Sequential Monte Carlo Methods for Epidemic Data

Jessica Welding

Department of Mathematics and Statistics

Submitted for the degree of Doctor of Philosophy at

Lancaster University

January 2020

Sequential Monte Carlo Methods for Epidemic Data

Jessica Welding

Submitted for the degree of Doctor of Philosophy at Lancaster University

January 2020

Abstract

Epidemics often occur rapidly, with new cases being observed daily. Due to the frequently severe social and economic consequences of an outbreak, this is an area of research that benefits greatly from online inference. This motivates research into the construction of fast, adaptive methods for performing real-time statistical analysis of epidemic data.

The aim of this thesis is to develop sequential Monte Carlo (SMC) methods for infectious disease outbreaks. These methods utilize the observed removal times of individuals, obtained throughout the outbreak. The SMC algorithm adaptively generates samples from the evolving posterior distribution, allowing for the real-time estimation of the parameters underpinning the outbreak. This is achieved by transforming the samples when new data arrives, so that they represent samples from the posterior distribution which incorporates all of the data.

To assess the performance of the SMC algorithm we additionally develop a novel Markov chain Monte Carlo (MCMC) algorithm, utilising adaptive proposal schemes to improve its mixing. We test the SMC and MCMC algorithms on various simulated outbreaks, finding that the two methods produce comparable results in terms of parameter estimation and disease dynamics. However, due to the parallel nature of the SMC algorithm it is computationally much faster.

The SMC and MCMC algorithms are applied to the 2001 UK Foot-and-Mouth outbreak: notable for its rapid spread and requirement of control measures to contain the outbreak. This presents an ideal candidate for real-time analysis. We find good agreement between the two methods, with the SMC algorithm again much quicker than the MCMC algorithm. Additionally, the performed inference matches well with previous work conducted on this data set.

Overall, we find that the SMC algorithm developed is suitable for the real-time analysis of an epidemic and is highly competitive with the current gold-standard of MCMC methods, whilst being computationally much quicker.

Acknowledgements

First I must credit the financial support offered to me by the EPSRC, which allowed me to complete this work.

I must also thank my eternally optimistic supervisor, Pete Neal, I am truly grateful for your supervision and guidance during my PhD, as well as your seemingly endless patience! I additionally extend my gratitude to Selina Wang, for teaching me to see problems from a different perspective and being an inspiring person to have worked with.

Thanks also to my ever-understanding family and friends, for listening to me moan and understanding when I'm grouchy because everything has broken. Thank you for the cat pictures, the food and cake supplies, for teaching and guiding me to try new things and providing endless (lively!) discussions.

Finally I have to thank my partner David, for being generally a wonderful human being/support team/chef/bug-fixer/bug-catcher/player 2.

Declaration

I declare that this thesis is my own work and has not been submitted in substantially the same form for the award of a higher degree elsewhere.

Jessica Welding

Contents

List of Figures

List of Tables

Chapter 1

Introduction

For many years statisticians have played a pivotal role in furthering the understanding of infectious disease outbreaks (see, for example, Bartlett (1949), Bailey and Thomas (1971), Becker (1979), Gibson (1997), Jewell et al. (2009), Deardon et al. (2010) and Stockdale et al. (2017)). The aim has always remained the same: to gain an understanding of the properties which allowed an epidemic to occur, and thus produce strategies for preventing future severe outbreaks.

Epidemics can be an incredibly destructive occurrence: causing the loss of harvests, livestock and often lives. In recent years we have seen many instances of such consequences; from the heavy financial burden of the UK Foot-and-Mouth epidemic, estimated at costing over £3 billion to the public sector and over £5 billion to the private sector (UK National Audit Office (2002)), to the devastating loss of life in the recent Ebola outbreak, an estimated 28,616 cases resulting in 11,310 deaths (WHO (2016)). By modelling epidemics we can gain vital insight, that is key to understanding and limiting the severity of future outbreaks of infectious diseases.

With the rapid advancement in technology we have gained the ability to collect vast amounts of information about an outbreak. Increasingly this data is extremely rich and often obtained instantaneously, throughout the course of an epidemic. With such data readily available, epidemic modelling has gained the capability to move from retrospective analysis, to real-time analysis. Swiftly obtaining information about the characteristics of an epidemic can then help to inform on control measures that can then be put in place during an active outbreak.

Drawing upon two strands of research, simulation methods and epidemic modelling, we aim to illustrate a novel way of utilising advances in computing power to construct a method of inferring the underlying parameters of an infectious disease outbreak, in real time.

1.1 Thesis Structure

This thesis is concerned with the construction of a sequential method of analysing epidemic data in real time. We will describe the formulation of a generic algorithm, for use in conjunction with epidemic data, and then apply it to both simulated and real data sets.

Chapter 1: Introduction

In this chapter we introduce the Bayesian paradigm and the concept of simulation methods. We then proceed to discuss simple simulation techniques such as inverse, rejection and importance sampling as well as more complex methods such as Markov chain Monte Carlo (MCMC) and sequential Monte Carlo (SMC). We aim to provide an overview of these methods and describe when each is most appropriate to use.

Chapter 2: Epidemic Modelling

In this chapter we introduce epidemic modelling. We begin by describing the key choices we must consider prior to analysing outbreak data. This is then followed by a brief overview of the historical and present work performed within this field. We also consider a selection of epidemic models in detail, specifically: the deterministic model, the Reed-Frost chain binomial model and the general stochastic epidemic model.

Chapter 3: Developing Sequential Monte Carlo Methods for Epidemic Data

This chapter is where we construct the sequential Monte Carlo algorithm that forms the focus of this thesis. We begin by outlining the discrete-time stochastic epidemic model we will use throughout, before forming the posterior distribution which will be the focus of our analysis. Once constructed we use the methods discussed in Chapters 1 and 2 to construct a novel MCMC algorithm, with an emphasis on ensuring we obtain an optimal acceptance rate. We then proceed to developing the SMC algorithm, with an in-depth discussion of each of its steps.

Chapter 4: A Comprehensive Simulation Study

In this chapter we conduct an in-depth study of the performance of the SMC algorithm on multiple simulated outbreaks. We illustrate the application of the SMC algorithm and compare it to the current 'gold-standard' of MCMC methods. This is with the aim of better understanding the performance and behaviour of the SMC algorithm we have developed.

Chapter 5: UK Foot-and-Mouth Disease Outbreak (2001)

In this chapter we apply the MCMC and SMC algorithms of Chapter 3 to the 2001 UK Foot-and-Mouth outbreak. We begin by reviewing the previous methods used to analyse this outbreak, as well as describing their key findings. Using this we then outline the assumptions we make when working with this data set and then discuss, in detail, the results we obtain. We compare the output generated using the SMC algorithm to both the results produced by the MCMC, as well as the work previously conducted on the Foot-and-Mouth data set.

Chapter 6: Conclusions

Finally, in this chapter we summarise the overall conclusions of the work in this thesis and propose future extensions to the SMC methods developed.

1.2 Bayesian Framework

1.2.1 Motivation

Within statistics we are often presented with situations in which we are required to make inference about an unknown set of parameters. Without any formal observations we may make an initial prediction about their form, for example using relevant previous research. If we then receive data, which is dependent on these parameters, it would be wasteful to fully discard our previous conclusions; instead we can update them using this new information. We therefore have made a prior estimate of the parameters and then updated our estimates post-observation. This is the underlying motivation behind the Bayesian framework.

1.2.2 The Posterior Distribution

Formally, let $\boldsymbol{\theta}$ denote the unknown parameters of interest and $\boldsymbol{x}$ the observed data. We wish to find the conditional distribution of $\boldsymbol{\theta}$ given $\boldsymbol{x}$, defined by $\pi(\boldsymbol{\theta} \mid \boldsymbol{x})$ and referred to as the *posterior distribution.* We assign to $\boldsymbol{\theta}$ a *prior distribution*, defined as $\pi(\boldsymbol{\theta})$, which is chosen to represent our current knowledge about $\boldsymbol{\theta}$ and which we choose prior to the collection of the data. We will discuss the form of the prior distribution in Section 1.2.3.

Once a prior distribution has been chosen we can use *Bayes' theorem* to construct the posterior distribution:

$$\pi(\boldsymbol{\theta} \mid \boldsymbol{x}) \;=\; \frac{\pi(\boldsymbol{\theta},\, \boldsymbol{x})}{\pi(\boldsymbol{x})} \;=\; \frac{\pi(\boldsymbol{x} \mid \boldsymbol{\theta})\, \pi(\boldsymbol{\theta})}{\pi(\boldsymbol{x})} \propto \mathcal{L}(\boldsymbol{\theta}\,;\boldsymbol{x})\, \pi(\boldsymbol{\theta}), \tag{1.2.1}$$

where $\mathcal{L}(\boldsymbol{\theta}\,;\boldsymbol{x})$ is the *likelihood* and treated as a function of $\boldsymbol{\theta}$. In the denominator we have $\pi(\boldsymbol{x})$: this is the normalising constant and will often not have a tractable form. This intractability is a major impediment to Bayesian inference. However, as this is independent of $\boldsymbol{\theta}$, we will often be able to avoid its calculation altogether, for example using MCMC methods (see Section 1.4).

1.2.3 The Prior Distribution

Clearly the choice of prior distribution will have an impact on the form of the posterior distribution. If we do not know much about the parameters then we may choose an *uninformative prior* (also called *non-informative* or *diffuse* prior), this form of prior only provides general information about the nature of $\boldsymbol{\theta}$. For example, if we allocate equal weight to all values $\boldsymbol{\theta}$ could take then this would be an uninformative prior distribution. This form of prior maximises the information about $\boldsymbol{\theta}$ provided by the data, $\boldsymbol{x}$. Conversely we could choose an *informative prior.* This form of prior can arise if we have some definite knowledge about the form $\boldsymbol{\theta}$ will take, for example from previous research.

One well-used group of prior distributions are *conjugate priors.* These are chosen such that the posterior and prior are from the same class of distributions. This has the advantage that the posterior distribution has a closed form, which can ease the computational burden during analysis. This class of prior distributions will be of particular importance when discussing *Gibbs sampling* in Section 1.4.3.

Example: Conjugate Prior

Suppose that we have $\boldsymbol{x} = (x_1, \ldots, x_n)$: n independent and identically distributed observations from a Poisson(θ) distribution. Then the likelihood is of the form

$$L(\theta\,;\boldsymbol{x}) \,\propto\, \theta^{\sum\limits_{j=1}^{n} x_j}\, e^{-\theta n}. \tag{1.2.2}$$

If we select Gamma(α, β) as the prior distribution for θ then the posterior distribution is

$$\pi(\theta\,|\,\boldsymbol{x}) \,\propto\, \left(\theta^{\sum\limits_{j=1}^{n} x_j}\, e^{-\theta n}\right) \left(\theta^{\alpha-1}\, e^{-\beta\theta}\right) \tag{1.2.3}$$

and therefore,

$$\pi(\theta\,|\,\boldsymbol{x}) \,\sim\, \text{Gamma}\left(\sum_{j=1}^{n} x_j + \alpha,\, n+\beta\right). \tag{1.2.4}$$

We see that both the posterior and prior belong to the Gamma class of distributions.

Overall it will often be that the data, and therefore the likelihood, dominates the posterior distribution and thus the prior distribution will be less influential. Therefore, although $\pi(\boldsymbol{\theta})$ must be chosen with care, it will not be the focus of our discussions, and throughout we will primarily use uninformative priors. Once the posterior distribution has been determined we can analyse it as we would any other distribution.

1.3 Introduction to Simple Simulation Methods

Suppose that we have constructed the posterior distribution and find that it has a complex, often high-dimensional, form. How can we obtain useful information about such a distribution? The underlying idea behind simulation methods is well described by Halton (1970):

> "*representing the solution of a problem as a parameter of a hypothetical population, and using a random sequence of numbers to construct a sample of the population, from which statistical estimates of the parameter can be obtained*".

Thus, if we have a method of sampling from some population then we can utilize these samples to estimate various statistical quantities about the distribution of interest. There exist many algorithms for computing such samples from a given distribution; we will discuss some of those most commonly used in the subsequent sections. However, first we illustrate in the next section how to use such samples to generate quantities of interest.

The set-up we shall use to discuss the methods will remain the same throughout: we are interested in a random variable, Θ, with probability density function, $\pi(\theta)$. However, we should note that the methods we will describe can be extended to more complex, high-dimensional, problems.

1.3.1 Perfect Monte Carlo Sampling

Frequently, we will be concerned with evaluating integrals of the form

$$\mathbb{E}_\pi[h(\Theta)] \;=\; \int \pi(\theta)\, h(\theta)\, d\theta. \tag{1.3.1}$$

However, direct calculation of (1.3.1) will often be impossible. Alternatively, if we have independent and identically distributed (i.i.d.) samples $\theta^{(1)}, \ldots, \theta^{(n)} \sim \pi$, then we can estimate (1.3.1) as

$$\hat{\mathbb{E}}_\pi[h(\Theta)] \;=\; \frac{1}{n}\sum_{j=1}^{n} h\big(\theta^{(j)}\big). \tag{1.3.2}$$

By the *Strong Law of Large Numbers* (*SLLN*) if $\mathbb{E}_\pi[h(\theta^{(i)})] = \mathbb{E}_\pi[h(\Theta)] < \infty$ then

$$\lim_{n\to\infty} \hat{\mathbb{E}}_\pi[h(\Theta)] \;=\; \mathbb{E}_\pi[h(\Theta)]. \tag{1.3.3}$$

This method requires i.i.d. samples from the distribution of interest. Unfortunately, many distributions are not easy to sample from, therefore producing estimates such as (1.3.2) is not straightforward. This provides the motivation for the remainder of this chapter where we discuss various methods for simulating samples from a distribution of interest. These can then be utilised to estimate quantities such as (1.3.1).

In the following sections we will discuss three well-studied simulation methods. We begin by considering one of the simplest simulation methods, *inversion sampling*, in Section 1.3.2. This method is incredibly simple to use, however, it can only be applied in a limited number of situations. Following this we shall discuss the more flexible *rejection sampling* in Section 1.3.3, this uses an intermediate distribution to facilitate generating samples from the target distribution. Finally in Section 1.3.4 we discuss *importance sampling*, this shares many characteristics with rejection sampling in that we use an intermediate distribution to generate samples from the target distribution, although now we do not 'reject' any of the samples.

1.3.2 Inversion Sampling

The first method we consider is *inversion sampling*, one of the most intuitive methods of generating samples from a target distribution. We begin by generating pseudo-random samples, typically using a computer, from a uniform, $U(0,1)$, distribution. The idea underpinning the *inversion method* is to then apply a transformation to such realisations, in order to generate samples from the distribution we desire.

We denote the cumulative distribution function (cdf) of the target distribution by $F(\theta) = P(\Theta \leq \theta)$. As suggested by its name, this simulation method requires the inverse of the cdf, denoted F^{-1}; however, this does not necessarily exist in a closed form. Instead we will define the *generalised inverse* as $F^{-}(\phi) = \inf\{\theta : F(\theta) \geq \phi\}$, when F is strictly increasing and continuous we have $F^{-1}(\phi) = F^{-}(\phi)$. Once the generalised inverse has been found we can use Theorem 1 to generate n samples from the distribution of interest, as displayed formally in Algorithm 1.

Theorem 1 (The Inversion Theorem)**.**
Let F be a cdf and F^{-} be its generalised inverse. If $\Phi \sim U(0,1)$ then $F^{-}(\Phi)$ has cdf F.

Proof. We start by observing that, as F^{-} is an increasing function, for all θ

$$F^{-}(\Phi) \leq \theta \iff \Phi \leq F(\theta).$$

Thus, for $\Phi \sim U(0,1)$,

$$P(F^{-}(\Phi) \leq \theta) = P(\Phi \leq F(\theta)) = F(\theta).$$

□

Algorithm 1: Inversion Sampling

```
1. for $j = 1, \ldots, n$
    (i). Sample $\phi^{(j)} \sim U(0, 1)$.
    (ii). Let $\theta^{(j)} = F^{-}(\phi^{(j)})$.
```

Assuming we can find F^{-}, this method is highly efficient and very straightforward to implement. However, in practice, there are few distributions for which F^{-} has a closed form, especially for higher dimensional problems. Devroye (1986) contains examples of when this method can be used, as well as an extension to using numerical solutions if an explicit form of the inverse cannot be found.

1.3.3 Rejection Sampling

The next simulation method we consider is *rejection sampling*, introduced by von Neumann (1951). The idea of rejection sampling (also referred to as the *Accept-Reject method*) is to first sample from an intermediate distribution, called the *proposal distribution*, and then accept or reject these samples as from the distribution we desire, according to some probability. This method aims to accept those samples which are most likely to have come from the target distribution.

We are interested in a target distribution with density π. Suppose that we have access to a second density, g, such that $\forall \theta \in \Omega$, $\pi(\theta) \leq Mg(\theta)$, for some $M > 1$, where Ω is the support of π. This condition ensures that $Mg(\theta)$ fully envelopes the target distribution. To generate samples from π rejection sampling uses Algorithm 2.

Algorithm 2: Rejection Sampling

```
1. Suppose we desire a sample of size $n$, then let $j = 0$.
2. while $j < n$
    (i). Generate a proposal sample, $\theta^* \sim g$.
    (ii). Calculate the acceptance probability,
          $$\alpha = \frac{\pi(\theta^*)}{Mg(\theta^*)}.$$
    (iii). Generate $u \sim U(0,1)$.
    (iv). if $u \leq \alpha$ then
        (a) Set $j = j + 1$.
        (b) Accept $\theta^{(j)} = \theta^*$.
```

We can easily see why using Algorithm 2 produces i.i.d. samples from the correct distribution. Let $\mathcal{X}$ be a subset of Ω and $\Theta^* \sim g$ then,

$$P(\Theta^* \text{ is accepted}) = \int \frac{\pi(\theta)}{Mg(\theta)} g(\theta)\, d\theta = \frac{1}{M}, \tag{1.3.4}$$

$$P(\Theta^* \in \mathcal{X} \text{ and is accepted}) = \int_{\mathcal{X}} \frac{\pi(\theta)}{Mg(\theta)} g(\theta)\, d\theta = \frac{1}{M}\int_{\mathcal{X}} \pi(\theta)\, d\theta. \tag{1.3.5}$$

Therefore we find,

$$P(\Theta^* \in \mathcal{X} \mid \Theta^* \text{ accepted}) = \frac{P(\Theta^* \in \mathcal{X} \text{ and is accepted})}{P(\Theta^* \text{ is accepted})} = \int_{\mathcal{X}} \pi(\theta)\, d\theta. \tag{1.3.6}$$

This states that the density of the accepted samples is the same as the target density, as required. Therefore the rejection sampling algorithm successfully uses samples generated from g to produce samples from π. To illustrate the intuition behind the rejection sampling algorithm we shall apply it to a simple example.

Example: Rejection Sampling

Suppose that we want to generate samples from a mixture of Beta distributions, with density function

$$f(x) = \frac{3}{10}\left(\frac{x^{10-1}(1-x)^{20-1}}{B(10,20)}\right) + \frac{7}{10}\left(\frac{x^{20-1}(1-x)^{10-1}}{B(20,10)}\right),$$

using a Beta$(3,2)$ as the proposal distribution, with density

$$g(x) = \frac{x^{3-1}(1-x)^{2-1}}{B(3,2)}.$$

Here $B(a,b)$ denotes the beta function. We display the two distributions in Figure 1.1. To find the value of M we will use the `optimize` function in R. We find $M = 1.83$ and show in Figure 1.2 that this value of M ensures that we satisfy the required condition and that this choice is optimal.

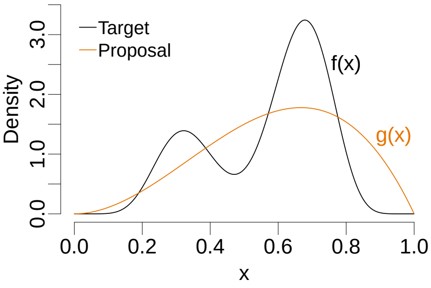


Figure 1.1: The target (black) and proposal (orange) density functions.

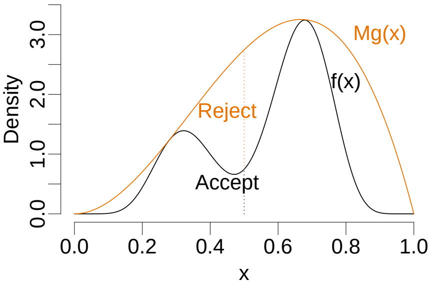


Figure 1.2: The target and scaled proposal densities.

We continue generating samples until we have accepted 500 samples from the target distribution, the results of which can be seen in Figure 1.3. As we can see in Figure 1.4 these are from the correct distribution.

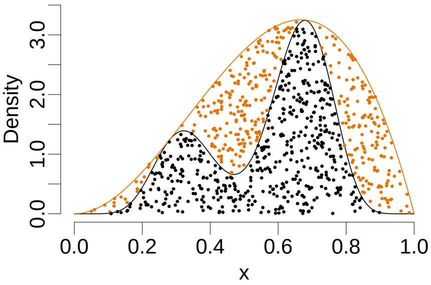


Figure 1.3: The samples accepted (black) and rejected (orange).

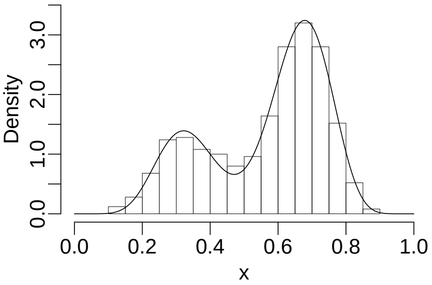


Figure 1.4: A histogram of the accepted samples with the truth overlaid.

Rejection sampling proves to be an effective simulation method as it only requires knowledge of the target density up to a constant of proportionality. However, it is by no means a perfect method: often finding an appropriate proposal distribution is not straightforward. We require the proposal distribution to have thicker tails than the target distribution in order for π/g to be bounded. For example, we could not use a Normal proposal to generate samples from a Cauchy-type distribution, although the reverse would work. Additionally, as the dimension of the distribution increases, it can become difficult to choose a proposal distribution that produces a usable acceptance rate. Rejection sampling is covered in detail in Robert and Casella (2005, Section 2.3), where it is called the *Accept-Reject* method.

1.3.4 Importance Sampling

The next simulation method we consider is *importance sampling.* This shares many characteristics with rejection sampling, however, instead of rejecting samples we will attach a weight to them.

We once again have a target distribution with density $\pi(\theta)$ and also assume that we have access to a proposal distribution with density $g(\theta)$, from which we can sample. The motivation underlying importance sampling is the observation that

$$P(\Theta \in \mathcal{X}) = \int_{\mathcal{X}} \pi(\theta)\, d\theta = \int_{\mathcal{X}} g(\theta)\frac{\pi(\theta)}{g(\theta)}\, d\theta = \int_{\mathcal{X}} g(\theta)w(\theta)\, d\theta, \tag{1.3.7}$$

for all measurable $\mathcal{X}$, where $w(\theta) := \frac{\pi(\theta)}{g(\theta)}$ is referred to as the *importance weight* and g is often referred to as the *importance distribution.* This naturally leads to the following relationship

$$\mathbb{E}_\pi[h(\Theta)] = \int h(\theta)\pi(\theta)\, d\theta = \int h(\theta)w(\theta)g(\theta)\, d\theta = \mathbb{E}_g[h(\Theta)w(\Theta)]. \tag{1.3.8}$$

If we can generate samples $\theta^{(1)}, \ldots, \theta^{(n)} \sim g$ then, under some mild assumptions (Geweke (1989)), we can use the Strong Law of Large Numbers to find

$$\frac{1}{n}\sum_{j=1}^{n} h\big(\theta^{(j)}\big)w\big(\theta^{(j)}\big) \xrightarrow[n\to\infty]{a.s.} \mathbb{E}_g[h(\Theta)w(\Theta)] = \mathbb{E}_\pi[h(\Theta)]. \tag{1.3.9}$$

As such we can estimate $\mathbb{E}_\pi[h(\Theta)]$ using

$$\hat{\mathbb{E}}_\pi[h(\Theta)] = \frac{1}{n}\sum_{j=1}^{n} h\big(\theta^{(j)}\big)w\big(\theta^{(j)}\big). \tag{1.3.10}$$

We can easily see that this will be an unbiased estimator of $\mathbb{E}_\pi[h(\Theta)]$.

The weights, $w\big(\theta^{(1)}\big), \dots, w\big(\theta^{(n)}\big)$, may not necessarily add up to n, therefore often of use is the *self-normalised* estimator,

$$\tilde{\mathbb{E}}_\pi[h(\Theta)] = \frac{1}{\sum_{i=1}^{n} w\big(\theta^{(i)}\big)}\sum_{j=1}^{n} h\big(\theta^{(j)}\big)w\big(\theta^{(j)}\big). \tag{1.3.11}$$

$\tilde{\mathbb{E}}_\pi[h(\Theta)]$ will also converge to $\mathbb{E}_\pi[h(\Theta)]$ although it is now a biased estimator, however, it can result in a smaller mean square error than the unbiased estimator (Liu (2008, Chapter 2)). Additionally, the biased estimator is useful for when we only know the target distribution up to proportionality. To see this we assume that the target distribution is known only up to a constant of proportionality, such that $\pi(\theta) = c\bar{\pi}(\theta)$, for some constant c. Then we can see that, in order to be computed, the self-normalised estimator does not require knowledge of c:

$$\begin{aligned}
\tilde{\mathbb{E}}_\pi[h(\Theta)] &= \frac{\sum_{j=1}^{n} h\big(\theta^{(j)}\big)w\big(\theta^{(j)}\big)}{\sum_{i=1}^{n} w\big(\theta^{(i)}\big)} \\
&= \frac{\sum_{j=1}^{n} h\big(\theta^{(j)}\big)\frac{\pi\left(\theta^{(j)}\right)}{g\left(\theta^{(j)}\right)}}{\sum_{i=1}^{n} \frac{\pi\left(\theta^{(i)}\right)}{g\left(\theta^{(i)}\right)}} \\
&= \frac{\sum_{j=1}^{n} h\big(\theta^{(j)}\big)\frac{c\bar{\pi}\left(\theta^{(j)}\right)}{g\left(\theta^{(j)}\right)}}{\sum_{i=1}^{n} \frac{c\bar{\pi}\left(\theta^{(i)}\right)}{g\left(\theta^{(i)}\right)}} \\
&= \frac{\sum_{j=1}^{n} h\big(\theta^{(j)}\big)\frac{\bar{\pi}\left(\theta^{(j)}\right)}{g\left(\theta^{(j)}\right)}}{\sum_{i=1}^{n} \frac{\bar{\pi}\left(\theta^{(i)}\right)}{g\left(\theta^{(i)}\right)}}. \qquad (1.3.12)
\end{aligned}$$

We can use a similar argument to show that we only need to know the proposal distribution, g, up to a multiplicative constant. Altogether, we note that for the biased estimator we only need to know the importance weight, $\frac{\pi(\theta)}{g(\theta)}$, up to a multiplicative constant. We illustrate using importance sampling in Algorithm 3.

Algorithm 3: Importance Sampling

```
1. for $j = 1, \ldots, n$
      (i). Sample $\theta^{(j)} \sim g$.
      (ii). Calculate $w\big(\theta^{(j)}\big) = \frac{\pi(\theta^{(j)})}{g(\theta^{(j)})}$.
2. Estimate $\mathbb{E}_\pi[h(\Theta)]$ as either
      $$\hat{\mathbb{E}}_\pi[h(\Theta)] = \frac{1}{n}\sum_{j=1}^{n} h\big(\theta^{(j)}\big)w\big(\theta^{(j)}\big) \qquad (1.3.13)$$
   or
      $$\tilde{\mathbb{E}}_\pi[h(\Theta)] = \frac{1}{\sum_{i=1}^{n} w\big(\theta^{(i)}\big)}\sum_{j=1}^{n} h\big(\theta^{(j)}\big)w\big(\theta^{(j)}\big). \qquad (1.3.14)$$
```

Underpinning importance sampling is the concept of *properly weighted samples* from Liu and Chen (1998) and Doucet et al. (2001, pages 227–228). A set of samples and their corresponding weights, denoted by

$$\Big\{\Big(\theta^{(j)},\, w\big(\theta^{(j)}\big)\Big) \,:\, j = 1, \ldots, n\Big\}, \qquad (1.3.15)$$

is called properly weighted with respect to the target distribution, π, if for any square integrable function, $h(\cdot)$,

$$\mathbb{E}\big[h\big(\theta^{(j)}\big)\, w\big(\theta^{(j)}\big)\big] = c\,\mathbb{E}_\pi[h(\Theta)] \qquad (1.3.16)$$

where c is a normalising constant. If we directly sampled from π then

$$\Big\{\big(\theta^{(j)},\, 1\big) \,:\, j = 1, \ldots, n\Big\} \qquad (1.3.17)$$

would be a properly weighted sample.

This idea is why we can use importance sampling to estimate the desired integrals. Additionally, using this idea, we can translate importance sampling to a method for generating samples from the target distribution by sampling from $\big\{\theta^{(1)}, \ldots, \theta^{(n)}\big\}$ with probability proportional to their weights. This will then produce samples, each with equal weighting, from the distribution we desire (Smith and Gelfand, 1992). For further details regarding importance sampling we refer the reader to Doucet et al. (2001, Chapter 1) and Robert and Casella (2005, Section 3.3).

1.3.5 Conclusions

In this section we have discussed three simulation methods. Although relatively simple, these methods have been applied to many real-world problems. If we consider the field we will be interested in, epidemic modelling, Clancy and O'Neill (2007) used rejection sampling to study outbreaks of influenza. This method is particular useful as the final samples are exact, avoiding the convergence issues encountered with other simulation methods (see Section 1.4 where we introduce MCMC methods). Importance sampling has also been applied within this field, for example Marion et al. (2003) used importance sampling within the context of plant epidemiology.

Although simple, these methods are highly intuitive and (usually) straightforward to implement, resulting in their usage still to this day.

