXiv preprint arXiv:1401.8205*, 2014.

[56] Leah F Price, Christopher C Drovandi, Anthony Lee, and David J Nott. Bayesian synthetic likelihood. *Journal of Computational and Graphical Statistics*, (just-accepted), 2017.

[57] George Papamakarios and Iain Murray. Fast ε-free inference of simulation models with bayesian conditional density estimation. In *Advances in Neural Information Processing Systems*, pages 1028–1036, 2016.

[58] Fernando V Bonassi, Lingchong You, and Mike West. Bayesian learning from marginal data in bionetwork models. *Statistical applications in genetics and molecular biology*, 10(1), 2011.

[59] Don O Loftsgaarden, Charles P Quesenberry, et al. A nonparametric estimate of a multivariate density function. *The Annals of Mathematical Statistics*, 36(3):1049–1051, 1965.

[60] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[61] Alina Beygelzimer, Sham Kakade, and John Langford. Cover trees for nearest neighbor. In *Proceedings of the 23rd international conference on Machine learning*, pages 97–104. ACM, 2006.

[62] YP Mack and Murray Rosenblatt. Multivariate k-nearest neighbor density estimates. *Journal of Multivariate Analysis*, 9(1):1–15, 1979.

[63] Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[64] Vassiliy A Epanechnikov. Non-parametric estimation of a multivariate probability density. *Theory of Probability & Its Applications*, 14(1):153–158, 1969.

[65] JL Hodges, EL Lehmann, et al. The efficiency of some nonparametric competitors of the t-test. *The Annals of Mathematical Statistics*, 27(2):324–335, 1956.

[66] Richard David Wilkinson. Approximate bayesian computation (abc) gives exact results under the assumption of model error. *Statistical applications in genetics and molecular biology*, 12(2):129–141, 2013.

[67] Oliver Ratmann, Pierre Pudlo, Sylvia Richardson, and Christian Robert. Monte carlo algorithms for model assessment via conflicting summaries. *arXiv preprint arXiv:1106.5919*, 2011.