1.4 Markov Chain Monte Carlo Methods

> *"In view of all that we have said in the foregoing sections, the many obstacles we appear to have surmounted, what casts the pall over our victory celebration? It is the curse of dimensionality, a malediction that has plagued the scientist from earliest days."*
>
> – Bellman (1961)

The simulation methods we have considered thus far each have their own strengths and weaknesses (summarised later in Section 1.5). One important drawback to all of the methods described is that they become increasingly difficult to implement as the dimension of the distribution we are interested in increases. Called the *curse of dimensionality* (Bellman (1961)) it renders these methods fairly inflexible and lacking in the generality we will often require. It is this weakness that will be the main advantage of the next subset of simulation methods we shall discuss: *Markov chain Monte Carlo* (MCMC). In this section, as we are now interested in higher dimensional problems, we shall consider target distributions of the form $\pi(\boldsymbol{\theta})$, where $\boldsymbol{\theta} = (\theta_1, \dots, \theta_d)$ for $d \geq 1$.

MCMC methods were first developed by Metropolis et al. (1953), before being later generalised by Hastings (1970). However, it was not until Gelfand and Smith (1990) first highlighted the wide range of problems MCMC methods could be used in that their popularity as a statistical tool began to grow. A substantial amount of research has been conducted into the advancement of MCMC methods; we recommend Robert and Casella (2011) and Brooks et al. (2011) for a review of their development.

The aim of this section is to provide an overview of some of the properties and techniques used when considering MCMC methods. We will begin by providing the motivation behind MCMC methods in Section 1.4.1 before describing the *detailed balance condition* in Section 1.4.2, which we use to check that the MCMC is generating samples from the required distribution. In Section 1.4.3 we construct the *Gibbs sampler*, a special form of MCMC algorithm which makes use of the marginal distributions of each parameter. In Section 1.4.4 we extend this idea: utilising a more generic set of distributions to facilitate sampling from the target distribution, with the *Metropolis-Hastings* algorithm. This includes discussion of the form of MCMC we will use throughout, *random-walk Metropolis*, in Section 1.4.4.2, with further discussion of optimising this in Sections 1.4.4.3–1.4.4.4. This optimisation is primarily achieved using adaptive MCMC schemes, which adaptively choose an efficient proposal distribution. We then discuss hybrid MCMC algorithms, which combine the ideas of Metropolis-Hastings and Gibbs samplers, in Section 1.4.5.

Frequently the data we will be working with will only be partially observed, often resulting in an intractable likelihood. In Section 1.4.6 we discuss utilising MCMC methods in conjunction with *data augmentation* and *hierarchical models*, to overcome the issue of missing data. An extension to this is discussed in Section 1.4.6.1, where we describe constructing efficient MCMC methods using *non-centering* methods. In Section 1.4.7 we discuss another extension to MCMC methods which can propose jumps between spaces of differing dimensions, termed *reversible-jump MCMC* (RJ-MCMC).

1.4.1 The Key Idea

Recall that we have a distribution of interest, π; the underlying idea of MCMC methods is to generate a Markov chain, $\{\boldsymbol{\theta}_n : n \geq 0\}$, which admits a stationary distribution of π. We can then use the values of this converged Markov chain as samples from π. We will be interested in the form of the transition kernel, $K(\boldsymbol{\theta}, A) = P(\boldsymbol{\theta}_{n+1} \in A \,|\, \boldsymbol{\theta}_n = \boldsymbol{\theta})$, for this chain. We should note that throughout we shall discuss these methods in relation to densities, however the ideas also relate to more general measures.

Formally, suppose that π exists on a space $\Omega \subseteq \mathbb{R}^d$ and we can find a π-invariant transition kernel which admits a density K, i.e.

$$\int_A \int_\Omega \pi(\boldsymbol{\theta}^*) K(\boldsymbol{\theta}^*, \boldsymbol{\theta}) \, d\boldsymbol{\theta}^* \, d\boldsymbol{\theta} \;=\; \int_A \pi(\boldsymbol{\theta}) \, d\boldsymbol{\theta} \tag{1.4.1}$$

for all sets A. We say K is preserving the distribution of π. Therefore, if $\boldsymbol{\theta}_s \sim \pi$ then $\boldsymbol{\theta}_t \sim \pi$ for all $t > s$. The question is, do we ever have an s such that $\boldsymbol{\theta}_s \sim \pi$? We do not cover here the conditions under which the Markov chain converges to its stationary distribution, nor the rate of convergence. We instead refer the reader to Tierney (1994) if they wish to consider the theory underpinning the methods we describe.

Due to its construction we may often begin the chain far from the stationary distribution, however, as n increases the chain will get arbitrarily close to it. Therefore we discard the first b iterations as a *burn-in* period, after which point we begin keeping the samples. Not all MCMC algorithms require a burn-in period (see, Brooks et al. (2011, pages 19-20)), but we shall use one throughout. The choice of b will depend on the problem we are working with: unfortunately MCMC methods are notoriously slow to converge, and therefore often a significant burn-in is necessary. We will return to this idea later.

1.4.2 Detailed Balance Condition

Determining the stationary distribution of a Markov chain is simplified by the *detailed balance condition.* We begin by assuming that we have a Markov chain with transition kernel, K. The Markov chain is called *reversible* if there exists some function, f, such that

$$K(\boldsymbol{\theta}^*, \boldsymbol{\theta}) f(\boldsymbol{\theta}^*) = K(\boldsymbol{\theta}, \boldsymbol{\theta}^*) f(\boldsymbol{\theta}), \tag{1.4.2}$$

for all $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^*$. This is called the detailed balance condition. From this condition we can see that f will be the stationary distribution of this Markov chain:

$$\int_\Omega K(\boldsymbol{\theta}^*, \boldsymbol{\theta}) f(\boldsymbol{\theta}^*)\, d\boldsymbol{\theta}^* = \int_\Omega K(\boldsymbol{\theta}, \boldsymbol{\theta}^*) f(\boldsymbol{\theta})\, d\boldsymbol{\theta}^* = f(\boldsymbol{\theta}) \int_\Omega K(\boldsymbol{\theta}, \boldsymbol{\theta}^*)\, d\boldsymbol{\theta}^* = f(\boldsymbol{\theta}), \quad (1.4.3)$$

as, by construction, the transition kernel integrates over $\boldsymbol{\theta}^*$ to 1. The detailed balance condition is a simple way of checking the stationary distribution of a Markov chain.

1.4.3 Gibbs Sampler

We require a method of constructing Markov chains with the desired stationary distribution, $\pi = \pi(\boldsymbol{\theta}) = \pi(\theta_1, \ldots, \theta_d)$. One possibility is the *Gibbs sampler* (as described by Geman and Geman (1984)), which successively samples from the conditional distributions of the parameters. This is formalised in Algorithm 4 where we describe the (*systematic scan* or *deterministic scan*) *Gibbs sampler.* Under mild regularity conditions the Gibbs sampler will converge to the desired target distribution, see Roberts and Smith (1994).

Algorithm 4: The (Systematic-Scan) Gibbs Sampler

```
1. Start the chain at $\boldsymbol{\theta}^{(0)} = \left(\theta_1^{(0)}, \ldots, \theta_d^{(0)}\right)$.
2. for  $j = 1, 2, \ldots, (n + b)$
       Sample $\theta_1^{(j)}$ from $\pi\left(\theta_1 \mid \theta_2^{(j-1)}, \ldots, \theta_d^{(j-1)}\right)$.
       Sample $\theta_2^{(j)}$ from $\pi\left(\theta_2 \mid \theta_1^{(j)}, \theta_3^{(j-1)}, \ldots, \theta_d^{(j-1)}\right)$.
           ⋮              ⋮
       Sample $\theta_d^{(j)}$ from $\pi\left(\theta_d \mid \theta_1^{(j)}, \theta_2^{(j)}, \ldots, \theta_{d-1}^{(j)}\right)$.
3. Discard samples $\boldsymbol{\theta}^{(0)}, \ldots \boldsymbol{\theta}^{(b)}$ and use the remaining $n$ samples.
```

As mentioned previously in Section 1.2.3 this method is often used in conjunction with conjugate priors, as these ensure the posterior distribution takes a known and tractable form. This is the main restriction of the Gibbs sampler: often the conditionals will not take a 'nice' form from which we can easily sample.

Finally, we note that we have described the systematic scan Gibbs sampler, however, other forms exist. For example the *random scan Gibbs sampler* which updates a component at random, in each iteration.

1.4.3.1 Collapsed Gibbs Sampler

An extension of the Gibbs sampler is described in Liu (1994), who illustrate a method of reducing the space over which the MCMC algorithm must search. For example, suppose that we have three parameters of interest, $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$, and that we are able to integrate out parameter θ_3. We can begin by generating samples for θ_1 and θ_2 using a standard Gibbs MCMC applied to $\pi(\theta_1, \theta_2)$. Once these samples have been collected we can then use them to draw θ_3 directly from $\pi(\theta_3|\theta_1, \theta_2)$. Liu (1994) noted this has two benefits: firstly, it can reduce the computational cost of the MCMC algorithm by sampling θ_3 directly; secondly, it can reduce the autocorrelation between the samples. Collapsing is important within epidemic modelling, where there is considerable need for efficient MCMC algorithms, see, for example, Xiang and Neal (2014).

1.4.4 Metropolis-Hastings Algorithm

To generate a Markov chain with the required stationary distribution we have introduced the Gibbs sampler, this is a special case of the more general *Metropolis-Hastings algorithm* (Brooks et al. (2011, Section 1.12)), where the probability of accepting a proposed sample is 1. The Metropolis-Hastings (MH) method shares many characteristics with the rejection sampling algorithm described previously in Section 1.3.3. Specifically, we once again use an intermediate distribution to propose values and accept or reject these based on how probable they are.

Firstly, assume that we again wish to sample from the distribution, $\pi(\boldsymbol{\theta})$. For this method we define a *proposal distribution*, $g(\boldsymbol{\theta}, \boldsymbol{\theta}^*)$, from which we generate a candidate sample, denoted $\boldsymbol{\theta}^*$, given the previous value, $\boldsymbol{\theta}$. We then either accept or reject this new value as being from the required distribution. This method is displayed in Algorithm 5; under mild conditions (Hastings (1970)) this will construct a Markov chain that converges to the distribution we desire.

Algorithm 5: Metropolis-Hastings Algorithm

```
1. Start the chain at $\boldsymbol{\theta} = \boldsymbol{\theta}^{(0)}$.
2. for $j = 1, \ldots, (n+b)$
    (i). Generate a candidate sample, $\boldsymbol{\theta}^* \sim g(\boldsymbol{\theta}^{(j-1)}, \cdot)$.
    (ii). Calculate the acceptance probability,
        $$\alpha(\boldsymbol{\theta}^{(j-1)}, \boldsymbol{\theta}^*) = \min\left\{1, \frac{\pi(\boldsymbol{\theta}^*)\, g(\boldsymbol{\theta}^*, \boldsymbol{\theta}^{(j-1)})}{\pi(\boldsymbol{\theta}^{(j-1)})\, g(\boldsymbol{\theta}^{(j-1)}, \boldsymbol{\theta}^*)}\right\}.$$
    (iii). Generate $u \sim U(0,1)$.
    (iv). if $u \leq \alpha(\boldsymbol{\theta}^{(j-1)}, \boldsymbol{\theta}^*)$ then
        Set $\boldsymbol{\theta}^{(j)} = \boldsymbol{\theta}^*$.
    (v). else
        Set $\boldsymbol{\theta}^{(j)} = \boldsymbol{\theta}^{(j-1)}$.
3. Discard samples $\boldsymbol{\theta}^{(0)}, \ldots \boldsymbol{\theta}^{(b)}$ and use the remaining $n$ samples.
```

1.4.4.1 Convergence of the Metropolis-Hastings Algorithm

To prove that constructing a Markov chain in this way does produce samples from the target distribution we require the detailed balance condition described in Section 1.4.2. If we can prove that the Metropolis-Hastings algorithm satisfies detailed balance with $f = \pi(\boldsymbol{\theta})$ then, upon convergence, it will generate samples from the required distribution.

We are interested in proving that the transition kernel, K, satisfies the condition $K(\boldsymbol{\theta}, \boldsymbol{\theta}^*)\pi(\boldsymbol{\theta}) = K(\boldsymbol{\theta}^*, \boldsymbol{\theta})\pi(\boldsymbol{\theta}^*)$. Using the MH algorithm we see that

$$K(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = \delta_{\boldsymbol{\theta}}(\boldsymbol{\theta}^*)\left(1 - \int \alpha(\boldsymbol{\theta}, \boldsymbol{\theta}')g(\boldsymbol{\theta}, \boldsymbol{\theta}')\, d\boldsymbol{\theta}'\right) + \alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*)g(\boldsymbol{\theta}, \boldsymbol{\theta}^*), \tag{1.4.4}$$

where $\delta_{\boldsymbol{\theta}}$ is the Dirac delta function with a mass of one at $\boldsymbol{\theta}$. To show that the Metropolis-Hastings transition kernel satisfies the detailed balance condition we consider two cases.

Firstly if $\boldsymbol{\theta} \neq \boldsymbol{\theta}^*$, then

$$\begin{aligned}
K(\boldsymbol{\theta}, \boldsymbol{\theta}^*)\, \pi(\boldsymbol{\theta}) &= \alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*)\, g(\boldsymbol{\theta}, \boldsymbol{\theta}^*)\, \pi(\boldsymbol{\theta}) \\
&= \min\left\{1, \frac{\pi(\boldsymbol{\theta}^*)\, g(\boldsymbol{\theta}^*, \boldsymbol{\theta})}{\pi(\boldsymbol{\theta})\, g(\boldsymbol{\theta}, \boldsymbol{\theta}^*)}\right\} \pi(\boldsymbol{\theta})\, g(\boldsymbol{\theta}, \boldsymbol{\theta}^*) \\
&= \min\left\{\pi(\boldsymbol{\theta})\, g(\boldsymbol{\theta}, \boldsymbol{\theta}^*), \pi(\boldsymbol{\theta}^*)\, g(\boldsymbol{\theta}^*, \boldsymbol{\theta})\right\} \\
&= \min\left\{\frac{\pi(\boldsymbol{\theta})\, g(\boldsymbol{\theta}, \boldsymbol{\theta}^*)}{\pi(\boldsymbol{\theta}^*)\, g(\boldsymbol{\theta}^*, \boldsymbol{\theta})}, 1\right\} \pi(\boldsymbol{\theta}^*)\, g(\boldsymbol{\theta}^*, \boldsymbol{\theta}) \\
&= \alpha(\boldsymbol{\theta}^*, \boldsymbol{\theta})\, g(\boldsymbol{\theta}^*, \boldsymbol{\theta})\, \pi(\boldsymbol{\theta}^*) \\
&= K(\boldsymbol{\theta}^*, \boldsymbol{\theta})\, \pi(\boldsymbol{\theta}^*). \qquad (1.4.5)
\end{aligned}$$

Thus we see detailed balance has been satisfied. In the case where $\boldsymbol{\theta} = \boldsymbol{\theta}^*$ it is trivial to show that the detailed balance condition has been met. Therefore, under some mild conditions on the proposal distribution (Tierney (1994)), we find that, once converged, the Metropolis-Hastings algorithm will generate samples from the target distribution (see Tierney (1994) and Robert and Casella (2005)).

We can note that the Markov chain will suffer from a high correlation between samples. One way to reduce this is to *thin* the output so that only every k^{th} value is kept. Thinning is predominantly justified for computational reasons, such as memory or time constraints, therefore in many cases it will not be required.

An important property of the Metropolis-Hastings algorithm is the *acceptance rate* we achieve. This is the proportion of proposed samples which are accepted (we may also refer to this as a percentage). If this value is very high then we are possibly proposing steps which are too small, therefore we will converge slowly to the target distribution. In contrast, if the acceptance rate is very low then we are likely proposing jumps that are too large and thus rarely move about the space. Balancing these two properties is key to the success of the Metropolis-Hastings algorithm.

1.4.4.2 Random-Walk Metropolis

The Metropolis-Hastings algorithm's popularity is in part due to the relative freedom we have when choosing the proposal distribution. A widely used choice is to center the proposals on the current value, we call this subset of algorithms the *(symmetric) random-walk Metropolis* (RWM) (Tierney (1994)). If we are in iteration j of the MCMC, with current value $\boldsymbol{\theta}^{(j)}$, then the RWM algorithm generates a new sample using a proposal of the form

$$\boldsymbol{\theta}^{*} \,=\, \boldsymbol{\theta}^{(j)} \,+\, \boldsymbol{\epsilon}, \tag{1.4.6}$$

where in each iteration $\boldsymbol{\epsilon}$ is from some (symmetric) distribution, which is independent of $\boldsymbol{\theta}^{(j)}$.

A common choice, which we shall use throughout, is to choose a Gaussian proposal, such that

$$\boldsymbol{\theta}^{*} \,=\, \boldsymbol{\theta}^{(j)} \,+\, N(\mathbf{0},\, \boldsymbol{M}_d), \tag{1.4.7}$$

which we refer to as the *Gaussian RWM.* The choice of matrix, $\boldsymbol{M}_d$, will determine the acceptance rate we achieve, as well as how well we explore the sample space. The selection of a matrix that balances these two aims is often a non-trivial task. We shall discuss the optimal choice for $\boldsymbol{M}_d$ in Section 1.4.4.3 and describe how to achieve this in Section 1.4.4.4.

1.4.4.3 Optimal Acceptance Rate

Before we can answer the question of how to generate an optimal Gaussian RWM algorithm we need to define what we mean by the term optimal. We will be interested in what the optimal acceptance rate is: we expect this is the value that balances the rate of convergence with the rate at which we explore the sample space. It seems reasonable to expect that some optimal value for this exists. In this section we provide only the key results, with little explanation or background. This is a pragmatic approach as the topic of optimal MCMC algorithms could fill many books. We would encourage those who wish to gain a greater understanding of this topic, to consult the papers which we reference. Additionally we suggest Sherlock (2006) or Brooks et al. (2011, Chapter 4) as an in-depth and clear explanation of the key results.

Some of the first major optimality results for Metropolis-Hastings algorithms can be found in Roberts et al. (1997) and Roberts and Rosenthal (2001), where it is shown that, under certain conditions, as the dimension of state space tends to infinity, the optimal acceptance rate is 0.234. The optimality here refers to the efficiency of the MCMC chain. In general, as our focus is not on optimizing MCMC algorithms, we will not be too concerned with the acceptance rate our MCMC achieves, as long as it is not too high or too low. This is based on the work in Roberts and Rosenthal (2001) which found that "*for RWM on smooth densities, any acceptance rate between 0.1 and 0.4 ought to perform close to optimal.*". Although this is found under specific conditions, such as taking the dimension to infinity ($d \to \infty$), we shall use it as a good 'rule-of-thumb'. For example, Roberts and Rosenthal (2001) found that even with just five dimensions ($d = 5$) the optimal acceptance rate is close enough to 0.234 to make little difference in practice.

1.4.4.4 Adaptive MCMC

Now that we have an optimality criterion we can return to the question of how exactly we obtain this acceptance rate. Fortunately, there has been significant work performed in the optimizing of MCMC algorithms. In general for high-dimensional target distributions a good choice of proposal distribution is $N(\boldsymbol{\theta},\, (2.38)^2\boldsymbol{\Sigma}/d)$ (see Roberts and Rosenthal (2001, 2009)), where $\boldsymbol{\theta}$ is the current position of the chain and $\boldsymbol{\Sigma}$ is the covariance matrix of the target distribution. The factor $(2.38)^2/d$ ensures the chain produces the optimal acceptance rate, 0.234, as determined by Roberts and Rosenthal (2001).

Often we will not know $\boldsymbol{\Sigma}$, therefore another reasonable proposal distribution would be $N(\boldsymbol{\theta},\, (2.38)^2\hat{\boldsymbol{\Sigma}}/d)$, where $\hat{\boldsymbol{\Sigma}}$ is some estimation of the true covariance matrix. However, often we will have little information about the form of the underlying distribution of the parameters, therefore the estimation of the covariance matrix may be poor. One option is to use an adaptive scheme that aims to learn about the form of the distribution as we run the MCMC.

The first major advancement in easily applicable adaptive MCMC algorithms was described by Haario et al. (2001). Here they defined an adaptive algorithm based on the random-walk Metropolis centred on the current state, with covariance matrix determined using all previous states visited so far, denoted $\boldsymbol{\Sigma}_j$ in iteration j. This takes the form of the proposal

$$g_j(\boldsymbol{\theta},\, \cdot) = \begin{cases} N(\boldsymbol{\theta},\, \boldsymbol{\Sigma}_0) & \text{if } j \le t_0 \\ N(\boldsymbol{\theta},\, s_d\,\boldsymbol{\Sigma}_j) + N(\boldsymbol{\theta},\, s_d\,\epsilon\,\boldsymbol{I}_d) & \text{if } j > t_0 \end{cases}, \tag{1.4.8}$$

where $\mathbf{\Sigma}_0$ is an initial guess at the covariance matrix, $\boldsymbol{I}_d$ is the d-dimensional identity matrix, $\epsilon > 0$ is a constant that can be chosen to be very small and s_d is a scaling parameter dependent on the dimension, d. Using the results mentioned previously in Roberts et al. (1997) the choice of $s_d = (2.4)^2/d$ is found to be optimal. Some work is required to prove that this adaptive scheme satisfies the convergence conditions required of MCMC algorithms; for the interested reader we refer them to Haario et al. (2001). A variant on this algorithm is displayed in Roberts and Rosenthal (2009) which uses the proposal distribution

$$g_j(\boldsymbol{\theta},\cdot) = \begin{cases} N(\boldsymbol{\theta},\,(0.1)^2\boldsymbol{I}_d/d) & \text{if } j \leq 2d \\ \\ (1-\beta)N(\boldsymbol{\theta}, s_d\mathbf{\Sigma}_j) + \beta N(\boldsymbol{\theta},\,(0.1)^2\boldsymbol{I}_d/d) & \text{if } j > 2d \end{cases}, \tag{1.4.9}$$

where $s_d = (2.34)^2/d$ and β is a small (positive) constant.

Throughout we will use the ideas of these algorithms to adaptively tune our MCMC algorithms. However, we will avoid the need to prove the algorithms suitability by only using these adaptive schemes between iterations (b_1, b_2) where $b_2 < b$ i.e. this all occurs within the burn-in period (up to iteration b). Thus we will use the adaptive scheme,

$$g_j(\boldsymbol{\theta},\cdot) = \begin{cases} N(\boldsymbol{\theta},\,\mathbf{\Sigma}_0) & \text{if } j \leq b_1 \\ \\ (1-\beta)N(\boldsymbol{\theta},\,s_d\mathbf{\Sigma}_j) + \beta N(\boldsymbol{\theta},\,\boldsymbol{M}_j) & \text{if } b_1 < j \leq b_2 \\ \\ (1-\beta)N(\boldsymbol{\theta},\,s_d\mathbf{\Sigma}_{b_2}) + \beta N(\boldsymbol{\theta},\,\boldsymbol{M}_{b_2}) & \text{if } j > b_2 \end{cases} \tag{1.4.10}$$

where $\boldsymbol{M}_j$ is a matrix, dependent on the dimension of the problem and s_d is as previously stated. Throughout we will choose $\boldsymbol{M}_j = A\boldsymbol{V}_j$, where $\boldsymbol{V}_j$ is a diagonal matrix containing the empirical variances of each parameter, estimated using the values sampled up to iteration j, and A is a constant. Additionally, matching Roberts and Rosenthal (2009), we set $\beta = 0.05$.

1.4.5 Hybrid MCMC

The Metropolis-Hastings algorithm has the advantage over the Gibbs sampler that we do not require any knowledge of the conditional distributions of the parameters. However, it does require the selection of a proposal distribution and a poor choice can result in a chain that converges slowly, or does not explore the entire sample space. A hybrid of both Metropolis-Hastings and Gibbs samplers is an alternative choice, this method is sometimes called a *Metropolis-within-Gibbs* algorithm or *component-wise* MCMC.

One of the simplest ways to construct this form of algorithm is to update each parameter individually; often referred to as a *single-site update* (Brooks et al. (2011)). However, this can be slow if we have parameters that are highly correlated with each other. Similarly we could instead update the parameters in 'blocks'. For example, if we have parameters of interest, $\boldsymbol{\theta}$, which can be split into k, not necessarily equally sized, blocks e.g. $\boldsymbol{\theta} = (\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_k)$, then we can update each of the k blocks separately within each iteration. This can be useful for ensuring the MCMC mixes well and fully explores the sample space.

Throughout we will be using a hybrid model, which uses a mix of different chains, choosing proposal steps that are best suited to our problem.

1.4.6 Hierarchical Models and Data Augmentation

The posterior distribution will inform us about the nature of the parameters we are interested in. Commonly one would generate samples from the posterior distribution and then produce summary statistics or density estimates to learn about the parameters of interest. However, to accomplish this many algorithms rely on the likelihood being analytically and numerically tractable. In many situations this will not be the case, one example we will be looking at is the case of *hierarchical models.*

The hierarchical models we will be considering will involve three components: an observed process, $\boldsymbol{X}$, dependent on an unobserved process, $\boldsymbol{Y}$, dependent on a set of underlying parameters, $\boldsymbol{\theta}$. Under this construction we have independence between $\boldsymbol{X}$ and $\boldsymbol{\theta}$, dependent on $\boldsymbol{Y}$ ($\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{\theta} \mid \boldsymbol{Y}$). This is often called the *centered parametrisation* (Papaspiliopoulos (2003), Neal and Roberts (2005)). We are thus now interested in learning about the posterior distribution of the parameters given the observed data, denoted by $\pi(\boldsymbol{\theta} \mid \boldsymbol{x})$.

Due to the missing information ($\boldsymbol{Y}$) the likelihood may not take a form which we can easily evaluate. As such, we will often employ the technique of *data augmentation.* This method is regularly used within missing value problems and is described in the context of calculating the posterior distribution by Tanner and Wong (1987) as *"augmenting* *the observed data so as to make it more easy to analyze".* The motivation behind data augmentation is the observation that if we have an the observed realisation of $\boldsymbol{X}$, denoted by $\boldsymbol{x}$, and denote the realisation of the unobserved process by $\boldsymbol{y}$, then

$$\pi(\boldsymbol{\theta} \,|\, \boldsymbol{x}) \;=\; \int_{\mathcal{Y}} \pi(\boldsymbol{\theta} \,|\, \boldsymbol{y},\, \boldsymbol{x})\, \pi(\boldsymbol{y} \,|\, \boldsymbol{x})\, d\boldsymbol{y}, \tag{1.4.11}$$

where $\mathcal{Y}$ is the space on which the unobserved process lies. If for the unknown process we can generate samples from $\pi(\boldsymbol{y} \,|\, \boldsymbol{x})$ then the average of $\pi(\boldsymbol{\theta} \,|\, \boldsymbol{y},\, \boldsymbol{x})$ over all of these samples will be approximately $\pi(\boldsymbol{\theta} \,|\, \boldsymbol{x})$ (see, Tanner and Wong (1987)). Thus if we have a likelihood that is intractable due to missing information we can additionally sample over this information, making analysis feasible.

For most algorithms sampling the augmented (missing) data, $\boldsymbol{y}$, can be difficult, especially as this is likely to increase the dimension of the distribution we are considering. This is not a problem for MCMC methods, which work well in conjunction with data augmentation. For example, if we now consider the posterior distribution which includes the augmented data, $\pi(\boldsymbol{\theta},\, \boldsymbol{y} \,|\, \boldsymbol{x})$, an MCMC algorithm can alternate between the steps:

(i). Update $\boldsymbol{\theta} \,|\, \boldsymbol{y},\, \boldsymbol{x}$,

(ii). Update $\boldsymbol{y} \,|\, \boldsymbol{\theta},\, \boldsymbol{x}$.

For obvious reasons this is often referred to as the *two-component Gibbs sampler.* We will explore this method as applied to epidemic modelling in later chapters.

1.4.6.1 Non-Centering

One problem with the centered parameterisation we have used to described the hierarchical model is that, due to the high a priori dependency between the parameters and the missing data, the MCMC can achieve poor mixing. *Non-centering* is a method of re-parametrisation that can greatly improve the mixing within MCMC algorithms. It is shown to be especially effective within the framework of epidemic models, which can suffer from a high dependency between the parameters and the missing data. This method aims to break this dependency by introducing a new variable. Using non-centering methods in conjunction with MCMC algorithms was popularised by Papaspiliopoulos (2003) and, as they can be used in a wide range of situations, they have gained in popularity since.

We continue to consider the hierarchical model $\boldsymbol{\theta} \to \mathbf{Y} \to \mathbf{X}$ and denote the *centered* parametrisation as $(\boldsymbol{\theta}, \mathbf{Y})$. In many cases $\mathbf{Y}$ and $\boldsymbol{\theta}$ will be highly dependent and thus the standard, centered, parametrisation will result in poor convergence of the MCMC. The non-centered method reparameterise the problem by finding a function h such that $\mathbf{Y} = h(\boldsymbol{\theta}, \tilde{\mathbf{Y}})$, where $\boldsymbol{\theta}$ and $\tilde{\mathbf{Y}}$ are a priori independent. The non-centered parametrisation is then defined as $(\boldsymbol{\theta}, \tilde{\mathbf{Y}})$. This breaks the a priori dependence between the parameters and the missing data, thereby hopefully hastening the exploration of the sample space. We note that we may not always be able to define a function h which allows us to break this dependency. Examples of using non-centering methods are described in Papaspiliopoulos (2003), additionally, application of these methods to epidemic modelling can be found in Neal and Roberts (2005), O'Neill (2009) and Jewell et al. (2009).

Which parametrisation to use will depend on the relationship between the missing data and the parameters. Within MCMC algorithms non-centering is generally preferred if there is a strong dependence between the model parameters and the missing data being analysed (Papaspiliopoulos (2003), Neal and Roberts (2005)). However, if we have informative data then the dependency seen in the centered parameterisation can break down and we may not benefit from this technique (Papaspiliopoulos et al., 2007), in this situation a centered parameterisation is preferred. A bridge between the two models is the proposed *partially non-centered* algorithm from Papaspiliopoulos (2003, Chapter 7), which we will not discuss further: Neal and Roberts (2005) provide examples of its application within an epidemic setting.

1.4.7 Reversible-Jump MCMC

So far the MCMC algorithms we have considered can only perform moves from spaces of equal dimension. However, often we shall face scenarios where we require proposals from a space with different dimensions to our current state, for example if we have missing data whose dimension is unknown (see, for example, Gibson and Renshaw (1998)). An extension to the Metropolis-Hastings methods we have discussed was proposed by Green (1995) who constructed a generic framework for generating a Markov chain which can switch between parameter subspaces of variable dimension, whilst satisfying the detailed balance condition. This is known as the *reversible-jump MCMC* (RJ-MCMC). We will only be discussing the principal ideas behind this method, for details of why it satisfies the detailed balance condition we refer the reader to the original paper by Green (1995), Robert and Casella (2005, Section 11.2) or Brooks et al. (2011, Chapter 3).

The condition developed by Green (1995) to ensure the Markov chain is reversible is to introduce the idea of *dimension matching.* Suppose that we have two models we are interested in: model 1, denoted by $\mathcal{M}_1$, with parameters $\phi^{(1)} \in \mathbb{R}^{d_1}$ and model 2, $\mathcal{M}_2$, with parameters $\phi^{(2)} \in \mathbb{R}^{d_2}$. We are interested in constructing a Markov chain that can jump between the subspaces on which the different models lie. To ensure detailed balance is met Green proposed the idea of constructing a bijection between the two spaces, this requires extending the spaces on which $\phi^{(1)}$ and $\phi^{(2)}$ lie. We achieve this by generating $v^{(1)} \sim g_1(\cdot)$ and $v^{(2)} \sim g_2(\cdot)$, such that there exists functions $Y^{(1)}$ and $Y^{(2)}$ satisfying

$$Y^{(1)}(\phi^{(1)},\, v^{(1)}) \,=\, (\phi^{(2)},\, v^{(2)}) \qquad \text{and} \qquad Y^{(2)}(\phi^{(2)},\, v^{(2)}) \,=\, (\phi^{(1)},\, v^{(1)}). \tag{1.4.12}$$

These two functions form a bijection between the two subspaces. The dimension matching requirement we have mentioned simply states that if $|v^{(1)}| = m_1$ and $|v^{(2)}| = m_2$ then $d_1 + m_1 = d_2 + m_2$. Thus we have constructed a mapping from $\mathbb{R}^{d_1+m_1} \to \mathbb{R}^{d_2+m_2}$. With this set-up Green proposed the acceptance probability for moving from $\mathcal{M}_1 \to \mathcal{M}_2$ as

$$\min\left\{1,\, \frac{\pi(\phi^{(2)})\, P(\mathcal{M}_2 \longrightarrow \mathcal{M}_1)\, g_2(v^{(2)})}{\pi(\phi^{(1)})\, P(\mathcal{M}_1 \longrightarrow \mathcal{M}_2)\, g_1(v^{(1)})} \left|\frac{\partial Y^{(1)}(\phi^{(1)},\, v^{(1)})}{\partial(\phi^{(1)},\, v^{(1)})}\right|\right\}, \tag{1.4.13}$$

where

$$J \,=\, \left|\frac{\partial Y^{(1)}(\phi^{(1)},\, v^{(1)})}{\partial(\phi^{(1)},\, v^{(1)})}\right| \tag{1.4.14}$$

is the determinant of the Jacobian for the transformation from model $\mathcal{M}_1$ to $\mathcal{M}_2$ and $P(\mathcal{M}_i \longrightarrow \mathcal{M}_j)$ is the probability of proposing a move from model i to model j.

Often we shall deal with the simplified form such that $m_2 = 0$, in this case the acceptance probability reduces to

$$\min\left\{1,\, \frac{\pi(\phi^{(2)})\, P(\mathcal{M}_2 \longrightarrow \mathcal{M}_1)}{\pi(\phi^{(1)})\, P(\mathcal{M}_1 \longrightarrow \mathcal{M}_2)\, g_1(v^{(1)})} \left|\frac{\partial(\phi^{(2)})}{\partial(\phi^{(1)},\, v^{(1)})}\right|\right\}. \tag{1.4.15}$$

Finally, for moving from $\mathcal{M}_2 \to \mathcal{M}_1$ we simply invert the fractions within the acceptance probability.

With the acceptance probability defined in this way the reversible-jump mechanism satisfies the detailed balance condition and thus it can be used to sample from the target distribution. We shall discuss applying RJ-MCMC in later sections, where we find the dimension matching condition is easily satisfied.

1.4.8 Conclusions

In this section we have introduced Markov chain Monte Carlo methods. They are a highly flexible group of algorithms, that can be applied in a wide range of situations. They work well in conjunction with missing data problems and can be applied when the dimension of the distribution may be unknown. Due to their ability to be used even in difficult problems they have become the gold-standard of Bayesian inference. This has subsequently lead to a considerable amount of research into their theoretical properties, as well as in improving their efficiency (for example, Tierney (1994), Roberts and Rosenthal (2001) and Brooks et al. (2011, Chapter 2)).

MCMC methods are not without their flaws, it can be a non-trivial task selecting an appropriate proposal distribution that fully explores the sample space, whilst converging within a reasonable amount of time. This proposal additionally needs to ensure that the chain does not get ‘stuck’ in a particular region, this is of particular importance when working with multi-modal distributions.

We have only touched upon this group of methods and we refer the reader to Robert and Casella (2005) or Brooks et al. (2011) for an in-depth review of MCMC algorithms. We will further discuss applying MCMC methods to infectious disease problems later in Chapters 3–5. These methods are well used within epidemic modelling which can suffer from large amounts of missing data, the dimension of which is also often unknown.

1.5 Comparison of Simulation Methods

Now that we have discussed a few different simulation methods we provide a summary of their key properties.

Advantages	**Disadvantages**
Inversion Sampling	
• Highly efficient. • Produces i.i.d samples.	• Requires the inverse cdf. • Increasingly difficult to implement in higher dimensions.
Rejection Sampling	
• Efficient if a good proposal is chosen. • Produces i.i.d samples. • Only requires knowledge of the target distribution up to proportionality.	• Inefficient if a poor proposal is chosen. • Increasingly difficult to implement in higher dimensions. • Requires calculation of the scaling factor.
Importance Sampling	
• Highly efficient. • Produces i.i.d samples. • Only requires knowledge of the target distribution up to proportionality. • Samples can be recycled for other target distributions.	• Can perform poorly if an inappropriate proposal is used. • Increasingly difficult to implement in higher dimensions.
Markov Chain Monte Carlo	
• Works well in high dimension problems. • Highly flexible and works well in conjunction with data augmentation. • Only requires knowledge of the target distribution up to proportionality.	• Does not produce i.i.d. samples. • The Markov chain can be slow to converge to the target distribution.

Overall the appropriate choice of simulation method will be highly dependent on the problem we are dealing with. For applications to epidemic modelling we are often working with complex, high-dimensional distributions with large amounts of missing data and as such MCMC methods are the current gold-standard for analysis (see, for example, Gibson and Renshaw (1998), O'Neill and Roberts (1999), Jewell et al. (2009), Deardon et al. (2010), Xiang and Neal (2014) etc.). The use of MCMC methods in the epidemic setting will be discussed further in later chapters.

However, a weakness of all the method discussed thus far is that if we received new data each method would require re-running in order to generate samples from the updated posterior distribution. This is highly inefficient and provides the motivation for the final class of simulation methods we shall discuss in detail: *sequential Monte Carlo* methods.

1.6 Sequential Monte Carlo Methods

In previous sections we have discussed how the advancement of computational power allowed for the emergence of MCMC methods as the gold standard of Bayesian analysis. This class of algorithms enables samples to be generated from distributions that previously would have been difficult, if not impossible, to work with. However, recently there have been advancements in data collection techniques, resulting in richer and larger data sets being collected with increasing efficiency. Although flexible, MCMC methods are required to restart each time we wish to incorporate new data; combined with an often slow rate of convergence this can be problematic for *on-line* (real-time) inference. This opens up the need for faster methods of analysing dynamically changing distributions.

Sequential Monte Carlo (SMC) algorithms are a group of methods for sequentially producing samples from an evolving set of distributions. They are a highly flexible set of algorithms that can be used on a wide range of problems, in many fields of science (see Doucet et al. (2001, Part 4) for a collection of applications). In this section we shall discuss a simple sequential Monte Carlo algorithm, along with some of the key extensions that have been proposed over the years. In the interest of time we shall omit many elements and only provide a broad overview of the methods. For the interested reader a summary of SMC algorithms can be found in Liu and Chen (1998), Doucet et al. (2000), Cappé et al. (2007), Doucet and Johansen (2011) and Wang et al. (2017), whilst a comprehensive collection of the advancements in SMC methods can be found in Doucet et al. (2001). It should be noted that due to their connection to methods used in in fluid mechanics the term *particle filter* is often used interchangeable with sequential Monte Carlo. We shall only use the latter term; however, a remnant of this connection is that we will often refer to the samples generated as *particles*.

SMC methods are a broad class of simulation methods, we shall discuss only a small subset of them in in the following sections. We begin in Section 1.6.1 by constructing an extension to the importance sampling method discussed in Section 1.3.4, with a discussion of using this method in conjunction with parallel computing in Section 1.6.1.1. Extensions to this basic algorithm are provided in Section 1.6.2, which incorporates a resampling step, and Section 1.6.3, which incorporates a movement step. Firstly, however, we define the set-up in which we discuss this collection of methods.

1.6.0.1 Set-Up

We will be considering the case where the posterior distribution is of the form described in Section 1.4.6, $\pi(\boldsymbol{\theta}, \boldsymbol{y} \,|\, \boldsymbol{x})$, as it closely matches the form the target distribution will take when we consider epidemic models in Chapter 3. Additionally, as we are now interested in posterior distributions as a function of time, we use a subscript to indicate the data we have access to. Therefore we are now interested in a sequence of probability distributions, $\{\, \pi(\boldsymbol{\theta}, \boldsymbol{y}_{0:t} \,|\, \boldsymbol{x}_{0:t}) \; : \; t = 1, 2, \ldots\}$, where $\boldsymbol{\theta}$ denotes the parameters the evolving system is dependent on, $\boldsymbol{y}_{0:t}$ is the unobserved process and $\boldsymbol{x}_{0:t}$ is the observed process. If the information at time t is denoted by $\boldsymbol{x}_t$ then we define the data observed up to time t as $\boldsymbol{x}_{0:t} = (\boldsymbol{x}_0, \ldots, \boldsymbol{x}_t)$, and we define $\boldsymbol{y}_{0:t}$ similarly.

We are interested in evaluating $\pi_t = \pi(\boldsymbol{\theta}, \boldsymbol{y}_{0:t} \,|\, \boldsymbol{x}_{0:t})$, at each time step. Using the standard MCMC methods described in Section 1.4 we can evaluate this distribution at each time step, however, for each t we would have to effectively 'restart' the MCMC algorithm, discarding all of the samples generated for the previous time steps. Similarly, to use importance sampling we would need to recompute the weights each time we receive new data. Within systems that are evolving slowly we would expect π_{t+1} to be similar to π_t, as such this appears to be an inefficient method of gaining information about the distribution we are interested in. Sequential Monte Carlo methods aim to utilise the samples generated from π_t, to aid in generating samples from π_{t+1}.

1.6.1 Sequential Importance Sampling

We begin with *sequential importance sampling*, a natural extension to the methods discussed in Section 1.3.4. At time t we are interested in generating n samples from $\pi(\boldsymbol{\theta},\, \boldsymbol{y}_{0:t} \,|\, \boldsymbol{x}_{0:t})$, we can easily achieve this using importance sampling with proposal distribution of the form, $g(\boldsymbol{\theta},\, \boldsymbol{y}_{0:t} \,|\, \boldsymbol{x}_{0:t})$. Thus we can generate a properly weighted sample,

$$\left\{ \big(\boldsymbol{\theta}^{(j)},\, \boldsymbol{y}_{0:t}^{(j)},\, w_t^{(j)}\big) : j = 1, \dots, n \right\}, \quad w_t^{(j)} = w\big(\boldsymbol{\theta}^{(j)},\, \boldsymbol{y}_{0:t}^{(j)}\big) = \frac{\pi\big(\boldsymbol{\theta}^{(j)},\, \boldsymbol{y}_{0:t}^{(j)} \,|\, \boldsymbol{x}_{0:t}\big)}{g\big(\boldsymbol{\theta}^{(j)},\, \boldsymbol{y}_{0:t}^{(j)} \,|\, \boldsymbol{x}_{0:t}\big)}. \tag{1.6.1}$$

Note that each particle, j, contains the proposed parameter values, $\boldsymbol{\theta}^{(j)}$, and a sample for the unobserved data, $\boldsymbol{y}_{0:t}^{(j)}$.

Next, suppose at time $t+1$ we observe new data, $\boldsymbol{x}_{t+1}$, and are therefore now interested in the 'up-to-date' posterior distribution, $\pi(\boldsymbol{\theta},\, \boldsymbol{y}_{0:t+1} \,|\, \boldsymbol{x}_{0:t+1})$. We can write this new distribution as

$$\begin{aligned} \pi(\boldsymbol{\theta}, \boldsymbol{y}_{0:t+1} \,|\, \boldsymbol{x}_{0:t+1}) &= \pi(\boldsymbol{y}_{t+1} \,|\, \boldsymbol{\theta}, \boldsymbol{y}_{0:t},\, \boldsymbol{x}_{0:t+1})\, \pi(\boldsymbol{\theta},\, \boldsymbol{y}_{0:t} \,|\, \boldsymbol{x}_{0:t+1}) \\ &= \pi(\boldsymbol{y}_{t+1} \,|\, \boldsymbol{\theta},\, \boldsymbol{y}_{0:t},\, \boldsymbol{x}_{0:t+1})\, \pi(\boldsymbol{\theta},\, \boldsymbol{y}_{0:t} \,|\, \boldsymbol{x}_{0:t}). \end{aligned} \tag{1.6.2}$$

Therefore rather than restarting we can generate

$$\boldsymbol{y}_{t+1}^{(j)} \sim g\big(\,\cdot\, \,|\, \boldsymbol{\theta}^{(j)},\, \boldsymbol{y}_{0:t}^{(j)},\, \boldsymbol{x}_{0:t+1}\big) \quad \text{and let} \quad \boldsymbol{y}_{0:t+1}^{(j)} = \big(\boldsymbol{y}_{0:t}^{(j)},\, \boldsymbol{y}_{t+1}^{(j)}\big), \tag{1.6.3}$$

for each particle $j = 1, \dots, n$, where g is some proposal distribution. We will refer to this as the '*augmentation*' step of the algorithm. Then the (unnormalised) weight of particle j, at time $t+1$, is

$$w_{t+1}^{(j)} = \frac{\pi\big(\boldsymbol{\theta}^{(j)},\, \boldsymbol{y}_{0:t+1}^{(j)} \,|\, \boldsymbol{x}_{0:t+1}\big)}{g\big(\boldsymbol{\theta}^{(j)},\, \boldsymbol{y}_{0:t+1}^{(j)} \,|\, \boldsymbol{x}_{0:t+1}\big)} = w_t^{(j)} \times \frac{\pi\big(\boldsymbol{y}_{t+1}^{(j)} \,|\, \boldsymbol{\theta}^{(j)},\, \boldsymbol{y}_{0:t}^{(j)},\, \boldsymbol{x}_{0:t+1}\big)}{g\big(\boldsymbol{y}_{t+1}^{(j)} \,|\, \boldsymbol{\theta}^{(j)},\, \boldsymbol{y}_{0:t}^{(j)},\, \boldsymbol{x}_{0:t+1}\big)}. \tag{1.6.4}$$

We can repeatedly apply this idea to sequentially analyse the distributions, as we obtain new data. This is formalised in Algorithm 6 and illustrated in Figure 1.5. At the end of each iteration we will have a properly weighted sample from the desired distribution.

```
Algorithm 6: Sequential Importance Sampling

1. At $t = 0$ generate $n$ samples from $\pi(\boldsymbol{\theta},\, \boldsymbol{y}_0 \,|\, \boldsymbol{x}_0)$, with $w_0^{(j)} = \frac{1}{n}$ for $j = 1, \dots, n$.
2. for $t = 1, 2, \dots$
    (i). for $j = 1, \dots, n$
        (a). Generate $\boldsymbol{y}_t^{(j)} \sim g\big(\,\cdot\,|\,\boldsymbol{\theta}^{(j)},\, \boldsymbol{y}_{0:t-1}^{(j)},\, \boldsymbol{x}_{0:t}\big)$.
        (b). Set $\boldsymbol{y}_{0:t}^{(j)} = \big(\boldsymbol{y}_{0:t-1}^{(j)},\, \boldsymbol{y}_t^{(j)}\big)$.
        (c). Calculate
             $$W_t^{(j)} = \frac{\pi\big(\boldsymbol{y}_t^{(j)}\,|\,\boldsymbol{\theta}^{(j)},\, \boldsymbol{y}_{0:t-1}^{(j)},\, \boldsymbol{x}_{0:t}\big)}{g\big(\boldsymbol{y}_t^{(j)}\,|\,\boldsymbol{\theta}^{(j)},\, \boldsymbol{y}_{0:t-1}^{(j)},\, \boldsymbol{x}_{0:t}\big)}.$$
        (d). Let $w_t^{(j)} = W_t^{(j)}\, w_{t-1}^{(j)}$.
```

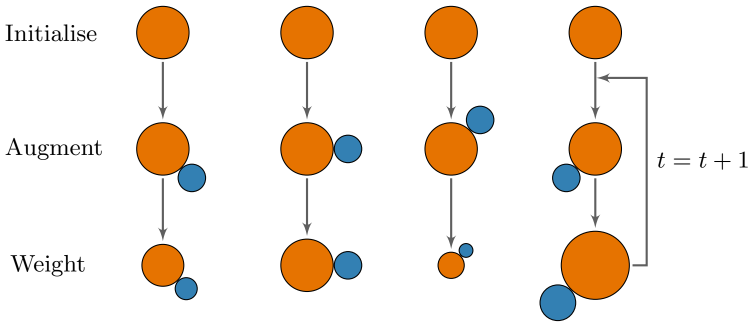


Figure 1.5: An illustration of the sequential importance sampling algorithm. The orange circles represent the (initial) particles and the blue circles represent the new information sampled during the augmentation step. The size of the particles represents the relative contribution of each particle, dependent on their weight. We can see that there is no interaction between the different particles.

We have described simple sequential importance sampling algorithm. Greater detail, and specific applications, can be found in Doucet et al. (2001). To prove the convergence of sequential Monte Carlo methods to the target distribution involves topics beyond the scope of this work. For those interested, we refer the reader to Doucet et al. (2001, Chapter 2) which handles the theoretical side of sequential Monte Carlo methods, with their required convergence results.

1.6.1.1 Parallel Computations

A key advantage of sequential methods is that the majority of the computations are performed on each particle independently. This means that we can partially run the algorithm in parallel. We describe a calculation or algorithm as *embarrassingly parallel* if it can be easily separated into independent, parallel tasks with no communication required between the tasks. As we can observed in Figure 1.5, each iteration of this algorithm is embarrassingly parallel as the particles require no communication with each other.

Let $T^{(j)}_{Augment}$ denote the time it takes to augment particle j and $T^{(j)}_{Weight}$ the time it takes to calculate the weight of particle j. Then, the total time for a single iteration of the sequential importance sampling algorithm performed in serial is

$$\sum_{j=1}^{n} \left(T^{(j)}_{Augment} + T^{(j)}_{Weight} \right). \tag{1.6.5}$$

If instead we perform the weight and the augmentation step fully in parallel then the time to run is

$$\max_{j=1,\ldots,n} \left(T^{(j)}_{Augment} + T^{(j)}_{Weight} \right) \quad \text{or} \quad \max_{j=1,\ldots,n} \left(T^{(j)}_{Augment} \right) + \max_{j=1,\ldots,n} \left(T^{(j)}_{Weight} \right), \tag{1.6.6}$$

dependent on if we perform the steps together or return the output between the augmentation and weighting steps. We can therefore see that by running in parallel we can significantly improve the efficiency of the algorithm.

The number of parallel jobs we can run will be depend on the resources we have access to. Additionally, care needs to be taken if the cost of splitting the jobs up is greater than the speed increase gained. In the examples we consider, however, this will be negligible when compared to the reduction in computation time.

There are many other methods of utilizing parallelization to improve computation time when using simulation methods. For example, Jewell et al. (2009) utilized shared memory architectures when evaluating the summations within their likelihood. This division of labour can be successful in increasing the speed of inference, as evaluation of the likelihood is often the most time intensive step. This is further discussed in Chapter 2.

1.6.1.2 Particle Degeneracy

The continual re-weighting without the introduction of any new particles means that this method will fail for some $t \geq 0$. Once normalised, eventually we will be left with a single unique particle holding all of the weight, with the other $n-1$ particles having little contribution. This is highly inefficient and we will spend a significant amount of time augmenting particles that have a close-to-zero weight and thus will contribute very little to our final estimates. We will refer to this as *particle degeneracy.* It is due to this degeneracy that many extensions to the sequential importance sampling algorithm have been proposed, we shall discuss some of these in the following sections.

One simple solution is to begin the sequential algorithm when we have already observed some of the data (rather than at $t = 0$), as proposed in Liu and Chen (1995). This can reduce the number of particles needed as we have access to a greater number of ‘good particles’ (large weight). However this can only reduce the problem, not eliminate it altogether.

1.6.2 Sequential Importance Resampling

The addition of a resampling step to the standard sequential importance sampling algorithm was proposed by Gordon et al. (1993) as a method of eliminating any particles that are contributing little due to their small weight. Since then it has been incorporated into most sequential Monte Carlo algorithms, see for example, Liu and Chen (1995), Berzuini et al. (1997) and Li et al. (2015).

This additional step resamples the particles with probability proportional to their weights. This is said to ‘rejuvenate’ the particles and allow for superior inference in later time steps. Once resampling has been performed all of the particles will have an equal weight. The addition of a resampling step is formally defined in Algorithm 7, with illustration in Figure 1.6.

Algorithm 7: Sequential Importance Resampling

```
1. At $t = 0$ generate $n$ samples from $\pi(\boldsymbol{\theta},\, \boldsymbol{y}_0 \,|\, \boldsymbol{x}_0)$, with $w_0^{(j)} = \frac{1}{n}$ for $j = 1, \dots, n$.
2. for $t = 1, 2, \dots$
    (i). for $j = 1, \dots, n$
        (a). Generate $\boldsymbol{y}_t^{(j)} \sim g(\,\cdot\,|\,\boldsymbol{\theta}^{(j)},\, \boldsymbol{y}_{0:t-1}^{(j)},\, \boldsymbol{x}_{0:t})$.
        (b). Set $\boldsymbol{y}_{0:t}^{(j)} = (\boldsymbol{y}_{0:t-1}^{(j)},\, \boldsymbol{y}_t^{(j)})$.
        (c). Calculate
             $$W_t^{(j)} = \frac{\pi\left(\boldsymbol{y}_t^{(j)} \,|\, \boldsymbol{\theta}^{(j)},\, \boldsymbol{y}_{0:t-1}^{(j)},\, \boldsymbol{x}_{0:t}\right)}{g\left(\boldsymbol{y}_t^{(j)} \,|\, \boldsymbol{\theta}^{(j)},\, \boldsymbol{y}_{0:t-1}^{(j)},\, \boldsymbol{x}_{0:t}\right)}.$$
        (d). Let $w_t^{(j)} = W_t^{(j)} w_{t-1}^{(j)}$.
    (ii). Resample $n$ particles from those in $\left\{(\boldsymbol{\theta}^{(j)}, \boldsymbol{y}_{0:t}^{(j)}) : j = 1, 2, \dots, n\right\}$ with
          probability proportional to their weight.
    (iii). for $j = 1, \dots, n$
        (a). Set $w_t^{(j)} = \frac{1}{n}$.
```

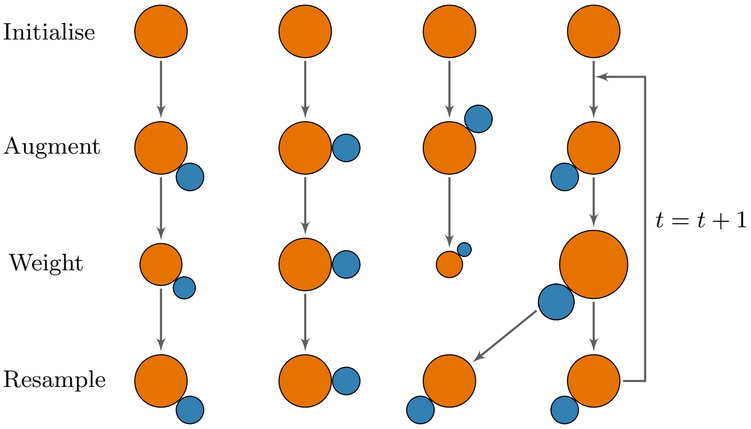


Figure 1.6: An illustration of the sequential importance resampling algorithm.

If the weight of each particle does not depend on the newly sampled part (augmentation step) then it is possible to perform resampling before the augmentation. This allows for a greater diversity in the particles as duplications within the resampling step will generate a different value in the augmentation step. In general it is better to perform resampling prior to any other steps which do not affect the weight (Doucet and Johansen (2011)).

1.6.2.1 The Problem with Resampling

Resampling has both its advantages and disadvantages (Liu and Chen (1995)), one advantage is that we do not waste computational power on particles that will contribute very little to our final estimates. One disadvantage is that we will reduce the number of unique particles, as those with a small weight are unlikely to be resampled. We can illustrate this using a simple example.

Example: Resampling

Suppose that we resample with equal probability at each step

Initial Samples	A	B	C	D	E	F
Resample #1	A	C	F	B	B	C
Resample #2	F	F	A	B	A	A
Resample #3	A	B	A	A	A	F
Resample #4	A	A	A	A	A	A

As we can see it has not taken long for us to be left with each particle sharing a common ancestor. This is clearly problematic, illustrating why resampling should be performed with caution.

1.6.2.2 Resampling Threshold

As we have seen, resampling has both benefits and drawbacks. As such it would be useful to have a criterion as to when we should resample and when we should not.

One common choice is to measure the *effective sample size* (see, for example Kong et al. (1994), Liu and Chen (1995), Doucet et al. (2000) and Doucet and Johansen

(2011)). This cannot be directly computed but can be estimated as

$$\widehat{ESS} = \frac{1}{\sum_{j=1}^{n} (\tilde{w}^{(j)})^2}, \tag{1.6.7}$$

where $\tilde{w}$ represents the normalised weights. We can interpret the effective sample size as the number of perfect samples from the target distribution that are equivalent to the n weighted samples we have generated, where equivalence is in terms of estimator variance. Therefore, a small effective sample size suggests resampling would be a good idea as we are wasting too many resources on particles with a small weight. We can therefore set some threshold, ESS_{th}, which if ESS is below then we resample the particles in that iteration.

1.6.2.3 Methods of Resampling

To resample the particles the most commonly used method is to resample the n particles with probability proportional to their weight, sometimes referred to as *simple random sampling* (see, for example, Liu and Chen (1998)). An alternative to simple random sampling is *residual sampling* (Liu and Chen (1998), Doucet and Johansen (2011), Li et al. (2015)), this can be inserted in place of the simple random sampling method.

Suppose that we have calculated the weight of each particle, $w^{(j)}$ for $j = 1, \ldots, n$. To use residual resampling we begin by rescaling the weights so that they add up to n,

$$\hat{w}^{(j)} = \frac{n w^{(j)}}{\sum_{j=1}^{n} w^{(j)}}. \tag{1.6.8}$$

We keep $n_j = \lfloor \hat{w}^{(j)} \rfloor$ copies of particle j where $\lfloor x \rfloor = \max\{y \in \mathbb{Z} \mid y \leq x\}$ is the floor function. We then define $\hat{n} = \sum_{j=1}^{n} n_j$. For the remaining $n - \hat{n}$ particles we sample them using simple random sampling with the probability of sampling particle j proportional to $\hat{w}^{(j)} - n_j$. This method of sampling the particles will have a lower variance than simple random sampling and can, therefore, be a more stable choice.

Many other resampling methods exist, a collection of which are thoroughly described in Li et al. (2015). Throughout we shall use simple random sampling; however, we reiterate that many alternatives exist, which are not considered in this thesis.

1.6.3 Sequential Importance Resampling and Move

The next extension we consider is the addition of a movement step to the sequential importance resampling algorithm. As mentioned previously the number of unique particles will decrease as t increases. This degeneracy is the main disadvantage of sequential Monte Carlo methods. One solution, as proposed in Berzuini et al. (1997) and extended in Gilks and Berzuini (2001), is to utilize MCMC methods to perform a final movement step.

After resampling we take each particle and perturb it. Therefore, even if we have duplications of particles in the resampling step, we will, at the end of each iteration, have a set of diverse particles. Suppose that we denote by $K_t(\cdot|\boldsymbol{\theta}, \boldsymbol{y}_{0:t}, \boldsymbol{x}_{0:t})$ an invariant Markov transition kernel for the target distribution at time t. If we perturb a sample from π_t using such a kernel then it will remain a sample from π_t (see, Section 1.4). This final step is formalised in Algorithm 8 and illustrated in Figure 1.7.

The additional movement step can be performed using any choice of an appropriate kernel. We can even move the particles multiple times, for example performing m steps of an MCMC algorithm, with the appropriate invariant distribution, on each of the particles independently. As the original particles are from the correct distribution we do not require any burn-in period for the MCMC.

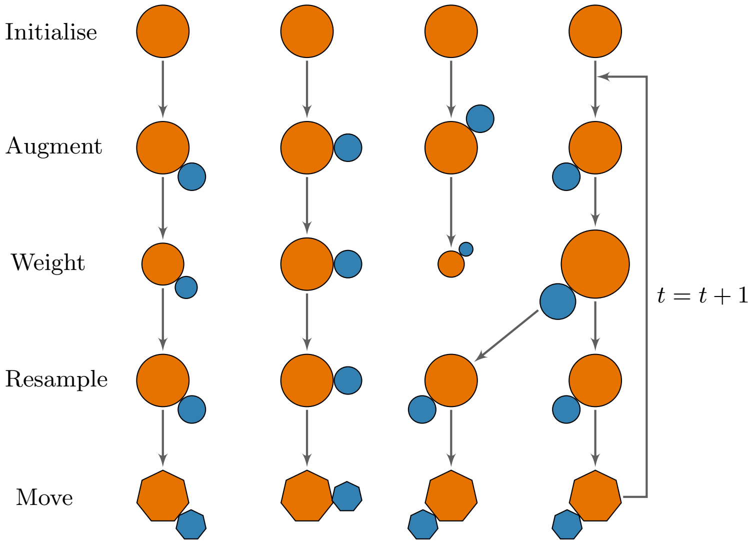


Figure 1.7: An illustration of the sequential importance resampling and move algorithm.

Algorithm 8: Sequential Importance Resampling and Move

```
1. At $t = 0$ generate $n$ samples from $\pi(\boldsymbol{\theta},\, \boldsymbol{y}_0 \,|\, \boldsymbol{x}_0)$, with $w_0^{(j)} = \frac{1}{n}$ for $j = 1, \dots, n$.
2. for $t = 1, 2, \dots$
    (i). for $j = 1, \dots, n$
        (a). Generate $\boldsymbol{y}_t^{(j)} \sim g\big(\,\cdot\,|\,\boldsymbol{\theta}^{(j)},\, \boldsymbol{y}_{0:t-1}^{(j)},\, \boldsymbol{x}_{0:t}\big)$.
        (b). Set $\boldsymbol{y}_{0:t}^{(j)} = \big(\boldsymbol{y}_{0:t-1}^{(j)},\, \boldsymbol{y}_t^{(j)}\big)$.
        (c). Calculate
             $$W_t^{(j)} = \frac{\pi\left(\boldsymbol{y}_t^{(j)} \,|\, \boldsymbol{\theta}^{(j)},\, \boldsymbol{y}_{0:t-1}^{(j)},\, \boldsymbol{x}_{0:t}\right)}{g\left(\boldsymbol{y}_t^{(j)} \,|\, \boldsymbol{\theta}^{(j)},\, \boldsymbol{y}_{0:t-1}^{(j)},\, \boldsymbol{x}_{0:t}\right)}.$$
        (d). Let $w_t^{(j)} = W_t^{(j)} w_{t-1}^{(j)}$.
    (ii). Resample $n$ particles from those in $\left\{\big(\boldsymbol{\theta}^{(j)},\, \boldsymbol{y}_{0:t}^{(j)}\big) : j = 1, 2, \dots, n\right\}$ with
          probability proportional to their weight.
    (iii). for $j = 1, \dots, n$
        (a). Set $w_t^{(j)} = \frac{1}{n}$.
        (b). Generate $(\tilde{\boldsymbol{\theta}}^{(j)},\, \tilde{\boldsymbol{y}}_{0:t}^{(j)}) \sim K_t(\cdot \,|\, \boldsymbol{\theta}^{(j)},\, \boldsymbol{y}_{0:t}^{(j)},\, \boldsymbol{x}_{0:t})$.
        (c). Set $\big(\boldsymbol{\theta}^{(j)},\, \boldsymbol{y}_{0:t}^{(j)}\big) = (\tilde{\boldsymbol{\theta}}^{(j)},\, \tilde{\boldsymbol{y}}_{0:t}^{(j)})$.
```

1.6.4 Conclusions

We have briefly discussed a selection of sequential Monte Carlo methods, focusing on those which adapt the ideas of importance sampling. There are many more extensions to these algorithms, although the key idea remains the same: to repeatedly update our previous analysis to incorporate the new data, avoiding the need to fully restart our analysis from the beginning. The SMC algorithm we will produce in Chapter 3 will be tailored towards use with epidemic data; however, the underlying principals will remain the same as those discussed in this section.

SMC methods have become a popular tool for working with epidemic data. For example, King et al. (2016) provide an extensive `R` package, `pomp`, containing variants of SMC methods, and illustrate the application of these methods in conjunction with epidemic data. Similarly, the work of Birrell et al. (2016) considers adapting the sequential importance reampling and move algorithm for use with epidemic data. This work is particularly interesting as, following the work by Liu and Chen (1995), they make use of the effective sample size to decide if and when to resample. Our application is inherently different to these approaches as we will be focussing on the scenario in which we have large amounts of unknown data and wish to model epidemics at an individual level (see, Section 2.6.3).

1.7 Likelihood-Free Simulation Methods

We conclude this chapter with a brief discussion of a final family of simulation techniques, which do not require computation of the likelihood to generate samples from the desired distribution.

1.7.1 Exact and Approximate Bayesian Computation

The final two methods we shall consider are Exact Bayesian Computation (EBC) and Approximate Bayesian Computation (ABC). These methods avoid computation of the likelihood, which for many distributions can only be computed via data augmentation. We will provide a brief description of these methods, and the interested reader is recommended to consider McKinley et al. (2009), Neal (2012) and Kypraios et al. (2017).

Exact Bayesian Computation (EBC) refers to the fact that we are sampling from the 'exact' target distribution whereas Approximate Bayesian Computation (ABC) samples from approximately the target distribution. EBC and ABC methods generate values for the parameters directly from the prior and then simulate data using these parameters and the specified model. Those sampled parameters are then accepted if they are sufficiently close to the true data. For EBC we require that the simulated and real data agree exactly. For ABC we only require that the distance between some chosen function (for example, $Q(\cdot)$) evaluated using the simulated ($\boldsymbol{x}^*$) and real ($\boldsymbol{x}$) data is below some predefined tolerance (for example, ϵ). We illustrate the EBC and ABC algorithms in Algorithm 9 and 10 respectively, where we can see that the only difference between the two methods is in the acceptance condition.

Algorithm 9: Exact Bayesian Computation (EBC)

```
1. Suppose we desire a sample of size $n$, then let $j = 0$.
2. while $j < n$
       (i). Generate $\boldsymbol{\theta}^*$ from $\pi(\boldsymbol{\theta})$.
       (ii). Sample $\boldsymbol{x}^*$ from the model defined, using $\boldsymbol{\theta}^*$.
       (iii). if $\boldsymbol{x} = \boldsymbol{x}^*$ then
           (a) Set $j = j + 1$.
           (b) Accept $\boldsymbol{\theta}^{(j)} = \boldsymbol{\theta}^*$.
```

Algorithm 10: Approximate Bayesian Computation (ABC)

```
1. Suppose we desire a sample of size $n$, then let $j = 0$.
2. while $j < n$
       (i). Generate $\boldsymbol{\theta}^*$ from $\pi(\boldsymbol{\theta})$.
       (ii). Sample $\boldsymbol{x}^*$ from the model defined, using $\boldsymbol{\theta}^*$.
       (iii). if $d(Q(\boldsymbol{x}), Q(\boldsymbol{x}^*)) \leq \epsilon$ then
           (a) Set $j = j + 1$.
           (b) Accept $\boldsymbol{\theta}^{(j)} = \boldsymbol{\theta}^*$
```

In practice, ABC is often preferred as EBC can result in very low acceptance rates. The acceptance rate will be affected by the tolerance we allow for: too much and our samples will not represent the true posterior distribution, too little and we will have a low acceptance rate. It is these two properties which must be balanced. Additionally if the prior and posterior distributions are significantly different then these methods can perform poorly.

A further extension to this idea is the ABC-MCMC algorithm, which offers a solution in the situation where the prior and posterior distributions are noticeably different. This method combines the ABC algorithm with the MCMC methods we have discussed previously. Another hybrid model incorporates the ABC simulation method within a SMC algorithm. Rather than sampling from a sequence of posterior distributions as in the sequential Monte Carlo method, we instead sample from a sequence of ABC posteriors using a decreasing tolerance. Both of these extensions are discussed within Kypraios et al. (2017) where ABC methods are applied to epidemic data.

Within this chapter we have discussed a collection of simulation methods. We began with discussion of simple simulation techniques such as inverse, rejection and importance sampling. We then discussed in detail MCMC methods and the variants on sequential Monte Carlo algorithms. Finally we concluded with a discussion of EBC and ABC algorithms. In the next chapter we begin the discussion of utilizing these techniques within the context of epidemic modelling.

Chapter 2

Epidemic Modelling

"The real purpose of epidemic theory is not to develop interesting and elegant mathematics, though this may be a delightful incidental byproduct, but is to facilitate the practical prevention or control of actual outbreaks of serious contagious disease."

– Bailey (1967)

Within the previous chapter we illustrated a selection of simulation methods. These techniques are incredibly powerful and can be applied to problems within many fields of research. In this chapter we introduce the specific application we are interested in: infectious disease outbreaks.

Just as Chapter 1 provided an overview of simulation methods, in this chapter we will illustrate the difficulties we will face when attempting to model an event as complicated as an epidemic. We will discuss both present and historic contributions to the field of epidemic modelling, with the overarching aim of better understanding the analysis of infectious disease data.

2.1 Motivation

2.1.1 Why Model Infectious Disease Outbreaks?

Gaining a greater understanding of infectious disease outbreaks is vital to discovering methods of reducing their impact on society. Through epidemic modelling we hope to understand the conditions that cause an outbreak to arise and spread and thus prevent any future epidemics. For example, we may be interested in determining:

- Once infectious, how likely is an individual to transmit the disease to those they have contact with? For example, when modelling the 2001 UK Foot-and-Mouth outbreak, Jewell et al. (2009) used a transmission probability which was a function of the size and composition of the both the infectious and the susceptible farms.

- How does the underlying social structure of the population affect the spread of the disease? For example, Gibson (1997) considered the spread of *citrus tristeza* virus in an orchard where the location of individual trees occured in a lattice structure.

- Once within a host, how does a disease develop? For example, how long is the time between exposure to the disease and the individual becoming infectious. Inference of this period has been incorporated in many infectious disease analyses, for example Groendyke et al. (2011).

- What level of vaccination, if any, is effective in ensuring the outbreak does not propagate? For example, Britton and Becker (2000) investigate the level of vaccination required to stop an epidemic occurring within a community of households, as applied to an influenza outbreak.

- Are the control measures effective in stemming the outbreak? For example, Neal and Roberts (2004) investigated the impact of closing schools during a measles outbreak.

Answering questions such as these is vital to understanding and preventing severe outbreaks.

With a well-formed model we can infer the values of parameters which, by construction, will directly relate to key elements of the outbreak. For example, we could deduce the length of a diseases infectious period, or determine the social structures key in allowing the disease to spread. For this reason the accurate construction of such models is an active area of research.

2.1.2 The Difficulties in Modelling Infectious Disease Outbreaks

Application of standard statistical methods to the epidemic setting is not straightforward. One reason this field of research has remained highly active (see, Section 2.3.1) is the unique nature of infectious disease outbreaks and the models required to study them.

Firstly, and perhaps most importantly, outbreaks cannot be repeatedly studied under controlled settings. As such to model an epidemic we will often rely on partially observed data from a concluded epidemic. Even rich data-sets will usually only contain information about when individuals exhibit specific symptoms, with information such as an individuals exposure time being treated as unknown. Additionally, each epidemic will have its own unique environmental, social and biological factors relating to its spread, which we aim to model. However, this limited data and lack of a 'baseline' with which to compare against means that determining the key characteristics of an outbreak can be problematic.

Also of difficulty is knowing which factors to incorporate: ideally we wish to capture the key aspects of an outbreak, whilst maintaining a model that can be worked with. Additionally, due to limited data, we might be restricted in what we can model. This can occur when conducting analysis on concluded outbreaks, as objectives may arise after the data has been collected.

Finally, of note is the inherent randomness in infectious disease outbreaks. From the movement of individuals to the development of the disease once it enters a host, the models we build must account for a lack of determinism in almost every aspect of an infectious disease outbreak.

2.2 Key Terms

When discussing epidemic modelling there are many definitions to keep track of. To aid in our subsequent discussion we summarise the more commonly used terms here, which the reader can reference throughout.

Individual Status

Each individual within the population, at any time, will be in one of x states, with the number of possible states dependent on the model we are using.

Susceptible A healthy individual who has the potential to be infected.

Exposed An individual who has been infected but is not yet able to transmit the disease to others, this is sometimes also referred to as the *latent* state.

Infectious An individual who has been exposed to the disease and can infect those who are susceptible.

Notified An individual who is known to be infectious but is yet to be removed from the population and thus can still infect others.

Removed An individual who has been infected but who is no longer infectious. This individual cannot be reinfected.

Data Type

Final size (data) (Data which contains information on) the number of individuals who ever become infected.

Complete Data in which we have knowledge of each individuals state, at all times, *cf. partially observed data.*

Partially observed Data in which we only witness part of the epidemic; for example only observing which individuals are infected but not who infected them, or observing when individuals are removed but not when they are infected, *cf. complete data.*

Temporal Data which involves observing when individuals enter a certain state.

Population Type

Closed A population which does not change, with the same individuals remaining throughout the time period considered, *cf. dynamic population.*

Dynamic A population which is allowed to grow or shrink in size: either through births, deaths, immigration or emigration, *cf. closed population.*

Heterogeneous A population of individuals who do not share the same characteristics. This may relate to social factors, for example interacting with those they live with more frequently, or biological, for example particular individuals being highly susceptible to infection, *cf. homogeneous population.*

Homogeneous A population of individuals who share the same characteristics. This will relate to both social and biological factors, *cf. heterogeneous population.*

Population Structure

Household Each individual belongs to a single household, with different rates of contact occurring within and between households.

Network The population has an underlying structure which can be represented by a graph. Individuals are defined as nodes with edges between nodes representing contact between pairs of individuals.

Spatial The probability an individual is infected, or infects others, is dependent on their location.

Model Type

Deterministic A model which states that, at any time point, the number of individuals in each state can be fully determined by knowing the initial conditions and the parameter values of the model. Thus no probabilistic element is incorporated within the model, *cf. stochastic model.*

Stochastic A model which incorporates the inherent randomness of an infectious disease outbreak, *cf. deterministic model.*

General Terminology

Endemic disease When the infection remains constant within a population, at some base level.

Infectious period The time during which an individual is infectious.

Mixing Individuals within a population interact with different groups of people, at different rates. For example, an individual who interacts with the global population and those they work with at different rates will have two levels of mixing.

On-line inference Analysing data in real time, as they are obtained. This is performed whilst the outbreak is still ongoing.

2.3 Choosing an Epidemic Model

In the following sections we will discuss the many advancements within epidemic modelling, as well as the specifics of constructing such models. As stated by Daley and Gani (2001, pages 15–16), we can loosely split the assumptions we will be required to make, when describing an epidemic model, into three parts:

1. The assumptions about the individuals within the populations
For example, is the population closed or dynamic? Do we assume individuals are fully homogeneous (behave identically), fully heterogeneous (each individual behaves uniquely), or somewhere in between?

2. The assumptions about the disease
These could relate to the mechanism of how the disease spreads. For example, most of the models we discuss will assume that contact between individuals is equivalent to infection. We also need to assume how the disease develops; for example, once recovered are individuals granted full immunity, or can they be reinfected?

3. The mathematical assumptions of the model
These assumptions are usually the practical aspects of the model, such as do we model in discrete or continuous time? Should we assume a stochastic or a deterministic model? These are justified by considering what is most appropriate for the epidemic we are considering.

The assumptions made will have a significant impact on any analysis conducted, however, often they will be determined by the capabilities of the algorithms currently available and the quality of the data we have access to. This is illustrated in the next section, where we review the progression of infectious disease modelling.

2.3.1 A Brief History of Modern Epidemic Modelling

In this section we will highlight a selection of the key moments in the evolution of infectious disease modelling. Our aim is only to provide an overview and, as such, there will be contributions overlooked or omitted. This is the nature of a field of research that has a wealth of work attached to it. The primary focus of our work will be on using data augmentation of temporal data in conjunction with simulation methods to conduct inference on an ongoing epidemic, this will show in the bias of which papers we discuss. Therefore, for a comprehensive overview, the reader is recommended to consider the abundance of literature that exists, for example: Bailey (1975), Becker (1989), Daley and Gani (2001), Andersson and Britton (2000) and Diekmann et al. (2012), amongst others.

2.3.1.1 *"A Contribution to the Mathematical Theory of Epidemics"*

The beginnings of modern epidemiology is often attributed to the paper by Kermack and McKendrick (1927). Contained within is what is considered by many to be the first widely accepted illustration of a fully formed epidemic model. The major contribution of this paper is the threshold result for a deterministic epidemic model, since termed "*Kermack and McKendrick's threshold theorem*". This result determines the boundary on which an epidemic will almost certainly occur and was one of the first of many theoretical advancements within epidemic modelling.

2.3.1.2 Introducing Randomness

Some years later the next major progression came from Bartlett (1949) and Bailey (1950) who popularised the use of stochastic models, analogous to the deterministic model proposed by Kermack and McKendrick (1927). These models could now account for some of the inherent randomness of an outbreak, providing a more realistic model of an epidemic.

The connection between an epidemic in a large population and the probability of extinction within a branching process was quickly established. By approximating an epidemic as a birth-death process, with births representing infectious contact, theoretical progress can be made. The first rigorous illustration of utilising branching process approximations was conducted by Whittle (1955), where this relationship was exploited to determine what has since been referred to as "*Whittle's threshold theorem*". Whittle's work determined the analogous stochastic result to the deterministic threshold found by Kermack and McKendrick (1927).

An alternative model was proposed by Lowell J. Reed and Wade Hampton Frost who developed an (unpublished) stochastic model in their class lectures. The key idea behind the named *Reed-Frost model* (see, Section 2.6.1) was to regard infections as occurring in generations. Under certain conditions the number of infections in the next generation followed a binomial distribution, dependent on the number of susceptible and infectious individuals in the previous generation. As their work was unpublished we refer the reader to Abbey (1952) which provides one of the first in-depth considerations of this model.

In the following years the advancement of stochastic models continued, for example Bailey and Thomas (1971) considered stochastic models, utilising maximum likelihood (ML) methods to estimate the rate of infection and removal, with an extension to the use of inter-removal times (the time interval between observations). Highlighted within this paper is what would be a recurrent issue within infectious disease modelling, chiefly that computation of the maximum likelihood is very costly as infection times are rarely observed. This issue was addressed in Becker (1979) which illustrated the use of martingales (see, Becker (1989, Chapter 7.1)), bypassing many of the computational issues encountered when using ML methods. Becker applied this method to independent measles outbreak within households, generating results which matched well with those found in Bailey (1975, page 252) which used standard ML methods and a Reed-Frost model.

2.3.1.3 The Mathematical Modelling of Epidemics

In the following years much progress was made on the theoretical side of epidemic modelling. For example, an interesting theoretical result was proved by Ball (1983a) who (like the earlier work of Whittle (1955)) used the relationship between infections and births in a population to show that, over a finite time interval, the general stochastic epidemic (see Section 2.6.2) converges to a birth-death process as the population size tends to infinity.

Another group of epidemic models, whose theoretical properties were being increasingly explored, were those which incorporated heterogeneities within the population. For example, Ball (1985) stated the result that in both deterministic and stochastic settings assumptions of a homogeneous susceptible population results in a worst-case scenario analysis when compared to allowing varying susceptibility. Also concerned with the theoretical properties of an epidemic, Ball (1986) obtained the distribution of the final size of an epidemic, for any form of infectious period whose moment generating function exists, extending the results to a heterogeneously mixing population.

The majority of research thus far focused on continuous-time models, these allow events (infection, removal etc.) to occur at any point in time (e.g. $t \in \mathbb{R}^+$). These models match how true events are likely to occur and can often produce more mathematically tractable problems. However, there was still an interest in discrete-time models which assume events occur at equally spaced time steps (e.g. $t = 0, 1, \ldots$), matching with how the data is often collected. For example, Longini Jr (1980) discussed an extension to the Reed-Frost model for endemic disease, Malice and Lefevre (1985) discussed discrete-time compartmental models and Rampey Jr et al. (1992) constructed a discrete-time model using incidence data (recording the onset of symptoms), where each individual in the community belongs to a household.

Research into the theoretical side of epidemic modelling has continued to expand, however, this is not something considered in great detail here. Any interested readers are encouraged to refer to Bailey (1975) for a description of the work up until 1975 and Daley and Gani (2001) which provides a comprehensive overview up to the end of the 20th century. Additionally Becker (1989) considers the application of standard statistical methods to infectious disease data.

2.3.1.4 Computational Advancements

In much of the work discussed, the use of ML estimators was still very time consuming. Although the work of Becker (1979) via martingales offered a solution, they could only be used in a limited number of situations. As such the majority of research thus far focused on the mathematical modelling of epidemics, rather than any statistical analysis. However, with the advancement of computing power a solution to the likelihood problem was proposed at the end of the 20th century by Gibson and Renshaw (1998) and by O'Neill and Roberts (1999). They raised the issue of partially observed data and both proposed the use of reversible-jump Markov chain Monte Carlo algorithms (see, Green (1995)). The methods proposed performed augmentation of the unknown infection times, allowing for the computation of likelihoods that were previously intractable. For example, an MCMC algorithm was successfully applied by Gibson (1997), who modelled the spread of *citrus tristeza* virus in an orchard where each individual tree is represented as a vertex in a rectangular lattice.

The use of MCMC methods allowed for a greater level of flexibility to be incorporated when constructing an epidemic model. Previously intractable likelihood functions could now be considered, and thus increasingly detailed inference could be made. O'Neill and Roberts (1999) applied MCMC methods to final size and temporal data using both chain binomial and general stochastic models. This method was then extended in O'Neill et al. (2000), who considered household models. O'Neill and Becker (2001) focused on using MCMC methods to determine parameters relating to the infectious periods of individuals, as well as allowing for random heterogeneity in the susceptibility of individuals. The ability to infer such informative parameters, which are descriptive of key characteristics of an outbreak, was an important and highly useful advancement.

During this period the topic of epidemic modelling was rapidly expanding, with increasing amount of research and routes of modelling to consider. For a review paper summarising this work see Becker and Britton (1999).

2.3.1.5 Random Graphs

With the popularisation of MCMC methods many alternative approaches to modelling a population could be used. For example, Britton and O'Neill (2002) utilised MCMC methods and temporal data to describe the population of individuals as existing on a Bernoulli random graph. Thus, in addition to updating the parameters and infection times, the pathway of infection also required updating. Their work was then extended and generalised by Groendyke et al. (2011). Demiris and O'Neill (2005) considered random graphs with two levels of mixing (local and global), using only final outcome data. The augmented data then took the form of a random graph describing the potential infectious contact individuals had. An alternative augmentation scheme was described in O'Neill (2009), where the number of contacts individuals had and whom they were with was incorporated into the likelihood function. This provided a level of analysis that prior to MCMC methods would have been impossible.

Considering structured populations has a certain level of appeal, as it matches how we might expect a population to interact. The use of data augmentation and MCMC methods mirrored the work performed on determining the theoretical properties of epidemics on such populations. Andersson (1998) considered a discrete-time model, where the heterogeneously mixing population existed on a random graph. By using branching approximations they determined (asymptotically) the final size of the outbreak. Another example of a structured population is handled by Ball et al. (1997) who considered the theoretical properties of a population partitioned into households, existing on a random graph. Rather than using data augmentation Ball et al. (1997) made various assumptions, for example that the number of households is large, which allowed for various theoretical properties of the outbreak to be determined. Similar work on a structured population with two-levels of mixing was performed in Ball and Neal (2008), were the number of neighbours each individual had (degree) was specified.

Thus, there have been two key strands to network modelling. Those that focus on using simulation methods, predominantly MCMC methods, and those that focus on determining the features of an outbreak, such as proportion of the population ultimately infected in Ball and Neal (2008), typically by using limiting behaviour arguments. Overall there has been a considerable amount of work conducted on infectious disease outbreaks occurring on networks, a review of which can be found in Danon et al. (2011).

2.3.1.6 Model Selection and Efficient MCMC

Another advantage of MCMC methods is that they can be easily used to conduct model selection, as demonstrated in Neal and Roberts (2004) and O'Neill and Marks (2005). This can prove highly useful, for example in Neal and Roberts (2004) model selection is used when deciding if classroom, household or spatial effects are important when modelling the Hagelloch measles outbreak. Clancy and O'Neill (2007) also considered the idea of model selection, using calculation of the Bayes factor to decide between competing models. Their analysis is exact as they additionally illustrated how rejection sampling can be a viable alternative to MCMC methods, which avoids their convergence issues. Knock and O'Neill (2014) consider model selection, where the choice is between a homogeneous population or a population with two levels of mixing. In this paper they additionally compared the outputs using both MCMC and path sampling methods.

The use of MCMC methods opened up the possibilities for inference on infectious disease models, however in many ways this created new problems. In particular, MCMC algorithms involving the imputation of a significant amount of missing data can be notoriously slow to converge, which is not ideal for inference that is required quickly. Neal and Roberts (2005) considered improving the efficiency of their MCMC algorithm using non-centering methods, with applications to both the general stochastic epidemic and an epidemic occurring on a random graph. Non-centering methods and efficient MCMC algorithms were additionally the focus of Jewell et al. (2009), who constructed an algorithm capable of on-line inference, and O'Neill (2009), where they are utilised to improve the mixing of an MCMC algorithm.

Once again the development of infectious disease modelling had made considerable jumps and as such review papers such as Isham (2005), O'Neill (2010) and Britton (2010) aimed to describe and capture the progress thus far.

2.3.1.7 Current Work

In more recent years increasingly complex models have been suggested, which aimed to better describe the mechanism of an outbreak. Some work has focused on capturing the behaviour of individuals in a population. For example, Deardon et al. (2010) constructed a model for the 2001 UK Foot-and-Mouth disease outbreak, with the aim of modelling each farm at an individual level. Other work has been concerned with the general social structure of the population, for example Britton et al. (2011) constructed a model which described three levels of mixing (e.g. household, school and global) and compared the results obtained using final size data versus complete data.

Additionally, there has been increasing interest in accurately describing the mechanism of transmission. For example, O'Neill and Wen (2012) considered a stochastic model with non-linear infectious pressure and Neal (2016) incorporated a time of day effect, ensuring individuals could only infect members of a single group at a time e.g. the whole community in the morning but only their household at night. Also of consideration were the assumptions about how individuals are identified, for example Ball et al. (2011) incorporated contact tracing with some probability into their model, this was then extended in Ball et al. (2015) which added delays to this tracing, as well as incorporating a latent period.

New methods of simulation have also being applied, such as the description of an efficient MCMC algorithm by Xiang and Neal (2014), which used parameter reduction methods and adaptive tuning. Also interested in efficient MCMC algorithms, Neal and Xiang (2017) used non-centering methods and collapsing (see, Section 1.4) to construct an efficient MCMC. Additionally there are the methods proposed by McKinley et al. (2009), which avoided calculation of the likelihood altogether. In a similar vein, Neal (2012) also considered avoiding calculation of the likelihood, by constructing a variation on the ABC algorithm that incorporated the ideas of non-centering to increase efficiency. ABC methods as applied to epidemic data have additionally been reviewed in Kypraios et al. (2017).

More recently, many of the extensions to infectious disease modelling have focused on non-parametric methods, allowing for increasing flexibility. This is achieved using Gaussian processes in Xu et al. (2016) and in Kypraios and O'Neill (2018).

2.3.1.8 Conclusions

As we can see the work on infectious disease modelling has been rapid and varied. In the later years within our summary we have focused, in the view of time, on those relating to simulation methods. However, there exists a wealth work into the theoretical side of epidemic modelling, which we have only briefly highlighted here. Additionally there exist many methods which focus on reconstructing the transmission tree of an outbreak ('who infected whom') using genetic data in conjunction with the epidemiological data. An example of this is implemented in the `outbreaker` package in R, as developed by Jombart et al. (2014)). There has also been interesting work performed in the area of malware modelling, which shares significant overlap with the advances in epidemic modelling, see for example del Rey (2015) for a review of the work performed in this field.

Finally, as we stated previously, there are many decisions we have to make when modelling an infectious disease outbreak. These choices are often made for pragmatic reasons, such as the capabilities of the available algorithms or the richness of the data, as well as for reasons specific to our aim; for example, choices about the spatial aspect of an outbreak will be important if our aim is to determine the effectiveness of ring-culling schemes (for example in the Foot-and-Mouth outbreak). Thus there has been an increasing amount of work focused on the assessment of if an epidemic model is appropriate. A review of this topic is provided in Gibson et al. (2018) who outline the boundaries under which one should be critical of any epidemic model formed.

In the following sections we begin to explore the practical construction of an epidemic model, with the aim of developing our own once we have decided which assumptions we will make with respect to those discussed in this chapter.

2.4 Compartmental Framework

All of the epidemic models we shall be considering will have an underlying compartmental framework. This means that we assume individuals within the population are, at any one time, in one of x states. Each state has its own properties with individuals within each compartment sharing some, or all, behaviour.

When individuals are assumed to be fully homogeneous within their compartment we refer to this as a *compartmental model*. This subset of models is only concerned with how many individuals are in each state at a given time, with the movement of individuals between each state treated deterministically or stochastically. Much of the work highlighted throughout this chapter will use an underlying compartmental model, as it will often lead to greater tractability.

There are many possibilities for the choice of compartments; we discuss some of those most commonly used next.

2.4.1 The SIR Model

The compartmental framework we will mostly be concerned with is the SIR model. This is one of the most commonly examined epidemic models see, for example, Bailey and Thomas (1971), Andersson and Britton (2000, Chapter 2), Streftaris and Gibson (2004), Neal and Roberts (2005), Britton (2010) and Xiang and Neal (2014). This model assumes that individuals within a population can be in one of three possible states: susceptible (S), infectious (I) or removed (R). Individuals progress through the three states in the order shown in Figure 2.1.

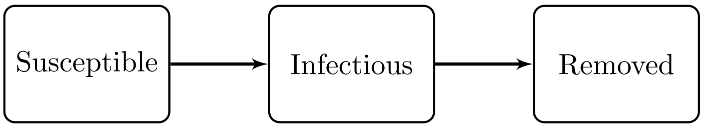


Figure 2.1: The SIR model, often referred to as the 'general' model.

Historically when describing epidemic models as *general* it is with reference to the SIR compartmental model. A simpler version of this, with no removal stage and individuals remaining infectious until the conclusion of the epidemic, is often referred to as a *simple* epidemic model (see, Figure 2.2). General and simple epidemic models can be deterministic or stochastic, and in discrete or continuous time, all combinations of which are discussed in detail in Daley and Gani (2001).

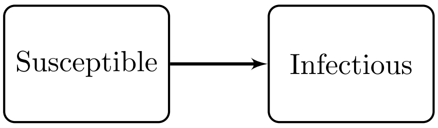


Figure 2.2: The SI model, often referred to as the 'simple' model.

2.4.2 The SEIR Model

We will primarily be focusing on an SIR epidemic throughout this chapter, however, it is by no means the only possible choice. One common extension is an SEIR model (see, Figure 2.3), which incorporates an *exposure* (E) period (also called *latent*) during which time an individual has been infected but cannot yet infect others. Examples of this model are discussed within Gibson and Renshaw (1998), Groendyke et al. (2011) and Britton et al. (2011). An example of when this model can be used is in the recent Ebola outbreak. The Ebola virus has a latent period of 2–21 days, between an individual being infected and becoming infectious (WHO, 2018), suggesting the SEIR model is more appropriate than the SIR model.

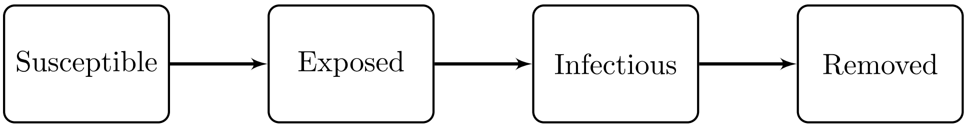


Figure 2.3: The SEIR model.

2.4.3 The SIS Model

Another frequently discussed model is the SIS model (see, Figure 2.4). This substitutes the removal state in the SIR model for another susceptible state, thus once recovered an individual can become infected again. This choice is most commonly used to model the spread of sexually transmitted diseases, which will often not provide immunity once an individual has recovered. This is the simplest model in which we can observe endemic behaviour, as we have a constant supply of individuals who can be infected. Examples of this model can be found in Weiss and Dishon (1971) and Andersson and Britton (2000, Chapter 8.2).

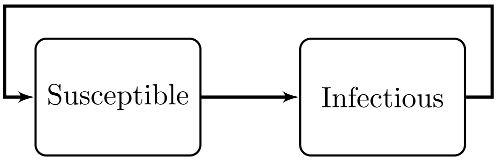


Figure 2.4: The SIS model.

2.4.4 The SINR Model

The final model we consider is the SINR model displayed in Figure 2.5. This model shares many similarities with the SIR model only with an additional *notification* (N) period, during which we are aware of an individual's infectiousness but they are yet to be removed from the population. This is commonly used to describe agricultural epidemics, which often have a delay between the notification that a farm is infected and its removal (quarantining or culling). This model is explored within Jewell et al. (2009) where it is applied to a simulated Avian Influenza outbreak and will be more thoroughly examined in Section 3.6.

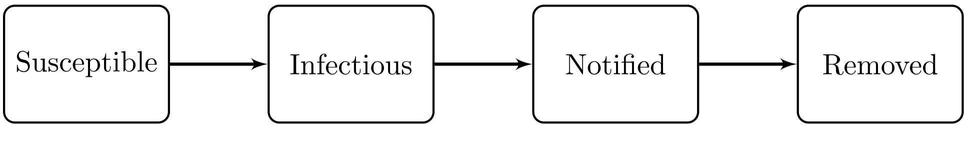


Figure 2.5: The SINR model.

2.5 A Deterministic SIR Model

Historically, one of the first deterministic models was considered by Bernoulli (1700–1782), who aimed to show that vaccination against smallpox reduces the rate of death. Further details of this model can be found in Daley and Gani (2001) where a thorough discussion of Bernoulli's methods is conducted. In modern epidemiology when we speak of deterministic models we refer to the derivation of differential equations which can describe the infection process. Focusing on an SIR model, this relies on the assumption that the number of susceptible, infectious and removed individuals are a function of discrete time, or continuously differentiable functions of continuous time. This means no randomness is incorporated when modelling an outbreak.

In the following section we shall provide a brief outline of a deterministic, continuous-time, SIR epidemic model in a homogeneous population. Note that throughout we refer to a population as '*homogeneous*' if the people are homogeneous and they interact homogeneously (uniform mixing). The construction we shall describe can be found in greater detail in Bailey (1975, Chapter 6) and in Daley and Gani (2001, Chapter 2), as well as in its original form in Kermack and McKendrick (1927).

For this model we assume that we have a closed, homogeneous population of size N_{pop} with individuals being in one of three states: susceptible, infectious or removed (see, the SIR model in Section 2.4.1). A deterministic model allows for the construction of differential equations which describe the rate of movement between each of the three states. We denote the number of susceptible, infectious and removed individuals at time t by $S(t)$, $I(t)$ and $R(t)$ respectively and allow these to be non-integer. We then fully define the deterministic SIR model with the following set of three differential equations:

$$\frac{dS(t)}{dt} = -\beta S(t)I(t), \tag{2.5.1}$$

$$\frac{dI(t)}{dt} = \beta S(t)I(t) - \eta I(t), \tag{2.5.2}$$

$$\frac{dR(t)}{dt} = \eta I(t), \tag{2.5.3}$$

and initial conditions

$$(S(0),\, I(0),\, R(0)) = (N_{pop} - c,\, c,\, 0), \tag{2.5.4}$$

where we have assumed the epidemic begins at time $t = 0$. Here β denotes the (pairwise) infection rate, η is the removal rate and c denotes the number of initially infectious individuals. Thus the rate at which individuals leave state S is equal to the rate at which they enter state I, and the rate at which they leave state I is equal to the rate at which they enter state R. The requirement of a closed population allows for the construction of the equations displayed in (2.5.1) – (2.5.3).

Due to equations (2.5.1) – (2.5.3) we can determine the number of individuals, in each of the three states, at any point in time. Additionally, as we have assumed the population is closed, we know that at each time step, t, $S(t) + I(t) + R(t) = N_{pop}$. Consequently, we only require two of the three differential equations to fully determine the model.

For an epidemic to grow, the number of infectious individuals has to be increasing, therefore we require $dI(t)/dt > 0$ for $t = 0$. If we consider (2.5.2) we can see that this is true when $\eta/\beta < S(0)$. Therefore $\eta/\beta = S(0)$ acts as a threshold for if an epidemic is to occur, this is called *Kermack and McKendrick's threshold theorem.* Its importance is immediately clear as it states that, dependent on if $S(0)$ is smaller or larger than η/β, the outbreak will exhibit very different behaviour.

2.5.0.1 The Epidemic Curve

A key advantage of the deterministic model is that by solving (2.5.1) – (2.5.3) we can determine how many individuals are in each state, at each time step. We begin by noting that

$$\frac{dS(t)}{dR(t)} = -\frac{S(t)}{\rho}, \qquad \text{where } \rho = \frac{\eta}{\beta} \text{ is the } \textit{relative removal rate.} \tag{2.5.5}$$

Solving (2.5.5) and using the initial conditions stated in (2.5.4) we find that

$$S(t) = S(0)e^{-\frac{R(t)}{\rho}} \tag{2.5.6}$$

and therefore

$$I(t) = N_{pop} - R(t) - S(t) = N_{pop} - R(t) - S(0)e^{-\frac{R(t)}{\rho}}. \tag{2.5.7}$$

Substituting this solution back into (2.5.3) yields,

$$\begin{aligned}\frac{dR(t)}{dt} &= \eta\left(N_{pop} - R(t) - S(0)e^{-\frac{R(t)}{\rho}}\right)\\ &= \eta\left(N_{pop} - R(t) - S(0)\left(1 - \frac{R(t)}{\rho} + \frac{R(t)^2}{2\rho^2} + O\left(\frac{R(t)^3}{\rho^3}\right)\right)\right)\\ &\approx \eta\left(N_{pop} - S(0) + \left(\frac{S(0)}{\rho} - 1\right)R(t) - \frac{S(0)}{2\rho^2}R(t)^2\right). \qquad (2.5.8)\end{aligned}$$

Assuming that $R(t)/\rho \ll 1$, this approximation will remain appropriate.

Equation (2.5.8) can be solved using standard methods, we print the results as stated in Bailey (1975, page 83) and equation (30) of Kermack and McKendrick (1927),

$$R(t) = \frac{\rho^2}{S(0)}\left\{\frac{S(0)}{\rho} - 1 + \alpha\tanh\left(\frac{1}{2}\alpha\eta t - \phi\right)\right\}, \qquad (2.5.9)$$

$$\text{where,} \qquad \alpha = \left\{\left(\frac{S(0)}{\rho} - 1\right)^2 + \frac{2S(0)I(0)}{\rho^2}\right\}^{\frac{1}{2}} \qquad (2.5.10)$$

$$\phi = \tanh^{-1}\left(\frac{1}{\alpha}\left(\frac{S(0)}{\rho} - 1\right)\right). \qquad (2.5.11)$$

Knowing (2.5.9) we can then substitute this back into (2.5.6) and (2.5.7) to determine the number of susceptible and infectious individuals respectively, at each time step. Additionally, if we differentiate (2.5.9) with respect to t, we can conclude that

$$\frac{dR(t)}{dt} = \frac{\eta\alpha^2\rho^2}{2S(0)}sech^2\left(\frac{1}{2}\alpha\eta t - \phi\right). \qquad (2.5.12)$$

This is often referred to as the *epidemic curve* and it informs us as to the rate at which the epidemic is progressing.

2.5.0.2 The Total Size of the Epidemic

Another quantity we are often interested in is how many individuals are infected during the course of the outbreak. To determine the total size of the epidemic we consider the limit of $R(t)$ in (2.5.9) as $t \to \infty$:

$$\lim_{t\to\infty} R(t) \;=\; \frac{\rho^2}{S(0)}\left(\frac{S(0)}{\rho} - 1 + \alpha\right). \tag{2.5.13}$$

Following Bailey (1975), if we assume that

$$\frac{2S(0)I(0)}{\rho^2} \ll \left(\frac{S(0)}{\rho} - 1\right)^2 \qquad \text{then} \quad \alpha \approx \left(\frac{S(0)}{\rho} - 1\right)$$

and therefore

$$\lim_{t\to\infty} R(t) \;\approx\; 2\rho\left(1 - \frac{\rho}{S(0)}\right) \;=\; \frac{2\rho}{S(0)}\left(S(0) - \rho\right). \tag{2.5.14}$$

As we have already shown, for an epidemic to occur requires $S(0) > \rho$. Noting that (due to the previous assumption) $S(0) \approx N_{pop}$, if we set $S(0) = \rho + x$ then, as shown by Kermack and McKendrick (1927),

$$\lim_{t\to\infty} R(t) \;\approx\; 2(N_{pop} - x)\left(1 - \frac{N_{pop} - x}{N_{pop}}\right) \;=\; 2x - \frac{2x^2}{N_{pop}}. \tag{2.5.15}$$

Therefore, if an epidemic occurs (2.5.15) will be the magnitude of it.

Here we conclude our discussion of deterministic epidemic models, however, we have only touched briefly upon a selection of the theoretical results. For a thorough review of this form of model we refer the reader to Daley and Gani (2001, Chapter 2). Included are discussions of deterministic epidemic models on non-homogeneous populations, as well as an extension to carrier models, where individuals can be infectious without showing any symptoms. These extensions illustrate the flexibility of deterministic models, which although non-random can be highly informative. Although they will not be our focus, deterministic models are an active area of research and Roberts et al. (2015) provides an overview of the current key challenges faced when using deterministic models to study the epidemiology of infectious diseases.

2.6 Stochastic Models

The work performed in constructing deterministic methods of modelling infectious disease outbreaks formed the basis for modern epidemic modelling. However, by their very nature epidemics will have an inherent randomness in their spread and evolution. As such stochastic models became increasingly popular as a method of capturing this underlying randomness. A discussion of stochastic models is conducted in Britton (2010), where various models are explored and analysed. Additionally, Bailey (1975) and Daley and Gani (2001, Chapters 3 and 4) provide an in-depth discussion of the theoretical properties of continuous and discrete-time stochastic epidemics. Stochastic models are also the focus of Becker (1989), which has an emphasis on household models and an introduction to martingale methods.

2.6.1 The Reed-Frost Model

One of the first stochastic models to gain popularity was the *Reed-Frost chain-binomial model* (see, for example, Abbey (1952), Bailey (1975, Chapter 14), Becker (1989, Chapter 2), Daley and Gani (2001, Chapter 4), or Andersson and Britton (2000)). This is a discrete-time SIR model on a closed, homogeneous population. This model describes outbreaks where the length of the latent period is much longer than the infectious period. As such we assume a single time step is the length of the latent period and individuals' infectious periods are concentrated at the instant of that time step. This means that the infections will occur in generations.

2.6.1.1 The Epidemic Chain

We are interested in tracking the number of susceptible and infectious individuals at each time step (generation), denoted by

$$S_t = \text{Number of susceptibles at time } t, \qquad I_t = \text{Number of infectives at time } t.$$

For the Reed-Frost model we denote the probability a susceptible individual avoids infection from an infectious individual within a single time step by p. Therefore the probability of witnessing a specific number of infectious individuals at time $t+1$, given the state of the system at time t, is

$$
\begin{aligned}
&P(I_{t+1} = i_{t+1} \,|\, S_0 = s_0,\, I_0 = i_0,\, \ldots,\, S_t = s_t,\, I_t = i_t) \\
&= P(I_{t+1} = i_{t+1} \,|\, S_t = s_t,\, I_t = i_t) \\
&= \binom{s_t}{i_{t+1}} \left(1 - p^{i_t}\right)^{i_{t+1}} \left(p^{i_t}\right)^{s_t - i_{t+1}} .
\end{aligned} \tag{2.6.1}
$$

We will often be interested in the *epidemic chain* defined by $\{i_1,\, \ldots,\, i_t, i_{t+1} = 0\}$, with initial conditions $s_0 = n$ and $i_0 = m$ such that the total population is of size $N_{pop} = m + n$. The probability of this realisation is

$$
\begin{aligned}
&P\left(I_1 = i_1,\, \ldots,\, I_t = i_t,\, I_{t+1} = 0 \,|\, S_0 = n,\, I_0 = m\right) \\
=& P\left(I_1 = i_1 \,|\, I_0 = m,\, S_0 = n\right) \ldots P\left(I_{t+1} = 0 \,|\, I_t = i_t,\, S_t = s_t\right) \\
=& \binom{n}{i_1} \left(1 - p^m\right)^{i_1} \left(p^m\right)^{n - i_1} \times \cdots \times \binom{s_t}{0} \left(1 - p^{i_t}\right)^0 \left(p^{i_t}\right)^{s_t} .
\end{aligned} \tag{2.6.2}
$$

We can then easily estimate the value of p using (2.6.2) and maximum likelihood methods (see, Bailey (1975, Chapter 14.3), Becker (1989, Chapter 2)).

For each susceptible individual at time t, the probability that they avoid infection at time $t + 1$ is a function of the number of infectious individuals at time t. If we denote this function by $h(I_t)$, then $S_{t+1} \sim \text{Binomial}(S_t,\, h(I_t))$, where for Reed-Frost model it is assumed that $h(I_t) = p^{I_t}$. The total probability of observing this epidemic is thus the product of a series of binomials, earning it the name of a *chain-binomial model.*

Although our focus is on the Reed-Frost model, another commonly discussed chain binomial model is the *Greenwood model* (see, for example, Becker (1989, page 16)). This model takes $h(I_t) = p$ if $I_t \geq 1$ and $h(I_t) = 1$ otherwise. Therefore the chance of infection is the same when there is a single infective as to when there are multiple. This model is suitable when the environment is ‘saturated’, thus the addition of more infectious individuals does not increase the likelihood of infection. This could be valid, for example, in the case of a well mixing household of individuals where it is unlikely that two infectious members results in a higher likelihood of infection compared to a single.

2.6.1.2 Theoretical Properties

The Reed-Frost model is highly intuitive, leading to its widespread use when first introducing the concept of epidemic models. It additionally lends itself nicely to the deduction of theoretical properties. For example, we can determine the distribution of the final size using a set of recursive equations. Suppose we denote by $P_{m,n}(x)$ the probability that in a population with n initial susceptibles and m initial infectives will have x new infections (not including the m initial infectives), then

$$P_{m,n}(x) = \binom{n}{x} p^{(n-x)(m+x)} P_{m,x}(x), \tag{2.6.3}$$

as shown in Bailey (1975, page 248). Additionally, similar to the deterministic model, Ball (1983b) showed the existence of a threshold result, determined by considering the limit as the population size tends to infinity.

Again more can be said about this form of model and we direct the interested reader to Daley and Gani (2001, Chapter 4) which provides detailed coverage of discrete-time, stochastic models, including the theoretical properties of the Reed-Frost model. Additionally Bailey (1975, Chapter 14) and Becker (1989, Chapter 2) contain a thorough discussion of chain binomial models, including a discussion of more general transmission probabilities.

2.6.2 A General Stochastic Epidemic in Continuous Time

The second stochastic model we consider is the *general stochastic epidemic (GSE)* model in continuous time. This model has been extensively discussed: see, for example, Bailey and Thomas (1971), Bailey (1975, Chapter 6.3), Ball (1983a), Daley and Gani (2001, Chapter 3.3) or Diekmann et al. (2012). We emphasize that this is not to be confused with the *generalised stochastic epidemic* which is also often referred to in the literature as GSE (see, for example, Demiris and O'Neill (2006)). Additionally, perhaps misleadingly, the more general 'standard SIR model' is discussed in Andersson and Britton (2000, Chapter 2) and Britton (2010) where the infectious periods take some arbitrary distribution. The general model is a special case of the generalised and the standard model, which makes assumptions about the form of the infectious period and will be our focus in the remainder of this chapter.

The general stochastic epidemic (GSE) model is analogous to the deterministic model discussed in Section 2.5 therefore, as with the deterministic model, this is an SIR model on a homogeneous, closed population of size N_{pop}. As before we denote the number of susceptible, infectious and removed individuals at time t by $S(t)$, $I(t)$ and $R(t)$ respectively. We additionally assume that we have the same initial conditions:

$$(S(0),\, I(0),\, R(0)) = (N_{pop} - c,\, c,\, 0). \tag{2.6.4}$$

A key differences is that we now assume that each infectious individual has infectious contact with each susceptible individual at points of an (independent) homogeneous Poisson process with rate β. Additionally, once infected individuals have independent and identically distributed infected periods.

For the GSE it is assumed that the infectious periods have an underlying exponential distribution with mean $1/\eta$. As a consequence, this process is memoryless and therefore the epidemic process defined by $(S(t), I(t))$ is Markovian. Consequently the movement between states can be described by a Markov chain:

$$\begin{aligned} (x,\, y) &\longrightarrow (x-1,\, y+1) && \text{with transition rate } \beta S(t)I(t), \\ (x,\, y) &\longrightarrow (x,\, y-1) && \text{with transition rate } \eta I(t) \end{aligned}$$

where we refer to β as the (pairwise) infection rate and η as the removal rate. For this process we have transition probabilities:

$$\begin{aligned} &P\big(S(t+\delta t) - S(t) = -1, I(t+\delta t) - I(t) = 1 \quad | \,\mathcal{H}_t\big) = \beta S(t)I(t)\delta t + o(\delta t), \\ &P\big(S(t+\delta t) - S(t) = 0, \quad I(t+\delta t) - I(t) = -1 \,|\, \mathcal{H}_t\big) = \eta I(t)\delta t + o(\delta t), \\ &P\big(S(t+\delta t) - S(t) = 0, \quad I(t+\delta t) - I(t) = 0 \quad | \,\mathcal{H}_t\big) = 1 - (\beta S(t)I(t) + \eta I(t))\delta t + o(\delta t) \end{aligned} \tag{2.6.5}$$

where $\mathcal{H}_t$ denotes the history of the outbreak at time t.

2.6.2.1 The Final Size

For the GSE we may be interested in determining the final size of the outbreak, this is considerably more difficult than its deterministic counterpart. This is due to the final size now having some underlying distribution. Using the assumption of an exponential infectious period some progress can be made by solving a system of equations formed using a recurrence relationship (see, Whittle (1955), Bailey (1975, Chapter 6)). A similar system of equations for an arbitrary infectious period whose moment generating function exists was described by Ball (1986). However, due to the final size distribution displaying bimodal properties, numerical instabilities can arise. This issue was surmounted by Demiris and O'Neill (2006) by using multiple precision arithmetic, which allows for greater accuracy.

2.6.2.2 The Basic Reproduction Number

Another important quantity in epidemic modelling is the *basic reproduction number*, denoted R_0 (see, Diekmann and Heesterbeek (2000)). This value represents the expected number of secondary infections an infectious individual has during their infectious period, under the assumption they are in a (large) population of susceptibles.

In this model we have assumed an individual is infectious for an average length of $1/\eta$, with pairwise infection rate β. Therefore in the case of the GSE we have $R_0 = \beta N_{pop}/\eta$. We can immediately see a similarity to the threshold for a deterministic epidemic, therefore R_0 is generally seen as a threshold parameter for a stochastic epidemic. If $R_0 > 1$ then each individual is, on average, infecting multiple individuals and thus the epidemic is likely to grow. If $R_0 \leq 1$ then we expect to see the number of infectious individuals decline. For this reason this quantity has remained of great importance within epidemic literature (see, Heesterbeek (2002)).

2.6.2.3 The Likelihood

The final size distribution and the basic reproduction number are important quantities within stochastic epidemic modelling. However, next we change direction and consider the statistical analysis of such a model. We begin by considering the form of the likelihood function, which is key to making inference about the underlying parameters of this outbreak.

To construct the likelihood for this model we shall, for now, assume that the outbreak is fully observed i.e. we know the infection and removal times of each individual (when they enter state I and state R, respectively), and we also assume that the epidemic has concluded. Taking the approach of Andersson and Britton (2000), we can think of the infections and removals as a counting process, with 'counts' being the number of each event that have occurred up until time t. This method is commonly used when constructing a GSE (see, for example, Becker (1989, Chapter 6) and Andersson and Britton (2000, Chapter 9)), although the methods can be generalised to other forms of epidemic model.

The following method of constructing the likelihood has a close relationship to methods used within survival analysis. Intuitively, we can recast the problem as being concerned with how long each individual 'survives' being infected and once infected how long they 'survive' being removed. For texts relating to survival analysis and counting processes we would refer the reader to Martinussen and Scheike (2007, Chapter 3) and Aalen et al. (2008, Chapter 5).

We denote the start of the outbreak as τ, the time of the first infection, and the conclusion of the outbreak as ν, the time of the final removal. We define i^j to be the time individual j becomes infective for $j = 1, \ldots, N_{pop}$, where $i^j = \infty$ if that individual never enters state I. Similarly we will set r^j as the time at which individual j becomes removed, with $r^j = \infty$ if that individual never enters state R. These times are such that

$$\tau = \min_{j=1,\cdots,N_{pop}} \left\{ i^j \right\} \qquad \text{and} \qquad \nu = \max_{j=1,\cdots,N_{pop}} \left\{ r^j \right\}. \tag{2.6.6}$$

We will then denote by $\boldsymbol{i} = (i^1, \ldots, i^{N_{pop}})$ and $\boldsymbol{r} = (r^1, \ldots, r^{N_{pop}})$ the set of infection and removal times respectively and let $\boldsymbol{\theta} = (\beta, \eta)$. To construct the likelihood, $L = L(\boldsymbol{\theta}; \boldsymbol{i}, \boldsymbol{r})$, we divide the time over which the outbreak occurs into short increments of length δt. Formally, we split the timespan of the outbreak into w time-steps:

$$(t_0, t_1], \quad (t_1, t_2], \quad \cdots \quad (t_{w-1}, t_w],$$

where

$$t_0 = \tau, \quad t_1 = t_0 + \delta t, \quad \cdots \quad, t_{w-1} = \nu - \delta t, \quad t_w = \nu,$$

such that $w\delta t = \nu - \tau$. Within each interval we will either observe a single event, or no event.

We split the likelihood into the contributions from the two (independent) processes such that $L = L_1 L_2$ where L_1 represents the infection process and L_2 the removal process. Focusing on the infection process, we begin by defining

$$\Delta S(t) = S(t + \delta t) - S(t). \tag{2.6.7}$$

Then we can write the likelihood of the infection process as

$$L_1 = \prod_{i=0}^{w-1} (\beta S(t_i) I(t_i) \delta t)^{|\Delta S(t_i)|} \, (1 - \beta S(t_i) I(t_i) \delta t)^{(1-|\Delta S(t_i)|)}. \tag{2.6.8}$$

The first term relates to the probability of observing an infection event within that time interval and the second term relates to the probability of observing no infections. For small δt we can see that

$$1 - \beta S(t) I(t) \delta t \approx \exp\{-\beta S(t) I(t) \delta t\}, \tag{2.6.9}$$

therefore we can rewrite (2.6.8) as,

$$L_1 = \prod_{i=0}^{w-1} (\beta S(t_i) I(t_i) \delta t)^{|\Delta S(t_i)|} \, \exp\{-\beta S(t_i) I(t_i) \delta t\}^{(1-|\Delta S(t_i)|)}. \tag{2.6.10}$$

Next we can note that, as infections are assumed to be instantaneous, if we let $\delta t \to 0$ then the event times can be replaced by the infection times. Therefore we can simplify this expression by noting that for the first term, individuals will be infected immediately prior to their infection time and for the second we require the probability of not observing an infection event over the entire outbreak (as $\delta t \to 0$, $w \to \infty$). Thus,

$$L_1 = \left(\prod_{\tau < i^j \leq \nu} \beta S\left(i^{j-}\right) I\left(i^{j-}\right) \delta t \right) \left(\exp\left\{ - \int_{\tau}^{\nu} \beta S(t) I(t) dt \right\} \right) \tag{2.6.11}$$

where we evaluate $\beta S(t) I(t)$ immediately prior to infection (the left limit).

Finally we can divide through by the δt factor observed for each infection event, as this is not dependent on the parameters. To avoid unnecessary confusion we now (re)define the contribution to the likelihood from the infection events to be

$$L_1 = \left(\prod_{\tau < i^j \leq \nu} \beta S\left(i^{j-}\right) I\left(i^{j-}\right) \right) \left(\exp\left\{ - \int_{\tau}^{\nu} \beta S(t) I(t) dt \right\} \right). \tag{2.6.12}$$

We could deduce L_2 in a similar way, however, as we have the assumption of exponentially distributed infectious periods we can easily note that

$$L_2 = \prod_{\tau < r^j \leq \nu} \eta \exp\left\{ -\eta \left(r^j - i^j \right) \right\}. \tag{2.6.13}$$

Often L_2 will be written in a form similar to L_1 (e.g. O'Neill and Roberts (1999), Kypraios (2007)), so we note that

$$L_2 = \prod_{\tau < r^j \leq \nu} \eta \exp\left\{ -\eta \left(r^j - i^j \right) \right\} \propto \left(\prod_{\tau < r^j \leq \nu} \eta I\left(r^{j-}\right) \right) \left(\exp\left\{ - \int_{\tau}^{\nu} \eta I(t) dt \right\} \right). \tag{2.6.14}$$

Altogether, if we combine the infection and removal processes we find

$$L = \left(\prod_{\tau < i^j \leq \nu} \beta S\left(i^{j-}\right) I\left(i^{j-}\right) \right) \left(\exp\left\{ - \int_{\tau}^{\nu} \beta S(t) I(t) dt \right\} \right) \left(\prod_{\tau < r^j \leq \nu} \eta \exp\left\{ -\eta \left(r^j - i^j \right) \right\} \right). \tag{2.6.15}$$

This method of constructing the likelihood can similarly be applied when inference is conducted whilst an outbreak is still progressing.

2.6.2.4 Analysing the Likelihood

Once we have constructed the likelihood we can make inference about the parameters we are interested in. Following O'Neill and Roberts (1999) we allocate β and η Gamma prior distributions, Gamma$(\lambda_\beta, \nu_\beta)$ and Gamma(λ_η, ν_η), respectively. If we denote the total number of infections and removal times observed throughout the outbreak by m (including the initial infection) then, using the prior distributions in conjunction with the likelihood, we have the marginal distributions

$$\pi(\beta \,|\, \eta,\, \boldsymbol{i},\, \boldsymbol{r},\, \tau) \;\sim\; \text{Gamma}\left((m-1)+\lambda_\beta,\;\; \nu_\beta + \int_\tau^\nu S(t)I(t)dt\right), \tag{2.6.16}$$

$$\pi(\eta \,|\, \beta,\, \boldsymbol{i},\, \boldsymbol{r},\, \tau) \;\sim\; \text{Gamma}\left(m+\lambda_\eta,\;\; \nu_\eta + \sum_{\tau<r^j\leq\nu} r^j - i^j\right). \tag{2.6.17}$$

With these distributions constructed we can obtain point estimates, such as the mean, or form credible intervals to learn about the parameters of interest.

We may not always have fully observed data, in this case we can utilise data augmentation and MCMC methods (see, Section 1.4.6) to sample from these distributions. If we assume that $\boldsymbol{r}$ is observed and let $\boldsymbol{\theta} = (\beta,\, \eta)$ then an MCMC could use the following scheme to update the parameters:

Step 1. Update $\boldsymbol{\theta} \,|\, \boldsymbol{i},\, \boldsymbol{r}$

Step 2. Update $\boldsymbol{i} \,|\, \boldsymbol{\theta},\, \boldsymbol{r}$

Step 3. Return to Step 1.

Here standard Gibbs steps can be used to update the parameters (Step 1), whereas Step 2 requires a slightly more complicated proposal. We do not detail the formation of the MCMC as it will be similar to that which we construct and then use in Chapter 3. For further details on using MCMC algorithms in the context of a general stochastic epidemic with partially observed data we refer the reader to O'Neill and Roberts (1999), where the algorithm is constructed for an ongoing outbreak. Additionally, an extension of similar methods to an SEIR outbreak is discussed in Gibson and Renshaw (1998).

2.6.3 Incorporating Heterogeneity

Stochastic models allow for the incorporation of many factors, thus there have been many proposed extensions: a particularly relevant extension is to remove the assumption of a homogeneous population. As of yet, for both the deterministic and stochastic models we have discussed, we have assumed that we are working with a homogeneous population. However, many extensions to both models exist. For example, Ball (1985) considered heterogeneous extensions to both deterministic and stochastic models, where the focus was on individuals with different levels of susceptibility; additionally, heterogeneous extensions are considered in detail in Becker (1989) and in Daley and Gani (2001).

2.6.3.1 Individual Level Models

Many heterogeneous population models focus on splitting the population into M groups, with individuals acting homogeneously within these groups. However, due to advancements in computational capabilities, models at an *individual level*, also referred to as *agent-based models*, have become increasing possible to utilise. These models allow the probability of events occurring to be dependent on the individual characteristics of those involved.

This type of model opens up the possibility for more realistic inference; for example, we could allow the probability of being infected to depend on an individuals proximity to those infected (spatial model), or the length of an individual's infectious period to be dependent on their age. The disadvantage of such models is that they are computationally very costly. However, due to the recent advances in computational capabilities—which make it increasingly possible to utilize methods such as data augmentation and MCMC algorithms—individual level models are becoming increasing popular.

Much of the work previously described is based on an *individual-level model* (ILM). For example, Gibson (1997) used an ILM to model the spread of *citrus tristeza* virus within an orchard, and the model of the Hagelloch measles epidemic by Neal and Roberts (2004) was performed at an individual level.

This form of model has been particularly useful in analysing the 2001 UK Foot-and-Mouth disease (FMD) outbreak, which we will later be interested in, where capturing the characteristics of the farms is important in understanding the spread of the disease. Jewell et al. (2009) and Deardon et al. (2010) analysed the FMD data set using an ILM, where they incorporated the structure and location of the infectious and susceptible farms into the transmission probability. This will be the focus of our work in later chapters, in particular we will be interested in constructing a discrete-time spatial model which is capable of incorporating the characteristics of the susceptible and infectious individuals into their (pairwise) transmission probability.

2.7 Deterministic versus Stochastic Models

We have illustrated in the models discussed that one key choice we must make when analysing an outbreak is if a deterministic or a stochastic model should be used. As described by Bailey (1950), deterministic models assume that

> *"for given numbers of susceptible and infectious individuals and given infection and removal rates, a certain definite number of fresh cases would arise in a given time".*

Clearly deterministic models will not capture the inherent randomness of an outbreak. Britton (2010) noted that deterministic models are more interested in answering questions such as "*How many will get infected if the epidemic takes off?*" In contrast, stochastic models are more interested in "*What is the probability of a major outbreak?*" The word 'probability' is the key here; deterministic models assume that if a certain condition is satisfied then an epidemic will occur. In contrast, stochastic models maintain that there is an inherent randomness in an outbreak that must be accounted for.

It is worth noting that deterministic models provide a reasonable approximation to the stochastic model when we have a large population (see, Diekmann et al. (2012, Chapter 3), Andersson and Britton (2000, Chapter 5)). These approximations can be very useful, however they must be used cautiously (Isham (1991), Isham (2005)).

Which model is more appropriate will depend on the application we are interested in. In many ways a deterministic model can be simpler and thus allow greater headway to be made in the analysis of the outbreak; a realistic model is of little use if it is completely intractable. However, a stochastic model can offer a more accurate description of the outbreak, which acts closer to how we would expect an epidemic to behave. Additionally, due to advancements in computational resources, simulations methods can be used on a greater range of problems. As such, stochastic models of increasing complexity can be analysed.

2.8 Discussion

In this chapter we have discussed a collection of epidemic models. Underpinning each model is a set of assumptions which must be made to produce a tractable problem. These assumptions are a result of questions such as, how to handle the problem of missing data and how to conduct inference in a reasonable amount of time. There are two key ways of approaching problems such as these:

1. Mathematically Driven Inference

The first is to focus on the mathematical properties of the models used to make inference about an outbreak. These methods are typically interested in the properties of the system constructed, therefore stronger assumptions about the outbreak are often made. These systems are generally further away from the truth, although they do allow for significant progress to be made in determining the theoretical properties of such outbreaks.

2. Data Driven Inference

The second approach is to perform analysis motivated by the collection of data. This is often the front line of statistical infectious disease analysis and that which is closest to the inference we aim to make. This form of analysis often involves the construction of the likelihood function and then utilising MCMC methods to make inference about the underlying parameters. These methods are well suited for incorporating heterogeneities and thus are generally tailored towards a specific outbreak.

The methods we will be developing are to be used in the real-time analysis of outbreak data. As a result we will take a data driven approach, utilising computing power and simulation methods to learn about the parameters of interest. We will be interested in constructing an epidemic model and analysis that works at an individual level. Thus, the general theoretical properties of the outbreak are not what we are interested in: rather, we wish to make inference about parameters which we believe are driving the outbreak, which will be specific to that epidemic.

Chapter 3

Developing Sequential Monte Carlo Methods for Epidemic Data

A large portion of the inference performed on epidemic data is conducted using Markov chain Monte Carlo (MCMC) methods. This is due to the highly flexible nature of these algorithms, as well as their ability to utilise data augmentation techniques to handle the problem of missing data. However, infectious disease outbreaks will often occur rapidly, with new information being obtained daily. With each new piece of data an MCMC algorithm must restart to produce parameter estimates. As a field of research that benefits greatly from on-line inference, MCMC methods do not appear to be the most suitable choice. Intuitively, a sequential method of updating the parameter estimates as new information is obtained would be better suited to the problem. This will act as the motivation for developing sequential Monte Carlo (SMC) methods with applications to epidemic data. The question that we will henceforth be focusing on is: how can we update the samples produced from the posterior distribution at time t to incorporate the new data collected at time $t+1$?

3.1 The Problem Statement

We have previously discussed in Chapter 2 the historic and current methods used within epidemic modelling. We shall focus on utilising simulation methods to generate samples from the posterior distribution of the parameters, given some observed data.

We are interested in posterior distributions of the form discussed in Section 1.6, i.e.

a sequence of distributions that evolves with time, t. This form of distribution is a function of an observed process, denoted by $\boldsymbol{x}_{0:t}$, and an unobserved process denoted by $\boldsymbol{y}_{\tau:t}$, where '0' is the time of the first observed event and τ represents the time of the first event in the unobserved process, such that $\tau \leq 0$. Therefore, at time t, we are interested in constructing the posterior distribution

$$\pi(\boldsymbol{\theta},\, \boldsymbol{y}_{\tau:t} \,|\, \boldsymbol{x}_{0:t}) \;\propto L(\boldsymbol{\theta}\,;\, \boldsymbol{y}_{\tau:t},\, \boldsymbol{x}_{0:t})\pi(\boldsymbol{\theta}). \tag{3.1.1}$$

A method of learning about the posterior distribution in (3.1.1) is to generate samples from it. This can be achieved by using the MCMC methods discussed previously in Section 1.4 (which we adapt for the epidemic setting in Section 3.3). However, the question we are concerned with is, can we use the samples generated for $\pi(\boldsymbol{\theta},\, \boldsymbol{y}_{\tau:t} \,|\, \boldsymbol{x}_{0:t})$ to inform us about the form of $\pi(\boldsymbol{\theta},\, \boldsymbol{y}_{\tau:t+1} \,|\, \boldsymbol{x}_{0:t+1})$?

3.1.1 Chapter Breakdown

In this chapter we will discuss the construction of a sequential Monte Carlo algorithm which utilises epidemic data.

Before we can adapt the SMC methods, we need to construct the posterior distribution which we wish to analyse. Therefore in Section 3.2 we describe the framework under which we will perform analysis. This section begins by describing the model assumptions we make to describe the epidemic in Section 3.2.1. Once we have stated these assumptions we can begin forming the posterior distribution, with discussion of the data augmentation scheme we choose to use in Section 3.2.2. We then proceed to provide a detailed description of the notation we use throughout in Section 3.2.3, this then allows us to formally describe the likelihood function in Section 3.2.4, which forms the key part of the posterior distributions we are interested in.

With the posterior distribution constructed we proceed to developing the simulation techniques we will use to generate samples from it. We begin by developing an MCMC algorithm in Section 3.3. Although our focus will be on the development of an SMC algorithm we are still interested in the analogous MCMC. This is for two reasons: firstly, we will assess the performance of the SMC by comparing it to the current gold-standard: MCMC methods. Secondly, we will utilise the MCMC to perform the movement step within the SMC algorithm (see, Section 1.6.3).

In Section 3.4 we begin adapting the ideas of the SMC methods discussed in Chapter 1 to be applied in the framework of the epidemic model we have described. The application is not straightforward and as such in the following sections we describe each stage of the SMC algorithm in detail, with a final summary of the algorithm constructed shown in Section 3.4.9. Additionally, in Section 3.5, we discuss an extension to the constructed SMC algorithm, which further utilizes information about the infectious periods of individuals to potentially improve inference.

Finally in Section 3.6 we describe an extension of the methods developed to an agricultural epidemic, motivated by the desire to apply the methods to the 2001 UK Foot-and-Mouth outbreak. The SMC methods developed can be readily applied to this type of outbreak, thus in this section we primarily focus on the form of the likelihood function.

3.2 A Discrete-Time Stochastic Epidemic Model

We described in Chapter 2 a collection of the current and historic work performed in epidemic modelling. Keeping in mind these advancements, we begin by constructing the posterior distribution which shall form the focus of our analysis. Our aim is to take the next step in utilizing increasingly advanced computational capabilities to construct a realistic epidemic model, on which we can apply simulation methods to learn about the outbreak. As such we must also keep in mind the techniques we described in Chapter 1, which we will hope to apply.

3.2.1 Model Choices

Before we can construct the posterior distribution we need to define the underlying epidemic model, which we shall use throughout the remaining chapters. This model will be used in conjunction with the observed data to form the posterior distribution of interest. We outline next the various choices we make in order to construct our model.

The Population

We assume that the outbreak occurs on a closed population, with the disease introduced by a single individual. This individual will be inferred with the other unknowns and corresponds to the individual with the earliest inferred infection time. We additionally will be interested in constructing a model capable of conducting inference at an individual-level. As such we currently make no further assumptions about the homogeneity of the population.

Continuous or Discrete Time?

As we highlighted in Section 2.3.1, the majority of current research on infectious disease modelling focuses on continuous-time models (see, for example, Gibson and Renshaw (1998), Jewell et al. (2009) and Xiang and Neal (2014)). These models intuitively make sense, as infections will spread on a continual-time basis. In contrast, an outbreak is usually observed in single, equally spaced, time steps (e.g. once a day, once a week etc.), and therefore the data we have access to will often be in discrete time (see, for example, Neal and Roberts (2004), Deardon et al. (2010)). Therefore, discrete time models can match the form of the data more realistically. However, a continuous model is often more flexible, as it can represent the disease dynamics in a way that is not dependent on the method of data collection associated with it.

We will choose to focus on discrete-time models throughout, which will fit the form of the data more closely. This choice is mainly for pragmatic reasons, as SMC methods are primarily used for the exploration of an evolving set of distributions in discrete time. As such, we assume that infections are concentrated in the instance between time t and time $t + 1$. Therefore an individual becomes infectious for the first time at time $t + 1$, having been exposed to infectious individuals at time t.

Compartmental Framework

Throughout we will primarily use an SIR model as it forms a basic template, which can be easily extended to more complex compartmental frameworks (see, Section 2.4). Therefore we assume that individuals move between three possible states:

$$\text{Susceptible} \quad \longrightarrow \quad \text{Infectious} \quad \longrightarrow \quad \text{Removed},$$

and once removed an individual cannot be infected again.

The Data

We assume the data we have access to takes the form of the removal times of each individual within a closed population. We assume that these times are recorded daily, up until the time at which we begin analysis. As such we know when individuals enter state R but not when they enter state I. This is reasonable as often we will only have recordings of when individuals first show symptoms, which we will assume coincides with an individual becoming removed (see, for example, Xiang and Neal (2014)).

Stochastic or Deterministic?

Following the discussion in Section 2.7, due to its ability to capture the randomness observed within an epidemic, we choose to construct a stochastic epidemic model.

Transmission Process

When constructing our model we aim to keep it fairly general, as choices about the behaviour of the population will be highly dependent on the outbreak we are considering. As such, we reserve discussion of this for later sections. In general we assume that each pair of individuals has contact with some probability, independent of all other individuals, with contact between a susceptible and infectious individual resulting in infection. Additionally we will allow the (pairwise) transmission probability to be a function of the properties of both the infectious and the susceptible individual.

Removal Process

Similar to the transmission process, we keep the distribution of the infectious period in a general form to allow for a flexible model. Therefore, we assume that the individuals' infectious periods are independent and identically distributed according to some known (discrete) distribution, often a function of unknown parameters.

3.2.1.1 Summary of the Key Model Assumptions

For clarity we collect the key assumptions we will make when constructing our epidemic model:

- The epidemic occurs within a closed population.

- There is initially a single infectious individual.

- The epidemic occurs in discrete time.
- This is an SIR-type outbreak.
- We observe when individuals become removed but not when they are infected.
- At each time step an infectious individual infects each susceptible individual with some probability. This probability will usually be based on certain characteristics of the susceptible and infectious individuals and will be independent of all other infections.
- Once infected an individual's infectious period is assumed to follow a known distribution. This will be a function of underlying, and often unknown, parameters.

3.2.2 The Posterior Distribution

With the assumptions of the model defined we can begin construction of the posterior distribution, which will form the focus of our analysis. We begin by considering a posterior distribution of the form

$$\pi(\boldsymbol{\theta}\,|\,\boldsymbol{x}_{0:t}) \;\propto\; L(\boldsymbol{\theta}\,;\,\boldsymbol{x}_{0:t})\,\pi(\boldsymbol{\theta}), \tag{3.2.1}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots \theta_d)$ contains the underlying parameters of the model, such that $d \geq 1$ and $\boldsymbol{x}_{0:t}$ is the observed data up to time t. Throughout we will assume that $\boldsymbol{x}_{0:t}$ contains the times individuals enter state R (removal times). We will also assume that we have independent priors, such that,

$$\pi(\tilde{\boldsymbol{\theta}}) = \prod_{k=1}^{d} \pi_{\theta_k}\left(\tilde{\theta}_k\right), \tag{3.2.2}$$

where $\pi_{\theta_k}(\cdot)$ is the prior distribution of θ_k.

We desire a method for conducting inference whilst an outbreak is still ongoing: as such, there will be individuals who are currently infectious, that we have not observed being removed. We refer to these as the *occult* individuals, taken from the medical term:

"*occult: (Medicine)(of a disease or process) not accompanied by readily discernible signs or symptoms*" (Oxford Dictionaries (2018)).

Additionally we choose to refer to those individuals we have observed as being infectious as the *observed infectives*, these are the individuals for whom we have an observed time of removal.

3.2.2.1 Data Augmentation

To ensure the likelihood is tractable we choose to utilise the data augmentation methods discussed previously in Section 1.4.6. We have a choice of which data augmentation scheme we use. It is possible to generate the random graph on which individuals have contact, as seen in Demiris and O'Neill (2005) and O'Neill (2009), however, the methods discussed are not easily applied to large data sets. Additionally, these applications are conducted post-outbreak: they are likely to struggle when the number of infectious individuals is unknown. For this reason, assuming that we have access to the removal times, we shall focus on augmenting the unknown infection times of each individual, as well as who is infectious (see, for example, Jewell et al. (2009)). Additionally we will infer the removal times of the occult individuals to ensure that we can easily construct and evaluate the likelihood. This avoids the need to consider the cdfs of sojourn-time distributions, at the cost of including an additional dimension.

In summary, we choose to infer the infection times of those individuals with an observed removal time, as well as inferring the occult individuals and their infection and removal times (or equivalently their infection times and infectious period). This forms the unobserved information, $\boldsymbol{y}_{\tau:t}$, where τ is the (to be inferred) time of the first infection. The addition of this term will ensure the likelihood can be evaluated. This will serve only to reduce the difficulty of the problem and will not affect the inference made. As such we now return our attention to the posterior distribution

$$\pi(\boldsymbol{\theta},\, \boldsymbol{y}_{\tau:t}\,|\,\boldsymbol{x}_{0:t}) \;\propto L(\boldsymbol{\theta}\,;\, \boldsymbol{y}_{\tau:t},\, \boldsymbol{x}_{0:t})\,\pi(\boldsymbol{\theta}). \tag{3.2.3}$$

3.2.3 Summary of Notation

To formally define the posterior distribution we choose to first set our notation from the start, to allow for clarity of exposition throughout this chapter.

3.2.3.1 The States

As stated, throughout we will assume that we observe when individuals are removed, but not when they are infected. We denote by m_t^I the number of individuals who have been infected at or before time t and define m_t^R as the number of removals observed at or before time t. Therefore, $m_{t-1}^I \leq m_t^I$, $m_{t-1}^R \leq m_t^R$ and $m_t^R \leq m_t^I$, for all times t. We denote by m_t^S the number of individuals in the population who are susceptible *at* time t. This is slightly different to the analogous terms for states I and R as for all time steps, t, $m_t^S \leq m_{t-1}^S$. We define the fixed size of the population by N_{pop}, then at all times, t, $N_{pop} = m_t^I + m_t^S$. We note that the values for m_t^S and m_t^I are usually unknown, therefore they will be implicitly included in the unknowns we are interested in determining. If we denote the number of occult individuals at time t by u_t then this acts as the unknown part of m_t^I, such that $m_t^I = u_t + m_t^R$ where m_t^R is known for all t. As such it will often be the value of u_t that we are interested in inferring, as well as which individuals are the occults.

To identify the individuals within the population we label each one by an index number: $1, \ldots, N_{pop}$. This index will only be used for identification and will not be related to an individual's status or covariates (e.g. location). Using these indexes we can keep track of which individuals are in each state, at each time step. We denote by $\mathcal{S}_t$, $\mathcal{I}_t$, and $\mathcal{R}_t$ the indexes of the individuals in state S, I, or R, respectively, at time t. As such we have $\mathcal{S}_t \cup \mathcal{I}_t \cup \mathcal{R}_t = \{1, \ldots, N_{pop}\}$ and $\mathcal{S}_t \cap \mathcal{I}_t = \mathcal{S}_t \cap \mathcal{R}_t = \mathcal{I}_t \cap \mathcal{R}_t = \emptyset$, for all t. In addition we may refer to the new observations at time t, denoted by $\mathcal{V}_t = \mathcal{R}_t \backslash \mathcal{R}_{t-1}$ with $v_t = |\mathcal{V}_t|$.

3.2.3.2 The Times

We define τ as the time of the first infection, its value is not observed and therefore it will be inferred along with the parameters. Throughout we will take τ to be the earliest inferred infection time, updated when we update the infection times. This is discussed further in Section 3.3. We will be starting our analysis at time $T \geq \tau$, at which point the epidemic may have just have started, reached its peak or be completed. We additionally will shift the data so that time $t = 0$ corresponds to the time of the first observed removal. As such, the time at which we first analyse the data will be T time steps after the first observed removal ($\tau \leq 0 \leq T$). Therefore, with respect to this notation, some infection times will occur at negative time points.

The infection times of those individuals with an observed removal time will need to be inferred, as will the infection and removal times of the occult individuals. This ensures that the likelihood assumes a tractable form and will be discussed in greater detail in the next sections. For an individual with index k we denote their time of infection by i^k and their removal time as r^k. If k does not become infected during the outbreak then we set $i^k = r^k = \infty$. As these values will change dependent on when we observe them we define i_t^k and r_t^k as the infection and removal times of individual k at time t, such that

$$i_t^k = \begin{cases} i^k & \text{if} \quad i^k \leq t \\ \infty & \text{if} \quad i^k > t \end{cases} \qquad r_t^k = \begin{cases} r^k & \text{if} \quad i^k \leq t \\ \infty & \text{if} \quad i^k > t \end{cases} . \tag{3.2.4}$$

We highlight that, under this notation, it is possible for an individual to be removed after time t, i.e. $r_t^k > t$. These two pieces of information, for each individual, then form the data that we shall use to construct the posterior distribution of interest. We additionally define data relating specifically to time t as,

$$\boldsymbol{i}_t = \left\{ i_t^k : k \in \mathcal{I}_t \backslash \mathcal{I}_{t-1} \right\} \qquad \text{and} \qquad \boldsymbol{r}_t = \left\{ r_t^k : k \in \mathcal{R}_t \backslash \mathcal{R}_{t-1} \right\}, \tag{3.2.5}$$

such that $\boldsymbol{i}_{\tau:t} = (\boldsymbol{i}_\tau, \dots, \boldsymbol{i}_t)$ and $\boldsymbol{r}_{0:t} = (\boldsymbol{r}_0, \dots, \boldsymbol{r}_t)$.

We may often be interested in the infectious period of an individual: this is the time between an individual's infection and their removal. We denote the infectious period of individual k in a similar way to the infection and removal times:

$$h^k = \begin{cases} r^k - i^k & \text{if} \quad i^k \neq \infty \\ 0 & \text{otherwise} \end{cases} \qquad h_t^k = \begin{cases} r_t^k - i_t^k & \text{if} \quad i^k \leq t \\ 0 & \text{if} \quad i^k > t \end{cases}, \tag{3.2.6}$$

where we assume h^k are independent and identically distributed realisations of H, an arbitrary non-negative distribution with probability mass function $g_H(\cdot)$. We assume that g_H is a known function of unknown parameters i.e. the infectious periods belong to some parametric family of distributions, but the exact parameter values are unknown. Note that we only require two values from h^k, i^k and r^k, to determine the third. Thus, we may switch between working with the infection and removal times, and working with the infectious period and either the infection times or the removal times.

We are often also interested in the infection and removal times of individuals in a particular state. Thus we define

$$\begin{aligned}
\boldsymbol{i}^{I}_{\tau:t} &= \left\{ i^k_t : k \in \mathcal{I}_t \right\}, & \boldsymbol{i}^{R}_{\tau:t} &= \left\{ i^k_t : k \in \mathcal{R}_t \right\}, \\
\boldsymbol{r}^{I}_{0:t} &= \left\{ r^k_t : k \in \mathcal{I}_t \right\}, & \boldsymbol{r}^{R}_{0:t} &= \left\{ r^k_t : k \in \mathcal{R}_t \right\}, \qquad (3.2.7)
\end{aligned}$$

such that we can instead define $\boldsymbol{i}_{\tau:t} = \left\{ \boldsymbol{i}^{R}_{\tau:t}, \boldsymbol{i}^{I}_{\tau:t} \right\}$ and $\boldsymbol{r}_{0:t} = \boldsymbol{r}^{R}_{0:t}$. We are therefore splitting the infection and removal times into those belonging to individuals who are removed at time t (‘R’), who we refer to as the observed infectives, and those who are infectious at time t (‘I’), the occults.

Using the notation for the times and the augmentation scheme we have described, we define the observed data to be

$$\boldsymbol{x}_{0:t} = \boldsymbol{r}_{0:t} = \boldsymbol{r}^{R}_{0:t} \qquad (3.2.8)$$

and the unobserved data to be

$$\boldsymbol{y}_{\tau:t} = \left\{ \boldsymbol{i}_{\tau:t}, \boldsymbol{r}^{I}_{0:t} \right\} = \left\{ \boldsymbol{i}^{R}_{\tau:t}, \boldsymbol{i}^{I}_{\tau:t}, \boldsymbol{r}^{I}_{0:t} \right\}. \qquad (3.2.9)$$

Therefore the posterior distribution at time t is

$$\pi(\boldsymbol{\theta}, \boldsymbol{y}_{\tau:t} \,|\, \boldsymbol{x}_{0:t}) = \pi(\boldsymbol{\theta}, \left\{ \boldsymbol{i}^{R}_{\tau:t}, \boldsymbol{i}^{I}_{\tau:t}, \boldsymbol{r}^{I}_{0:t} \right\} \,|\, \boldsymbol{r}^{R}_{0:t}). \qquad (3.2.10)$$

We will switch between the alternative notations as appropriate.

3.2.4 Constructing the Likelihood

We are now in a position to construct the likelihood function. We begin by noting that knowledge of the infection and removal times is equivalent to knowledge of which individuals are in each state, at each time step. As such to construct the likelihood,

$$L(\boldsymbol{\theta};\, \boldsymbol{y}_{\tau:t},\, \boldsymbol{x}_{0:t}) \,=\, L(\boldsymbol{\theta};\, \left\{\boldsymbol{i}^R_{\tau:t},\, \boldsymbol{i}^I_{\tau:t},\, \boldsymbol{r}^I_{0:t}\right\},\, \boldsymbol{r}^R_{0:t}),$$

we will focus two processes:

(i) Transmission Process: individuals moving from state S to state I as a result of infectious contact.

(ii) Removal Process: individuals moving from state I to state R due to their infectious period ending.

This construction shares many similarities to the general stochastic epidemic model considered in Section 2.6.2. Considering each part we note that for (i) we need to consider the infections that occur each day, as well as those individuals that escape infection. This is analogous to the continuous-time process which counts the infections which occur and will be similar to the chain binomial model discussed in Section 2.6.1. Part (ii) will contain the probability of witnessing each individual's infectious period and is analogous to the continuous-time removal process, as seen in Section 2.6.2.

We denote by $P_t(\ell\,;\boldsymbol{\theta})$ the probability that individual ℓ avoids infection at time t and therefore is still susceptible at time $t+1$. Then the likelihood is

$$L(\boldsymbol{\theta};\, \boldsymbol{y}_{\tau:t},\, \boldsymbol{x}_{0:t}) \,=\, \underbrace{\prod_{s=\tau}^{t-1}\left\{\prod_{\ell\in\mathcal{S}_{s+1}} P_s(\ell\,;\boldsymbol{\theta}) \prod_{\ell\in\mathcal{S}_s\setminus\mathcal{S}_{s+1}} \big(1-P_s(\ell\,;\boldsymbol{\theta})\big)\right\}}_{\text{(i) Transmission Process}} \underbrace{\prod_{j\notin\mathcal{S}_t} g_H(h^j_t\,;\boldsymbol{\theta})}_{\text{(ii) Removal Process}}\ . \tag{3.2.11}$$

To avoid infection an individual must avoid transmission of the disease from each infectious individual, therefore

$$P_t(\ell\,;\boldsymbol{\theta}) \,=\, \prod_{j\in\mathcal{I}_t} p_t(\ell,\, j), \tag{3.2.12}$$

where $p_t(\ell, j)$ is the probability that individual ℓ avoids transmission at time t from individual j. Linked to this we may instead work with $q_t(\ell, j) = 1 - p_t(\ell, j)$ which is the probability that individual ℓ is infected by individual j, at time t. When $p_t(\ell, j) = p$ if $j \in \mathcal{I}_t$, we can see the similarity to the Reed-Frost model discussed previously in Chapter 2. Strictly $p_t(\ell, j) = p_t(\ell, j\,;\,\boldsymbol{\theta})$, however, we will usually drop the explicit reference to $\boldsymbol{\theta}$ to avoid overly cumbersome notation.

In this section we shall keep g_H and P_t in this general form, as their selection will be highly problem specific. More detailed and tailored examples will be discussed in Chapters 4 and 5.

As a side note, often we will find that calculation of the likelihood is computationally very expensive, especially for larger populations. However, we can reduce the cost somewhat by calculating the likelihood using a sequential procedure. This is discussed in Appendix A.1.

3.3 The MCMC Algorithm

Recall that in Section 1.4 we discussed the construction of an MCMC algorithm for sampling from a distribution of interest. In this section we consider a more tailored algorithm, specifically with epidemic data in mind. As the precise form of the algorithm will be specific to each outbreak, we shall only discuss a generic algorithm for now. We will follow a similar updating scheme to those discussed in Gibson and Renshaw (1998), O'Neill and Roberts (1999), Jewell et al. (2009), Xiang and Neal (2014) and Lee and Neal (2018), all of which required the updating of unobserved infection time events.

It is not necessarily trivial to choose an appropriate proposal distribution, primarily due to the posterior distribution being constructed whilst the outbreak is still occurring. As a result we do not know how many individuals are currently infectious and thus must infer this value, this requires the dimension of the parameter space to be able to shrink and grow. Therefore, we shall use the ideas of the reversible-jump MCMC methods (see, for example, Gibson and Renshaw (1998)) found in Section 1.4.7, a method used when the dimension of the parameter space is unknown.

We are interested in producing samples for the unobserved data, $\boldsymbol{y}_{\tau:t}$, as well as samples for the parameters underpinning the outbreak, denoted by $\boldsymbol{\theta}$. To achieve this we can use an MCMC algorithm with the following updating schema:

Step 1: Update $\boldsymbol{\theta} \,|\, \boldsymbol{y}_{\tau:t},\, \boldsymbol{x}_{0:t}$

Step 2: Update $\boldsymbol{y}_{\tau:t} \,|\, \boldsymbol{\theta},\, \boldsymbol{x}_{0:t}$

Step 3: Return to Step 1.

We consider each of the steps individually in the following sections. Throughout we will assume that we are in iteration j of the MCMC and wish to propose the values for iteration $j+1$. Thus, when discussing the proposal steps, we assume that we currently have parameter values $\boldsymbol{\theta}^{(j)}$ and unobserved data $\boldsymbol{y}_{\tau:t}^{(j)}$. It should be noted that strictly speaking the time of the initial infection, corresponding to the earliest inferred infection time, is denoted $\tau = \tau^{(j)}$, as it too is inferred within the parameters. However, we drop the notation in order to remain succinct. Additionally, throughout we indicate proposals with a '$*$'.

3.3.1 Step 1: Update $\boldsymbol{\theta}$

We begin by updating the underlying parameters of the epidemic model we have defined. These will be problem specific but will frequently be divided between those relating to the transmission of the infectious disease and those relating to the progression of the infection within an individual. To update $\boldsymbol{\theta}$ we can either update each parameter individually, update the parameters in blocks or update all of the parameters at once. Additionally if parameter $\theta_k \in \boldsymbol{\theta}$ has a marginal distribution that takes a known form then we can use a Gibbs step to update it (see Section 1.4.3). The choice of updating scheme will be problem dependent and as such we do not discuss this in full here, instead we refer the interested reader to Section 1.4. Once updated we set $\boldsymbol{\theta}^{(j+1)}$ as the new parameter values.

3.3.2 Step 2: Update $\boldsymbol{y}_{\tau:t}$

In this section we shall focus on updating the unobserved data. The updating scheme we will implement is unique, although it shares many similarities to those previously used to infer the augmented data within a model of this form.

We choose to split the updating of $\boldsymbol{y}_{\tau:t}$ into two stages, as it contains two distinct pieces of information. For individuals whose removal time we have observed we need only infer their infection time, from this we can then infer their infectious period. For those individuals who are occults we must first determine who they are and then infer their infection and removal times (or equivalently just one of these times and their infectious period). This task involves changing the dimension of $\boldsymbol{y}_{\tau:t}$ itself and thus we restrict this to a separate step. Therefore the MCMC scheme is:

Step 1: Update $\boldsymbol{\theta} \,|\, \boldsymbol{y}_{\tau:t},\, \boldsymbol{x}_{0:t}$

Step 2: Update $\boldsymbol{y}_{\tau:t} \,|\, \boldsymbol{\theta},\, \boldsymbol{x}_{0:t}$

(a) Update the data relating the observed infectives, those in state R at time t, denoted by $\boldsymbol{i}^R_{\tau:t}$.

(b) Update the data relating to the unobserved infectives (the occults), those in state I at time t, denoted by $\left\{\boldsymbol{i}^I_{\tau:t},\, \boldsymbol{r}^I_{0:t}\right\}$.

Step 3: Return to Step 1.

By separating the updating steps we will better explore the sample space we are interested in and thus ensure the algorithm can quickly converge to the desired distribution. In summary, for the unobserved data, $\boldsymbol{y}_{\tau:t} = \left\{\boldsymbol{i}^R_{\tau:t},\, \boldsymbol{i}^I_{\tau:t},\, \boldsymbol{r}^I_{0:t}\right\}$, we choose to update $\boldsymbol{i}^R_{\tau:t}$ and $\left\{\boldsymbol{i}^I_{\tau:t},\, \boldsymbol{r}^I_{0:t}\right\}$ separately. We shall discuss the updating of $\boldsymbol{i}^R_{\tau:t}$ in Section 3.3.2.1 and the updating of $\left\{\boldsymbol{i}^I_{\tau:t},\, \boldsymbol{r}^I_{0:t}\right\}$ in Section 3.3.2.2.

3.3.2.1 Step 2(a): Update the Observed Infectives

We begin by focusing on updating the infection times of those individuals we know to have been infected during the outbreak, i.e. we have observed their removal time and at time t they are in state R. There are m^R_t of these individuals and we shall propose to update a subset of them in each iteration.

To update the infection times of the removed individuals we choose to, in each iteration, select a random sample $F \subseteq \mathcal{R}_t$, of size $|F| = \mu_i \leq m^R_t$, from the set of individuals with an observed removal time, where μ_i is the tuning value for this proposal step. Throughout we will have several *tuning parameters* which determine the acceptance rate of the MCMC we construct; we will discuss the choice of these values in a later section. We choose to update a fixed number of the observed infection times as we have observed the value of m^R_t. This is in contrast to the number of occult individuals, u_t, as we shall discuss in the next section.

Once selected, we update the μ_i individuals in the following way:

- For $\ell \in F$

  (i) Generate a proposed infectious period, $h_t^{\ell,*} \sim g_H$.

  (ii) Denote the proposed infection time as $i_t^{\ell,*} = r_t^\ell - h_t^{\ell,*}$.

- For $\ell \in \mathcal{R}_t \backslash F$

  (i) Keep the infectious period and infection time the same, such that $h_t^{\ell,*} = h_t^{\ell,(j)}$ and $i_t^{\ell,*} = i_t^{\ell,(j)}$.

We shall denote the new set of proposed infection times for the removed individuals by

$$\boldsymbol{i}_{\tau:t}^{R,*} = \left\{ i_t^{k,*} : k \in \mathcal{R}_t \right\}, \tag{3.3.1}$$

although this will contain some times that have not been changed.

This proposal will produce an acceptance probability of zero if it is not consistent with the data—for example if it leads to individuals being infected when no individuals are infectious. Otherwise, recalling the methods of Section 1.4.4, we have acceptance probability

$$\alpha = \min\left(1, \quad \frac{\pi\left(\boldsymbol{\theta}^{(j+1)},\ \boldsymbol{i}_{\tau:t}^{R,*},\ \ \boldsymbol{i}_{\tau:t}^{I,(j)},\ \boldsymbol{r}_{0:t}^{I,(j)} \mid \boldsymbol{r}_{0:t}^{R}\right)\ P\left(\boldsymbol{i}_{\tau:t}^{R,*} \ \longrightarrow\ \boldsymbol{i}_{\tau:t}^{R,(j)}\right)}{\pi\left(\boldsymbol{\theta}^{(j+1)},\ \boldsymbol{i}_{\tau:t}^{R,(j)},\ \boldsymbol{i}_{\tau:t}^{I,(j)},\ \boldsymbol{r}_{0:t}^{I,(j)} \mid \boldsymbol{r}_{0:t}^{R}\right)\ P\left(\boldsymbol{i}_{\tau:t}^{R,(j)} \ \longrightarrow\ \boldsymbol{i}_{\tau:t}^{R,*}\right)} \right), \tag{3.3.2}$$

where $P(x_1 \longrightarrow x_2)$ is the probability of proposing a new sample, x_2, given the current value, x_1. For this updating step we find

$$P\left(\boldsymbol{i}_{\tau:t}^{R,(j)} \longrightarrow \boldsymbol{i}_{\tau:t}^{R,*}\right) = \binom{m_t^R}{\mu_i}^{-1} \prod_{\ell \in F} g_H\left(h_t^{\ell,*}\,;\, \boldsymbol{\theta}^{(j+1)}\right), \tag{3.3.3}$$

$$P\left(\boldsymbol{i}_{\tau:t}^{R,*} \longrightarrow \boldsymbol{i}_{\tau:t}^{R,(j)}\right) = \binom{m_t^R}{\mu_i}^{-1} \prod_{\ell \in F} g_H\left(h_t^{\ell,(j)}\,;\, \boldsymbol{\theta}^{(j+1)}\right). \tag{3.3.4}$$

If accepted we set $\boldsymbol{i}_{\tau:t}^{R,(j+1)} = \boldsymbol{i}_{\tau:t}^{R,*}$, otherwise $\boldsymbol{i}_{\tau:t}^{R,(j+1)} = \boldsymbol{i}_{\tau:t}^{R,(j)}$. We note that by changing the times we are implicitly updating the indexes in each state, $\mathcal{S}_f$ and $\mathcal{I}_f$, for each $f \leq t$, that may change as a result. As such, we denote the updated sets as $\mathcal{S}_f^{(j+1)}$ and $\mathcal{I}_f^{(j+1)}$, for each $f \leq t$. Additionally, we note that, as we have not updated the occult individuals, $\mathcal{I}_t^{(j+1)} = \mathcal{I}_t^{(j)}$ and $\mathcal{S}_t^{(j+1)} = \mathcal{S}_t^{(j)}$, regardless of if we accept or reject this proposal.

Throughout we take the time of the first infection, τ, to be the earliest imputed infection time. Thus this is equivalent to applying a uniform prior to the time of initial infection. Within the likelihood, the initially infectious individual does not contribute to the infection process, as we have assumed that the outbreak began with a single, assumed to be spontaneous, infection.

Following the discussion in Section 1.4.4.3, the value of the tuning parameter μ_i will be chosen to achieve a reasonable acceptance rate between 0.25 ± 0.15. This will be difficult to precisely achieve, so we take inspiration from the methods of Section 1.4.4.4 and adaptively tune the MCMC so that we achieve an acceptance rate within the desired range. This will be further discussed in Section 3.3.6.

Overall this step of the MCMC algorithm is similar to the updating step used in previous applications of MCMC methods to epidemic data. From their early use the updating of infection times has been an essential component of the application of MCMC methods to epidemic data, for example it is key to both Gibson and Renshaw (1998) and O'Neill and Roberts (1999). However, both of these papers only allow the movement of a single infection time at once and this new time is proposed uniformly at random. We choose to extend this idea by allowing multiple new infections to be proposed at once and either all accepted or all rejected. This could result in a low acceptance probability, however, this is counteracted by the generation of the proposal times from the assumed infectious period distribution. This updating step shares many similarities to the updating scheme seen in Jewell et al. (2009), where the infection times are generated using the underlying infectious period distribution and non-centering methods. Additionally, a similar version of this updating step is used in Xiang and Neal (2014), Neal and Xiang (2017) and Lee and Neal (2018). In these papers multiple infection times are updated within a single iteration of the algorithm, with favourable results.

3.3.2.2 Step 2(b): Update the Unobserved Infectives

In this section we consider updating the occult individuals, who are infected but not removed when we consider the outbreak. We note that the current occults are those within $\mathcal{I}_t$, and therefore they have inferred infection and removal times for each sample. There are two ways to change the current information about the occult individuals: we could change the infection and removal times for those already inferred to be currently infectious, or we could choose to add or remove individuals from the set of occults.

The updating of the occult individuals will share many similarities with the updating of those who have been observed to be infectious. The key difference is that we require the number of individuals to change, therefore we choose to update the unobserved infectious individuals in two ways:

Step 2(b): Update the occult data, denoted by $\left\{\boldsymbol{i}^{I}_{\tau:t},\ \boldsymbol{r}^{I}_{0:t}\right\}$.

(i) Update the infection and removal times of those currently inferred to be in state $\mathcal{I}_t$.

(ii) Update which individuals are in $\mathcal{I}_t$, this will require the addition or deletion of some of the infection and removal times.

The first step does not require the dimension of the space to change and will closely follow the updating of the observed individuals described in the previous section. The second proposal step is more complicated, requiring the individuals proposed to be infectious at time t to change. We will discuss both of these updating steps next.

Step 2b(i) Changing the Times

The first update we propose is to change the infection times of those currently estimated to be occults. For each individual currently within state $\mathcal{I}_t^{(j)}$ $\left(=\mathcal{I}_t^{(j+1)}\right)$ we propose an update to their infection time with probability $1/\mu_{o_1}$, where μ_{o_1} is the tuning factor for this proposal step. The value of μ_{o_1} relates to the proportion of the occult times we change, on average, in each iteration of the MCMC. We could instead choose to update a fixed number, similar to the previous updating step, however, we found this to be more appropriate due to the value of u_t (and therefore the dimension of the augmented data) changing.

As the occult individuals are infectious at time t we need to not only generate their infectious period, but also where this lies with respect to time t. We therefore define the proposal step as follows:

1. For each $\ell \in \mathcal{I}_t^{(j)}$
   - (a) Set $h_t^{\ell,*} = h_t^{\ell,(j)}$, $i_t^{\ell,*} = i_t^{\ell,(j)}$ and $r_t^{\ell,*} = r_t^{\ell,(j)}$.
   - (b) Generate $x \sim U(0,1)$.
   - (c) If $x \leq \frac{1}{\mu_{o_1}}$
     - (i) Draw $h_t^{\ell,*} \sim g_H$.
     - (ii) Draw how far through their infectious period the individual is, denoted by $a^{\ell,*}$, from the set $\left\{0, \ldots, (h_t^{\ell,*} - 1)\right\}$.
     - (iii) Set $i_t^{\ell,*} = t - a^{\ell,*}$ and $r_t^{\ell,*} = i_t^{\ell,*} + h_t^{\ell,*}$.
2. Let $\boldsymbol{i}_{\tau:t}^{I,*} = \left\{i_t^{\ell,*} : \ell \in \mathcal{I}_t^{(j)}\right\}$, $\boldsymbol{r}_{0:t}^{I,*} = \left\{r_t^{\ell,*} : \ell \in \mathcal{I}_t^{(j)}\right\}$.

We see that we only change the information relating to the individuals already inferred to be occults.

For this proposal, if the new times are consistent, we have acceptance probability of the form

$$\alpha = \min\left(1, \frac{\pi\left(\boldsymbol{\theta}^{(j+1)}, \boldsymbol{i}_{\tau:t}^{R,(j+1)}, \boldsymbol{i}_{\tau:t}^{I,*}, \boldsymbol{r}_{0:t}^{I,*} \mid \boldsymbol{r}_{0:t}^{R}\right)}{\pi\left(\boldsymbol{\theta}^{(j+1)}, \boldsymbol{i}_{\tau:t}^{R,(j+1)}, \boldsymbol{i}_{\tau:t}^{I,(j)}, \boldsymbol{r}_{0:t}^{I,(j)} \mid \boldsymbol{r}_{0:t}^{R}\right)} \prod_{\ell \in \mathcal{I}_t^{(j)}} \frac{g_H\left(h_t^{\ell,(j)} ; \boldsymbol{\theta}^{(j+1)}\right) h_t^{\ell,*}}{g_H\left(h_t^{\ell,*} ; \boldsymbol{\theta}^{(j+1)}\right) h_t^{\ell,(j)}}\right). \tag{3.3.5}$$

If accepted, we set

$$\boldsymbol{i}_{\tau:t}^{I,(j+1)} = \boldsymbol{i}_{\tau:t}^{I,*}, \quad \boldsymbol{r}_{0:t}^{I,(j+1)} = \boldsymbol{r}_{0:t}^{I,*}.$$

If rejected, we set

$$\boldsymbol{i}_{\tau:t}^{I,(j+1)} = \boldsymbol{i}_{\tau:t}^{I,(j)}, \quad \boldsymbol{r}_{0:t}^{I,(j+1)} = \boldsymbol{r}_{0:t}^{I,(j)}.$$

As previously, by changing the times we are implicitly changing the indexes in each state, $\mathcal{S}_f^{(j+1)}/\mathcal{I}_f^{(j+1)}$, for each $f \leq t$, where again due to this form of proposal we still have $\mathcal{I}_t^{(j)} = \mathcal{I}_t^{(j+1)}$ and $\mathcal{S}_t^{(j)} = \mathcal{S}_t^{(j+1)}$.

Similarly to μ_i, the value for the tuning parameter, μ_{o_1}, will be selected to achieve a reasonable acceptance rate (again around 25%), this will be further discussed in Section 3.3.6.

Step 2b(ii) Adding and Deleting Occults

For the second stage of updating the occult data we choose to change the number of currently infectious individuals, either by increasing or decreasing the number of them. We begin by drawing c from the set $\{-\mu_{o_2}, \ldots, -1, 1, \ldots, \mu_{o_2}\}$, where similar to before μ_{o_2} is a tuning parameter, the value of which will be discussed in Section 3.3.6. If $0 \leq c \leq m_t^{S,(j)}$ then we add c occults individuals, if $0 \leq -c \leq u_t^{(j)}$ then we remove $-c$ occult individuals.

- In the case of $c > 0$ we select uniformly at random c individuals from the set of $m_t^{S,(j)}$ susceptible individuals: denoted $F \subseteq \mathcal{S}_t^{(j)}$, such that $|F| = c$. These will become the new occult individuals. Then for each $\ell \in F$ we generate new infection and removal times using the same method as seen when changing the occults in the previous updating step, part (b)(i).

- In the case of $c < 0$, we select at random a set $F \subseteq \mathcal{I}_t^{(j)}$, such that $|F| = -c$. We then remove the individuals within F from the set of occults, setting them as susceptible individuals.

We therefore have new proposed values, $u_t^* = u_t^{(j)} + c$ and $m_t^{S,*} = m_t^{S,(j)} - c$, and update the times accordingly. If the proposal is consistent then the acceptance probability, α, takes the form:

if $c > 0$:

$$\min\left(1, \frac{\pi\big(\boldsymbol{\theta}^{(j+1)},\, \boldsymbol{i}_{\tau:t}^{R,(j+1)},\, \boldsymbol{i}_{\tau:t}^{I,*},\, \boldsymbol{r}_{0:t}^{I,*} \mid \boldsymbol{r}_{0:t}^{R}\big)}{\pi\big(\boldsymbol{\theta}^{(j+1)},\, \boldsymbol{i}_{\tau:t}^{R,(j+1)},\, \boldsymbol{i}_{\tau:t}^{I,(j+1)},\, \boldsymbol{r}_{0:t}^{I,(j+1)} \mid \boldsymbol{r}_{0:t}^{R}\big)} \binom{m_t^{S,(j)}}{c} \binom{u_t^*}{c}^{-1} \prod_{\ell \in F} \frac{h_\ell^*}{g_H\left(h_\ell^*; \boldsymbol{\theta}^{(j+1)}\right)}\right), \tag{3.3.6}$$

if $c < 0$:

$$\min\left(1, \frac{\pi\big(\boldsymbol{\theta}^{(j+1)},\, \boldsymbol{i}_{\tau:t}^{R,(j+1)},\, \boldsymbol{i}_{\tau:t}^{I,*},\, \boldsymbol{r}_{0:t}^{I,*} \mid \boldsymbol{r}_{0:t}^{R}\big)}{\pi\big(\boldsymbol{\theta}^{(j+1)},\, \boldsymbol{i}_{\tau:t}^{R,(j+1)},\, \boldsymbol{i}_{\tau:t}^{I,(j+1)},\, \boldsymbol{r}_{0:t}^{I,(j+1)} \mid \boldsymbol{r}_{0:t}^{R}\big)} \binom{u_t^{(j)}}{c} \binom{m_t^{S,*}}{c}^{-1} \prod_{\ell \in F} \frac{g_H\Big(h_\ell^{(j+1)}; \boldsymbol{\theta}^{(j+1)}\Big)}{h_\ell^{(j+1)}}\right). \tag{3.3.7}$$

Although the dimension is changing the Jacobian is just the identity matrix, thus this simplifies to the Metropolis-Hastings acceptance probability (as shown in Section 3.3.5).

As before we update $\mathcal{S}_f/\mathcal{I}_f$ for $f \leq t$ to reflect the update, as well as updating $\boldsymbol{i}^{I,(j+1)}_{\tau:t}$ and $\boldsymbol{r}^{I,(j+1)}_{\tau:t}$, as required.

We perform both of these occult updating steps within a single iteration of the MCMC algorithm, as it will allow the chain to effectively explore the sample space. Although there are many other choices of updating scheme, this is the method we shall use throughout. As in the previous steps, we will reserve discussion of μ_{o_2} to Section 3.3.6.

One of the key difficulties of using data-augmented MCMC is the updating of the unobserved data, specifically the information relating to the occult individuals. Many of the methods previously used when utilising MCMC methods in conjunction with epidemic data have been applied when an outbreak has been completed, thus there is complete information about who was infected. Our method is to be used whilst an outbreak is still ongoing; therefore, as well as the infection times of the observed individuals, we need to propose updates for the occults. The method we have proposed, of splitting the occult step into two, is similar to that used by Jewell et al. (2009), who treated the movement of times and the addition and deletion separately. However, our method is unique in the further splitting the updating of the observed individuals and the occults into two separate steps.

3.3.3 Summary of the MCMC Steps

In summary, the MCMC we have constructed uses the following updating scheme:

Step 1. Update $\boldsymbol{\theta} \,|\, \boldsymbol{y}_{\tau:t},\, \boldsymbol{x}_{0:t}$

Step 2. Update $\boldsymbol{y}_{\tau:t} \,|\, \boldsymbol{\theta},\, \boldsymbol{x}_{0:t}$

(a) Update the observed infectives data, denoted by $\boldsymbol{i}^{R}_{\tau:t}$.

(b) Update the unobserved infectives data, denoted by $\left\{\boldsymbol{i}^{I}_{\tau:t},\, \boldsymbol{r}^{I}_{0:t}\right\}$.

(i) Change the infection and removal times of the current occults.

(ii) Change who the occult individuals are.

Step 3. Return to Step 1.

3.3.4 A Note on the Removal Times

We will briefly highlight a key point about the removal times relating to the occult individuals, denoted $\boldsymbol{r}^I_{0:t}$. We could instead choose to not infer these and instead work with the conditional distribution. Put simply if at time t we have an occult individual with infection time z and removal time w then, rather than working with $P(H = w - z)$, we could instead calculate $P(H > t - z)$, where H is a random variable denoting the infectious period.

This would avoid the need to infer the removal times, only requiring inference of the infection times. However, due to the partially observed nature of our problem, we have found in practice that when we do not infer the removal times the MCMC algorithm can have severe convergence issues. Therefore, as MCMC methods work well with data augmentation, we choose to additionally infer the removal times of the occult individuals. This will not be the case with the SMC algorithm as this does not have the same convergence issues that MCMC algorithms have.

3.3.5 Satisfying the Detailed Balance Condition

In Section 1.4.7 we described the dimension matching condition that ensured the detailed balance condition is satisfied, when proposing steps which change the dimension of the space we are considering. We briefly check here that the proposal step to add or remove occult individuals satisfies this condition. We also illustrate why the acceptance probability collapses down to the familiar Metropolis-Hastings formula.

Recall that we denote the unobserved data by $\boldsymbol{y}_{\tau:t} = \left\{\boldsymbol{i}^R_{\tau:t},\, \boldsymbol{i}^I_{\tau:t},\, \boldsymbol{r}^I_{0:t}\right\}$, therefore we can define the size of $\boldsymbol{y}_{\tau:t}$ to be

$$\left|\boldsymbol{y}_{\tau:t}\right| = \left|\boldsymbol{i}^R_{\tau:t}\right| + \left|\boldsymbol{i}^I_{\tau:t}\right| + \left|\boldsymbol{r}^I_{0:t}\right| = m^R_t + u_t + u_t. \tag{3.3.8}$$

This is as for those with an observed removal time we only need to infer their infection time, however, for the occults we must infer both their infection and removal times.

We consider the move between dimensions which occurs when we add $c > 0$ individuals to the set of occults. We denote the two proposed models as

$$\mathcal{M}_1 : \phi^{(1)} = \boldsymbol{y}^{(1)}_{\tau:t} \quad \text{such that} \quad \left|\boldsymbol{y}^{(1)}_{\tau:t}\right| = m^R_t + 2u^{(1)}_t,$$

$$\mathcal{M}_2 : \phi^{(2)} = \boldsymbol{y}^{(2)}_{\tau:t} \quad \text{such that} \quad \left|\boldsymbol{y}^{(2)}_{\tau:t}\right| = m^R_t + 2u^{(2)}_t = m^R_t + 2u^{(1)}_t + 2c,$$

where each model has a specific configuration of individuals infected, such that moves between the two models are possible using the proposal scheme described previously. We define $v^{(1)} = \left(v^{(1)}_I, \, v^{(1)}_R\right)$ as the new times, infection and removal respectively, generated using the proposal distribution required to move from $\mathcal{M}_1$ to $\mathcal{M}_2$. Therefore,

$$\phi^{(2)} = \left(\phi^{(1)}, \, v^{(1)}\right) = \left(\phi^{(1)}, \, v^{(1)}_I, \, v^{(1)}_R\right) \tag{3.3.9}$$

with $\left|v^{(1)}_I\right| = c = \left|v^{(1)}_R\right|$. We can easily see that the dimension matching condition is satisfied:

$$\left|v^{(1)}\right| + \left|\phi^{(1)}\right| = 2c + m^R_t + 2u^{(1)}_t = m^R_t + 2u^{(2)}_t = \left|v^{(2)}\right| + \left|\phi^{(2)}\right|, \tag{3.3.10}$$

as $v^{(2)} = \emptyset$. Additionally, we can define the bijection between the two subspaces as

$$Y^{(1)}\left(\phi^{(1)}, \, v^{(1)}\right) = Y^{(1)}\left(\boldsymbol{y}^{(1)}_{\tau:t}, \, v^{(1)}\right) = \boldsymbol{y}^{(2)}_{\tau:t} = \phi^{(2)} \tag{3.3.11}$$

and therefore we can determine the Jacobian,

$$J = \frac{\partial Y^{(1)}\left(\phi^{(1)}, \, v^{(1)}\right)}{\partial\left(\phi^{(1)}, \, v^{(1)}\right)} = \frac{\partial\left(\phi^{(2)}\right)}{\partial\left(\phi^{(1)}, v^{(1)}\right)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{3.3.12}$$

with $|J| = 1$. The reverse move from $\mathcal{M}_2 \to \mathcal{M}_1$ also has $|J| = 1$, thus the acceptance probability reduces down to the familiar form of a Metropolis-Hastings acceptance probability, with the included probability of moving from one model to another.

3.3.6 Acceptance Rate

We have discussed previously in Section 1.4.4.3 why it is desirable to have an acceptance rate close to 25% when using a random-walk Metropolis updating step. This result has been found to be highly robust to different forms of target distributions (see, Roberts and Rosenthal (2001)). As such for the parameter updating step described in Section 3.3.1, this appears to be a sensible aim.

For the updating of the infection times the justification is less clear, however, Lee and Neal (2018) found that this acceptance rate is close to optimal when updating infection times. This paper considered the algorithm used in Xiang and Neal (2014), focusing on a homogeneous SIR epidemic. Although our problem is slightly different from that in this paper, we have found this acceptance rate produces satisfying results. As such we aim to achieve it within each of the proposal steps in the MCMC algorithm described.

3.3.6.1 Adaptive Tuning

Now that we have decided what our ideal acceptance rate is, we need to provide a method of achieving it. Unfortunately it can be difficult to immediately construct an algorithm that produces such an acceptance rate, especially when we have very little data. For these reasons we choose to adaptively tune the MCMC algorithm we have constructed.

Suppose that we run an MCMC with burn-in b; we choose to adaptively tune between iterations b_1 and b_2, where $0 < b_1 < b_2 < b$. This ensures that we will always satisfy the conditions required for the MCMC to converge. We have previously discussed in Section 1.4.4.4 how to adaptively tune a random-walk Metropolis algorithm, which is the form of the proposal we use to update the parameters. As such for the parameter proposal step we refer the reader to Section 1.4.4.4 for a general method and Chapter 4 for the specifics of an adaptive method, as applied to a simulated data set. In this section we instead focus on selecting the tuning parameters relating to the updating of $\boldsymbol{y}_{\tau:t}$.

Recall that $\boldsymbol{y}_{\tau:t}$ contains the unobserved infection times, as well as the removal times of the occult individuals. As such we updated it with three steps: updating the observed infectives, changing the occult individuals' infection and removal times and changing the number of occult individuals. Specifically we introduced three tuning parameters:

- μ_i, the number of observed individuals we propose to update in each iteration (see, Section 3.3.2.1).

- $\frac{1}{\mu_{o_1}}$, the probability we propose an update for each occult individual (see, Section 3.3.2.2).

- μ_{o_2}, the range for how many occults we propose to add/delete in each iteration (see, Section 3.3.2.2).

The method we use to tune is fairly crude, however, it will achieve a reasonable acceptance rate if performed over a sufficient number of iterations. For each of the updating steps we define the acceptance rates as AR_i, AR_{o_1} and AR_{o_2}, respectively. In general, to ensure we are monitoring the change in acceptance rate, we only consider the last x iterations when considering if the acceptance rate has drifted too low or too high. Then in every k^{th} iteration before b_2, for $k \in \mathbb{N}$, we adaptively tune by considering the following conditions.

$$\begin{array}{llll}
\text{If } AR_i < l_- & \text{and} \quad \mu_i > 1 & \text{then let} & \mu_i \longrightarrow \mu_i - 1. \\
\text{If } AR_i > l_+ & \text{and} \quad \mu_i < m_t^R & \text{then let} & \mu_i \longrightarrow \mu_i + 1. \\
\\
\text{If } AR_{o_1} < l_- & \text{and} \quad \mu_{o_1} < U_1 & \text{then let} & \mu_{o_1} \longrightarrow \mu_{o_1} + 1. \\
\text{If } AR_{o_1} > l_+ & \text{and} \quad \mu_{o_1} > 1 & \text{then let} & \mu_{o_1} \longrightarrow \mu_{o_1} - 1. \\
\\
\text{If } AR_{o_2} < l_- & \text{and} \quad \mu_{o_2} > 1 & \text{then let} & \mu_{o_2} \longrightarrow \mu_{o_2} - 1. \\
\text{If } AR_{o_2} > l_+ & \text{and} \quad \mu_{o_2} < U_2 & \text{then let} & \mu_{o_2} \longrightarrow \mu_{o_2} + 1.
\end{array}$$

We set l_- and l_+ to be the acceptance rate we are aiming between. Throughout we will choose $l_\pm = 0.25 \pm 0.15$, as we find in general that this choice performs well. The values of U_1 and U_2 are not always required, but they can be selected if we wish to restrict the size of the proposal jumps. Throughout, unless stated otherwise, we adaptively tune every 100^{th} iteration, based on acceptance rate over the last 100 iterations. This is in contrast to the adaptive RWM, which recomputes the covariance matrix in each iteration (within the tuning period), using all of the samples generated thus far.

3.3.7 Conclusions

We have constructed a flexible MCMC algorithm, which will explore the posterior distribution produced by the epidemic data effectively. Using such an MCMC we can easily generate n samples, denoted

$$\left\{\left(\boldsymbol{\theta}^{(j)},\, \boldsymbol{y}^{(j)}_{\tau:t}\right) : j = 1, \ldots, n\right\} = \left\{\left(\boldsymbol{\theta}^{(j)},\, \left\{\boldsymbol{i}^{R,(j)}_{\tau:t},\, \boldsymbol{i}^{I,(j)}_{\tau:t},\, \boldsymbol{r}^{I,(j)}_{0:t}\right\}\right) : j = 1, \ldots, n\right\}, \tag{3.3.13}$$

from the posterior distribution at time t.

This MCMC algorithm has been constructed to be

- Flexible: we have aimed to define as little of the specifics of the epidemic as possible to ensure that the MCMC can be applied to a wide range of outbreaks, incorporating a variety of behaviour as desired.
- Efficient: we have constructed an MCMC algorithm which can effectively explore the space on which the posterior distribution lies. In particular we have introduced three steps to update the augmented data, which are applied in each iteration.
- Optimal: we have incorporated various measures to ensure we are achieving an optimal acceptance rate. This is through utilising adaptive random walk methods to efficiently update the parameters, as well as a method of adaptively selecting the tuning parameters.

This MCMC algorithm will act as the current ‘gold-standard’ against which we shall compare the performance of the SMC algorithm. In the next section we consider the construction of the SMC algorithm that will use these samples and iteratively update them as new information is obtained.

3.4 The SMC Algorithm

The particles generated using the MCMC algorithm (and the data obtained up to time t) represent a sample from the posterior distribution at time t. However, we have assumed that the epidemic is still in progress and therefore at time $t+1$ we will receive new data. We could choose to repeatedly apply the constructed MCMC, from scratch, each time we observed any new data. Alternatively, in this section we aim to take the particles generated at time t forward in time and use them to inform us about the posterior distribution at time $t+1$.

We will achieve this by adapting the sequential Monte Carlo methods described in Section 1.6 to epidemic data. We choose to use a similar structure to the sequential-importance-resampling-and-move algorithm discussed in Section 1.6.3. This form of algorithm has the following general structure:

Step 1. Generate n particles.

Step 2. Obtain the new data.

Step 3. Augment the particles.

Step 4. Calculate the weight of each particle.

Step 5. Resample the particles (optional).

Step 6. Move the particles (optional).

Step 7. Return to Step 2.

If the weight of a particle (step 4) is independent of the newly sampled part of the particle (step 3) then we may choose to perform the weight and resampling steps before the augmentation step. This will allow a greater amount of diversity to be incorporated into the particles (Doucet and Johansen (2011)).

In this section we will discuss how to adapt the underlying ideas within the sequential Monte Carlo algorithms such that they can be applied to infectious disease outbreak data. We will find that many of the steps are not straightforward to apply and we must make some concessions in order to use this method in conjunction with epidemic data.

3.4.1 Generating the Initial Particles

The first step in the SMC algorithm is to generate n initial particles, which we can then reweight and resample as we acquire new data. We will sample the initial particles from the posterior distribution at time T, $\pi_T = \pi(\boldsymbol{\theta},\, \boldsymbol{y}_{\tau:T} \,|\, \boldsymbol{x}_{0:T})$, where $T \geq \tau$. To generate the initial samples we can utilize one of the simulation methods discussed previously in Chapter 1. We choose to use MCMC methods (Section 1.4) to generate the initial particles as they are proven to be highly flexible, with the ability to handle large amounts of augmented data well. Additionally, with not many infectives (small T) we can apply the MCMC algorithm relatively quickly. Although we will have less data, and therefore greater uncertainty, there are also fewer infectious individuals. It is at the peak of the outbreak, when there are many occult individuals, and at the end of the outbreak, with a large amount of data, that the MCMC algorithm can take a significant amount of time to converge. This is discussed later in Section 4.5 where we compare the time it takes to apply the MCMC and SMC algorithms.

The choice of T will be problem-dependent; however, we will often use the idea of Liu and Chen (1995) and generate the initial particles when we have obtained some data. This should aid in reducing the particle degeneracy and is realistic, as usually we will not begin analysis until we have received some reported cases.

The MCMC algorithm will follow that described in Section 3.3. Throughout we will generate $n = 1000$ particles as this is more than sufficient for most problems, however, in high dimensional problems we may require a larger n. Additionally, the number of particles required will depend on the practical aspects of the particular research question. For example, if we are only concerned with specific parameters we may require fewer particles. Note: as we choose to use a finite number of samples we will often thin the MCMC, thereby reducing the autocorrelation between the particles we take forward.

We denote the initial particles as

$$\left\{\left(\boldsymbol{\theta}^{(j)},\ \boldsymbol{y}_{\tau:T}^{(j)}\right) : j = 1, \ldots, n\right\} = \left\{\left(\boldsymbol{\theta}^{(j)},\ \left\{\boldsymbol{i}_{\tau:T}^{R,(j)},\ \boldsymbol{i}_{\tau:T}^{I,(j)},\ \boldsymbol{r}_{0:T}^{I,(j)}\right\}\right) : j = 1, \ldots, n\right\}, \tag{3.4.1}$$

these represent samples from the posterior distribution at time T.

3.4.2 Incorporating the New Data

As mentioned previously we are interested in what happens on the next day, when we receive new information in the form of new individuals being removed.

The question we are interested in is, can we use the n samples generated at time T (represented in (3.4.1)) to inform us about the posterior distribution at time $T + 1$, which incorporates all of the data we now have access to? We begin by considering the relationship between the posterior distributions at time T and at time $T + 1$. We discard the samples, $\boldsymbol{r}_{0:T}^{I}$ (the removal times of the occult individuals), as these were only generated to ensure the MCMC converged. We reserve full explanation as to why we discard these samples to later sections, where it will become immediately clear.

Therefore, the unobserved data now only contains the infection times, $\boldsymbol{y}_{\tau:T} = \boldsymbol{i}_{\tau:T} = \{\boldsymbol{i}^R_{\tau:T},\ \boldsymbol{i}^I_{\tau:T}\}$, and, as before, we have observed those who have been removed at or before time T, $\boldsymbol{x}_{0:T} = \boldsymbol{r}_{0:T} = \boldsymbol{r}^R_{0:T}$.

We assume that at time $T+1$ we observe some new data, $\boldsymbol{r}_{T+1}$, we therefore are now interested in

$$\pi(\boldsymbol{\theta},\ \boldsymbol{i}_{\tau:T+1},\,|\,\boldsymbol{r}_{0:T+1}) \quad (=\pi(\boldsymbol{\theta},\ \boldsymbol{y}_{\tau:T+1}\,|\,\boldsymbol{x}_{0:T+1})\,), \tag{3.4.2}$$

the posterior distribution at time $T+1$. Considering this distribution and the analogous distribution at time T we have the following relationship

$$\pi(\boldsymbol{\theta},\ \boldsymbol{i}_{\tau:T+1}\mid\boldsymbol{r}_{0:T+1}) \;=\; \frac{\pi(\boldsymbol{r}_{T+1},\ \boldsymbol{i}_{T+1}\mid\boldsymbol{\theta},\ \boldsymbol{i}_{\tau:T},\ \boldsymbol{r}_{0:T})\,\pi(\boldsymbol{\theta},\ \boldsymbol{i}_{\tau:T}\mid\boldsymbol{r}_{0:T})}{\pi(\boldsymbol{r}_{T+1}\mid\boldsymbol{r}_{0:T})}. \tag{3.4.3}$$

We can note that the new infections and removals on day $T+1$ are independent, given $\{\boldsymbol{\theta},\ \boldsymbol{i}_{\tau:T},\ \boldsymbol{r}_{0:T}\}$, and therefore

$$\pi(\boldsymbol{r}_{T+1},\ \boldsymbol{i}_{T+1}\,|\,\boldsymbol{\theta},\ \boldsymbol{i}_{\tau:T},\ \boldsymbol{r}_{0:T}) \;=\; \pi(\boldsymbol{r}_{T+1}\,|\,\boldsymbol{\theta},\ \boldsymbol{i}_{\tau:T},\ \boldsymbol{r}_{0:T}) \times \pi(\boldsymbol{i}_{T+1}\,|\,\boldsymbol{\theta},\ \boldsymbol{i}_{\tau:T},\ \boldsymbol{r}_{0:T}). \tag{3.4.4}$$

Thus, using (3.4.4) we can rewrite (3.4.3) as

$$\begin{aligned}
\pi(\boldsymbol{\theta},\ \boldsymbol{i}_{\tau:T+1}\mid\boldsymbol{r}_{0:T+1}) \quad\propto\quad & \pi(\boldsymbol{\theta},\ \boldsymbol{i}_{\tau:T}\mid\boldsymbol{r}_{0:T}) && \text{(3.4.5a)}\\
& \times\ \pi(\boldsymbol{r}_{T+1}\mid\boldsymbol{\theta},\ \boldsymbol{i}_{\tau:T},\ \boldsymbol{r}_{0:T}) && \text{(3.4.5b)}\\
& \times\ \pi(\boldsymbol{i}_{T+1}\mid\boldsymbol{\theta},\ \boldsymbol{i}_{\tau:T},\ \boldsymbol{r}_{0:T}). && \text{(3.4.5c)}
\end{aligned}$$

(3.4.5) shows that the posterior distribution at time $T+1$ can been deconstructed into the posterior distribution at time T, (3.4.5a), multiplied by two additional terms ((3.4.5b) and (3.4.5c)). Considering each term in (3.4.5) individually we can understand this breakdown better:

Equation 3.4.5a: $\pi(\boldsymbol{\theta},\, \boldsymbol{i}_{\tau:T} \,|\, \boldsymbol{r}_{0:T})$

The posterior distribution at time T, from which we generated the original n samples (see, Section 3.4.1).

Equation 3.4.5b: $\pi(\boldsymbol{r}_{T+1} \,|\, \boldsymbol{\theta},\, \boldsymbol{i}_{\tau:T},\, \boldsymbol{r}_{0:T})$

This represents the chance that we witness this new data given our previous inference and will be the weighting of each particle (see, Section, 3.4.6).

Equation 3.4.5c: $\pi(\boldsymbol{i}_{T+1} \,|\, \boldsymbol{\theta},\, \boldsymbol{i}_{\tau:T},\, \boldsymbol{r}_{0:T})$

This part of the equation involves the new infections that occur at time $T+1$, which will be unobserved. However, for each particle, we can easily generate values from this distribution, as illustrated in Section 3.4.7. This can then be added to the samples previously generated at time T and acts as the 'augmentation' step of the SMC algorithm (see Chapter 1).

In summary we can see that if we take our original samples, generated at time T, reweight them and generate the new infections occurring at time $T+1$ then we will have samples from the posterior distribution at time $T+1$. We can immediately see how this matches the SMC methods discussed in Chapter 1.

As discussed previously if the weight and augmentation steps are independent, then we can swap these steps around, allowing for greater diversity within the samples. We observed in (3.4.4) that this is the case here. This is to be expected as the new infections at time $T+1$ will clearly be independent of those removed at time $T+1$. As such we will choose to generate the new infections after we have weighted and resampled the particles (see Section 3.4.7). Thus the order of the steps we perform is now:

Step 1. Generate n particles.

Step 2. Obtain the new data.

Step 3. Calculate the weight of each particle.

Step 4. Resample the particles.

Step 5. Augment the particles.

Step 6. Move the particles.

Step 7. Return to Step 2.

3.4.3 A Problem with the Weights

Returning to the SMC, as we have generated the initial particles the next step is to calculate the weight of each, $w_{T+1}^{(j)}$ for $j = 1, \ldots, n$, as defined in (3.4.5b):

$$w_{T+1}^{(j)} = \pi\big(\boldsymbol{r}_{T+1} \,|\, \boldsymbol{\theta}^{(j)},\, \boldsymbol{i}_{\tau:T}^{(j)},\, \boldsymbol{r}_{0:T}\big). \tag{3.4.6}$$

This will have contribution from those individuals whose infectious period has ended and those whose infectious period is continuing:

$$\begin{aligned}\pi(\boldsymbol{r}_{T+1} \,|\, \boldsymbol{\theta},\, \boldsymbol{i}_{\tau:T},\, \boldsymbol{r}_{0:T}) = &\prod_{\ell \in \mathcal{V}_{T+1}} P\big(H = (T+1) - i_T^{\ell} \mid H > T - i_T^{\ell}\big) \\ &\times \prod_{\ell \in \mathcal{I}_T \setminus \mathcal{V}_{T+1}} P\big(H > (T+1) - i_T^{\ell} \mid H > T - i_T^{\ell}\big),\end{aligned} \tag{3.4.7}$$

where H is a random variable with probability mass function g_H and $\mathcal{V}_{T+1} = \mathcal{R}_{T+1} \setminus \mathcal{R}_T$ are those newly infected at time $T+1$.

If we consider the form of the weight more closely we can immediately observe a problem. Consider a particle, j, sampled at time T. Within this particle we will have inferred which individuals were infectious at time T, ensuring that this was consistent with the currently observed data. At time $T+1$ we will witness new individuals being removed who were infectious at time T: this is the form of the new data that we wish to incorporate. However, if particle j has not predicted that all those removed at time $T+1$ were infectious at time T then that particle will be inconsistent with the new information. This is due to the key assumption of our model that individuals must progress through the states in the order S → I → R. If an individual is removed without having an infection time it would violate this key model assumption.

Formally, we have new data denoted by $\boldsymbol{r}_{T+1}$, this contains the indexes and removal times of those newly removed at time $T+1$. We are interested in evaluating (3.4.7), the conditional probability of observing these removals given the infection times of those infectious at time T ($\boldsymbol{i}_{\tau:T}^{I}$, contained within $\boldsymbol{i}_{\tau:T}$). However, if an individual whose time is contained within $\boldsymbol{r}_{T+1}$ does not have a time within $\boldsymbol{i}_{\tau:T}^{I}$ then this conditional probability cannot be computed.

In summary, (3.4.7) describes the probability that certain individuals are removed, conditional on the data up to the previous time-step. If there are individuals whom the particle never inferred to be infected then this particle will be given a weight of zero. This is due to it being incompatible with the newly observed data and thus in the resampling step it will be discarded. An example of this problem is described below.

Example Part I: a problem with the new data

Consider a population consisting of individuals $\{A, B, C, D, E, F, G, H, I, J\}$. Additionally, assume that we have observed the outbreak up until time T, with $\mathcal{R}_T = \{J\}$. As a result one possible (consistent) particle is j, with:

$$\mathcal{S}_T^{(j)} = \{F, G, C, D\} \qquad \mathcal{I}_T^{(j)} = \{E, A, B, H, I\} \qquad \mathcal{R}_T = \{J\}\,. \tag{3.4.8}$$

However, suppose that at time $T+1$ we witness $\mathcal{R}_{T+1} = \{J, E, F, G\}$ and therefore the new observations are $\mathcal{V}_{T+1} = \{E, F, G\}$. This is not consistent with the previously defined particle as individuals F and G were inferred to be susceptible at time T. Consequently this particle is inconsistent with the newly observed data and would be rejected (zero probability of being resampled).

As the size of the population we are considering increases the chance of any particle fully matching the newly observed data will shrink. This is especially true if we have many new observations on each day. Therefore, using the current method, we would suffer from mass particle degeneration as we would discard most of the samples in this way. However, we do not expect any inference drawn from the inconsistent particles to be poor, even if they do not perfectly match the new data. With this in mind we are motivated to find an alternative method that does not suffer from a mass loss of unique samples.

3.4.4 Producing Consistent Particles

One option, to avoid this particle degeneracy, is to adjust the samples so that they are consistent with the newly observed data. If we can achieve this in such a way as to not significantly alter the samples then this may be preferable to the mass loss of unique particles.

The procedure we adopt maintains the number of occults present at time T, in each of the particles, as well as keeping the infection times generated within each particle the same. In this way we aim to change each particle as little as possible. In this section we will drop the superscript defining which sample we are working with, however, it should be remembered that this adjustment will be performed on each particle independently.

Suppose that we have an individual, k, who is newly removed at time $T+1$ but who was not inferred to be infectious within our particle generated at time T. To amend this particle so that it is consistent with this new information we choose at random an individual, ℓ, from the set of infectious individuals at time T who remain infectious at time $T+1$, i.e. they are not newly removed. We then let individual k take the place of individual ℓ. Therefore, if individual ℓ is inferred to be infected at time i^{ℓ}, then this becomes the time at which k becomes infected and ℓ is now assumed to be susceptible through to time T.

We repeat this process for each newly removed individual that is inconsistent, until the particle is compatible with the new data. If we have multiple new observations which are inconsistent, we adjust them in a random order. We can apply this adjustment as long as $v_{T+1} \leq u_T$ and we provide greater explanation of how individuals move between the states in Appendix A.2.

We will apply this adjustment to each particle, independently, once completed we will refer to the new samples as the *adjusted particles* and we will differentiate between the original and the adjusted samples by placing a '$\sim$' on the values and equations relating to the adjusted samples.

As we have not changed the number of individuals in each of the three states (S, I or R), we do not need to change the values of m_t^S or m_t^I, at any time point, t. We describe an example of this process next, where we continue the example given in the previous section.

Example Part II: a possible adjustment

Recall that the newly observed individuals are $\mathcal{V}_{T+1} = \{E, F, G\}$. Suppose that we have a particle with

$$\boldsymbol{i}_{\tau:T}^{I,(j)} = \left\{ \{v, w, x, y, z\} : \mathcal{I}_T^{(j)} = \{E, A, B, H, I\} \right\}, \tag{3.4.9}$$

where, for example, individual E is inferred to have infection time $i_T^E = v$. This particle has correctly guessed E was infectious at time T and thus E has an inferred infection time. However, individuals F and G have been, incorrectly, inferred as susceptible at time T.

One possible way of adjusting this particle so that it is consistent with the new data is

$$\tilde{\boldsymbol{i}}_{\tau:T}^{I,(j)} = \left\{ \{v, w, x, y, z\} : \tilde{\mathcal{I}}_T^{(j)} = \{E, A, G, F, I\} \right\}. \tag{3.4.10}$$

We have therefore allocated the infection time originally attached to individual B to individual G and the time attached to individual H to individual F. This is illustrated in Figure 3.1.

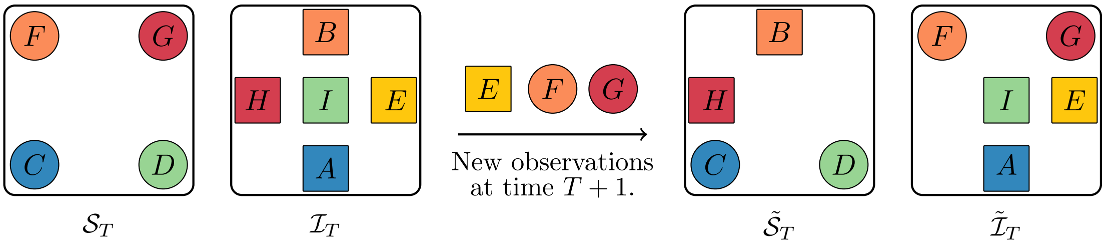


Figure 3.1: An illustration of the adjustment process applied to each particle, the circles and squares indicate different individuals. The left-hand side shows the original labelling of the particle and the right-hand side shows a possible amendment, which ensures that the particle is consistent with the new removals observed.

We apply this adjustment process to each of the n particles to obtain a set of samples that are fully consistent with the new information. There will be some loss incurred in this alteration as the particles we now have will only be approximations of samples from the true posterior distribution at time T. However, we propose that this slight loss in the accuracy of the particles will be less detrimental than the large loss that would otherwise be incurred in the resampling step. We denote the adjusted unobserved data as

$$\tilde{\boldsymbol{y}}_{\tau:T} = \tilde{\boldsymbol{i}}_{\tau:T} = \left\{ \tilde{\boldsymbol{i}}^{I}_{\tau:T},\ \boldsymbol{i}^{R}_{\tau:T} \right\} \tag{3.4.11}$$

and we define the set of adjusted particles by

$$\left\{ \left(\boldsymbol{\theta}^{(j)},\ \tilde{\boldsymbol{i}}^{(j)}_{\tau:T} \right)\ :\ j = 1, \ldots, n \right\}. \tag{3.4.12}$$

3.4.4.1 The Number of Possible Adjustments

Under the method of readjustment we have described, there are multiple ways to manipulate a particle so that it is consistent with the newly observed data. The question we are interested in is, how many possibilities are there? Understanding the number of possible adjustments is important for determining the impact such a transformation will have on the samples we have generated.

We begin by considering the example presented previously as motivation. Note that we define the number of orderings of x objects from y as the x-permutations of y, denoted by ${}^{y}P_{x} = \frac{y!}{(y-x)!}$.

Example Part III: all of the possible adjustments

Suppose that we have newly observed individuals $\mathcal{V}_{T+1} = \{E, F, G\}$ and have sampled particle j with

$$\boldsymbol{i}^{I,(j)}_{\tau:T} = \left\{ \{v, w, x, y, z\} : \mathcal{I}^{(j)}_{T} = \{E, A, B, H, I\} \right\}, \tag{3.4.13}$$

therefore we have incorrectly guessed the status of $c = \left| \mathcal{V}_{T+1} \backslash \mathcal{I}^{(j)}_{T} \right| = 2$ individuals. When adjusting we keep E constant, as this has been correctly guessed, we then have a choice of which individuals to swap for F and G. Thus we could correct this as

1. $\tilde{\boldsymbol{i}}^{I,(j)}_{\tau:T} = \left\{ \{v, w, x, y, z\}\ :\ \tilde{\mathcal{I}}^{(j)}_{T} = \{E, F, G, H, I\} \right\}$
2. $\tilde{\boldsymbol{i}}^{I,(j)}_{\tau:T} = \left\{ \{v, w, x, y, z\}\ :\ \tilde{\mathcal{I}}^{(j)}_{T} = \{E, F, B, G, I\} \right\}$
3. $\tilde{\boldsymbol{i}}^{I,(j)}_{\tau:T} = \left\{ \{v, w, x, y, z\}\ :\ \tilde{\mathcal{I}}^{(j)}_{T} = \{E, F, B, H, G\} \right\}$

4. $\tilde{\boldsymbol{i}}_{\tau:T}^{I,(j)} = \Big\{\{v, w, x, y, z\} : \tilde{\mathcal{I}}_T^{(j)} = \{E, G, F, H, I\}\Big\}$

5. $\tilde{\boldsymbol{i}}_{\tau:T}^{I,(j)} = \Big\{\{v, w, x, y, z\} : \tilde{\mathcal{I}}_T^{(j)} = \{E, A, F, G, I\}\Big\}$

6. $\tilde{\boldsymbol{i}}_{\tau:T}^{I,(j)} = \Big\{\{v, w, x, y, z\} : \tilde{\mathcal{I}}_T^{(j)} = \{E, A, F, H, G\}\Big\}$

7. $\vdots$

We can see that there will be ${}^4P_2 = \frac{4!}{2!} = 12$ ways we can transform $\boldsymbol{i}_{\tau:T}^{I,(j)}$ so that it is consistent with the new observations.

In general, suppose that the particle we are interested in has incorrectly guessed the status of c individuals, where $0 \leq c \leq v_{T+1}$ with v_{T+1} the number of new observations at time $T+1$. Then the number of possible amendments will be

$$^{u_T-(v_{T+1}-c)}P_c = \frac{(u_T - (v_{T+1} - c))!}{(u_T - v_{T+1})!}, \tag{3.4.14}$$

where u_T and $c = c\big(\boldsymbol{i}_{\tau:T}^I; \mathcal{V}_{T+1}\big)$ will depend on the specific particle.

This is as there are $\binom{u_T-(v_{T+1}-c)}{c}$ possible ways of selecting c individuals from those inferred to be infectious who were not removed, who can take the place of those c individuals we did not correctly guess. However, as each index is attached to a time the order matters and thus the number of possibilities is

$$\binom{u_T - (v_{T+1} - c)}{c} c! = \frac{(u_T - (v_{T+1} - c))!}{(u_T - v_{T+1})!}. \tag{3.4.15}$$

As we select the adjustment uniformly at random, the probability of adjusting a sample, $\boldsymbol{i}_{\tau:T}$, to adjusted sample, $\tilde{\boldsymbol{i}}_{\tau:T}$, will be

$$P\left(\boldsymbol{i}_{\tau:T} \longrightarrow \tilde{\boldsymbol{i}}_{\tau:T}\right) = \frac{(u_T - v_{T+1})!}{(u_T - (v_{T+1} - c))!}. \tag{3.4.16}$$

We can see in (3.4.14) that for each particle the number of alterations only depends on how many occult individuals are within that particle and how many of newly removed are not within the particle's set of occults. This is as the greater agreement the particle has with the truth, the less adjustment we need to apply to it. Additionally we expect there to be a greater number of possible adjustments if we have a greater number of occult individuals.

3.4.5 The Adjusted Posterior Distribution

We have adjusted the particles we sampled at time T so that they are consistent with the new data observed at time $T+1$. This appeared a necessary solution to the problem of particle degeneration. However, the cost of this is that the samples no longer represent samples from the true posterior distribution at time T. In this section we are interested in the distribution the adjusted samples come from.

Suppose that we have an adjusted sample, $\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T}\big)$, taken from the 'adjusted distribution', $\tilde{\pi}_T = \tilde{\pi}\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{r}_{0:T+1}\big)$. We expect this distribution will be similar to the original distribution, π_T, due to the nature of our adjustment, which aimed to limit the difference between the adjusted and true samples. To learn about the form of $\tilde{\pi}_T$ we must consider the updating regime we have implemented. We can first note that $\tilde{\pi}_T$ will satisfy

$$\tilde{\pi}\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{r}_{0:T+1}\big) \;=\; \sum_{\boldsymbol{a}} P\big(\boldsymbol{a} \longrightarrow \tilde{\boldsymbol{i}}_{\tau:T}\big)\, \pi\big(\boldsymbol{\theta},\, \boldsymbol{a} \,|\, \boldsymbol{r}_{0:T}\big), \tag{3.4.17}$$

where the summation is applied over all possible $\boldsymbol{a}$ which can be adjusted to $\tilde{\boldsymbol{i}}_{\tau:T}$ ($P\big(\boldsymbol{a} \longrightarrow \tilde{\boldsymbol{i}}_{\tau:T}\big) \neq 0$).

Equation (3.4.17) may appear as though it will be computationally intensive to calculate, however, as we found in the previous section, where we defined the adjustment regime, there are significant restrictions on which samples could have been adjusted to $\tilde{\boldsymbol{i}}_{\tau:T}$. For $\boldsymbol{a}$ to be adjusted to $\tilde{\boldsymbol{i}}_{\tau:T}$ it must satisfy the following:

- It must be the same size, such that $|\boldsymbol{a}| = |\tilde{\boldsymbol{i}}_{\tau:T}|$, i.e. the same value of u_T.

- It must contain the same infection times (not necessarily attached to the same individuals).

- The infection times and indexes of those individuals infectious at time T and time $T+1$ must match, the adjustment only relates to those removed at time $T+1$.

- The infection times relating to those in state R at time T match, this is as the adjustment scheme only relates to $\boldsymbol{i}^{I}_{\tau:T}$.

If we denote the set of occult indexes within $\boldsymbol{a}$ by $\mathcal{I}^{\boldsymbol{a}}_T$ then $c = |\mathcal{V}_{T+1} \backslash \mathcal{I}^{\boldsymbol{a}}_T|$ is the number of indexes the two sets differ by, as by construction $\tilde{\mathcal{I}}_{T+1} \subseteq \mathcal{V}_{T+1}$ (assuming

$u_T \geq v_{T+1}$). These differences must only lie within those newly removed at time $T+1$. Therefore the maximum for c is v_{T+1}. Additionally $\boldsymbol{a}$ cannot differ by more than m_T^S of the occult indexes due to the limited pool of susceptibles we could swap, however, in general v_{T+1} is much smaller than m_T^S.

We set $z_{T+1} = \min\left\{v_{T+1},\, m_T^S\right\}$, then we can write (3.4.17) as

$$\tilde{\pi}\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{r}_{0:T+1}\big) \;=\; \sum_{c=0}^{z_{T+1}} \sum_{\boldsymbol{a}_c} P\big(\boldsymbol{a}_c \;\longrightarrow\; \tilde{\boldsymbol{i}}_{\tau:T}\big)\, \pi(\boldsymbol{\theta},\, \boldsymbol{a}_c \,|\, \boldsymbol{r}_{0:T}), \tag{3.4.18}$$

where $\boldsymbol{a}_c$ differs from $\tilde{\boldsymbol{i}}_{\tau:T}$ by c indexes and $P\big(\boldsymbol{a}_c \longrightarrow \tilde{\boldsymbol{i}}_{\tau:T}\big) \neq 0$. To calculate $P\big(\boldsymbol{a}_c \longrightarrow \tilde{\boldsymbol{i}}_{\tau:T}\big)$ we can refer to (3.4.16), where we found

$$P\big(\boldsymbol{a}_c \;\longrightarrow\; \tilde{\boldsymbol{i}}_{\tau:T}\big) \;=\; \frac{(u_T - v_{T+1})!}{(u_T - (v_{T+1} - c))!}. \tag{3.4.19}$$

This is the reciprocal of the total number of possible adjustments we could make to this sample, as each occurs with equal probability. Returning to the adjusted distribution we now find

$$\tilde{\pi}\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{r}_{0:T+1}\big) \;=\; \sum_{c=0}^{z_{T+1}} \sum_{\boldsymbol{a}_c} \frac{(u_T - v_{T+1})!}{(u_T - (v_{T+1} - c))!}\, \pi(\boldsymbol{\theta},\, \boldsymbol{a}_c \,|\, \boldsymbol{r}_{0:T}). \tag{3.4.20}$$

It is difficult to evaluate (3.4.20) without a considerable computation cost. Therefore, in order to make progress, we consider calculations in the special case of a homogeneous population. This leads to a simpler form of (3.4.20),

$$\tilde{\pi}\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{r}_{0:T+1}\big) \;=\; \pi\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{r}_{0:T}\big) \sum_{c=0}^{z_{T+1}} \sum_{\boldsymbol{a}_c} \frac{(u_T - v_{T+1})!}{(u_T - (v_{T+1} - c))!}. \tag{3.4.21}$$

Additionally, as v_{T+1} is fixed and we require the value of u_T to remain the same, we can rewrite (3.4.21) as

$$\tilde{\pi}\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{r}_{0:T+1}\big) \;=\; \pi\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{r}_{0:T}\big) \sum_{c=0}^{z_{T+1}} \frac{(u_T - v_{T+1})!}{(u_T - (v_{T+1} - c))!} \times N_c, \tag{3.4.22}$$

where N_c is the number of sets, $\boldsymbol{a}_c$, that differ from $\tilde{\boldsymbol{i}}_{\tau:T}$ in c indexes. To calculate N_c we return to the example we have considered throughout this adjustment process.

Example Part IV: the possible, original, particles

Recall that we have a population of individuals, $\{A, B, C, D, E, F, G, H, I, J\}$, with $\mathcal{R}_T = \{J\}$ and $\mathcal{R}_{T+1} = \{J, E, F, G\}$ such that $\mathcal{V}_{T+1} = \{E, F, G\}$ are the new observations. Suppose that we have adjusted sample:

$$\tilde{\boldsymbol{i}}^{I,(j)}_{\tau:T} = \Big\{\{v, w, x, y, z\} : \tilde{\mathcal{I}}^{(j)}_T = \{E, F, G, H, I\}\Big\}.$$

This implies that $\tilde{\mathcal{S}}^{(j)}_T = \{A, B, C, D\}$. We are interested in how many versions of $\boldsymbol{i}^I_{\tau:T}$ exist, that could have been adjusted to $\tilde{\boldsymbol{i}}^{I,(j)}_{\tau:T}$?

We broke-down the summation in (3.4.21) into those particles which incorrectly guessed the status of c individuals. Suppose that we consider $c = 2$ and assume that we correctly guessed E as being infectious at time T. Additionally suppose that individuals $\{A, B\}$ are those that were switched, then there are the following possibilities:

- $\boldsymbol{i}^{I,(j)}_{\tau:t} = \Big\{\{v, w, x, y, z\} : \mathcal{I}^{(j)}_T = \{E, A, B, H, I\}\Big\}$
- $\boldsymbol{i}^{I,(j)}_{\tau:t} = \Big\{\{v, w, x, y, z\} : \mathcal{I}^{(j)}_T = \{E, B, A, H, I\}\Big\}$.

However, we could have replaced $\{A, B\}$ for any of those in $\tilde{\mathcal{S}}^{(j)}_T$. For example, as $\{C, D\} \in \tilde{\mathcal{S}}^{(j)}_T$ we could also have

- $\boldsymbol{i}^{I,(j)}_{\tau:t} = \Big\{\{v, w, x, y, z\} : \mathcal{I}^{(j)}_T = \{E, C, D, H, I\}\Big\}$
- $\boldsymbol{i}^{I,(j)}_{\tau:t} = \Big\{\{v, w, x, y, z\} : \mathcal{I}^{(j)}_T = \{E, D, C, H, I\}\Big\}$.

Overall there are ${}^{m^{S,(j)}_T}P_c = \frac{m^{S,(j)}_T!}{(m^{S,(j)}_T - c)!}$ permutations of individuals who could have been inferred to be in state I and then adjusted to be in state S.

Additionally we could have any combination of the $(v_{T+1} - c)$ individuals from $\mathcal{V}_{T+1}$ correctly guessed. For example we could also have

- $\boldsymbol{i}^{I,(j)}_{\tau:t} = \Big\{\{v, w, x, y, z\} : \mathcal{I}^{(j)}_T = \{A, F, B, H, I\}\Big\}$.

Therefore, there will be $\binom{v_{T+1}}{c}$ possibilities for which infectious individuals we incorrectly guessed to be susceptible.

Using this example as motivation, we can deduce a general form of N_c from (3.4.22).

For $c = 0, \dots, z_{T+1}$,

$$N_c = \binom{v_{T+1}}{c} \times \frac{m_T^S!}{(m_T^S - c)!} \tag{3.4.23}$$

where m_T^S will depend on the sample. We can then substitute the value of N_c into (3.4.22), such that

$$\tilde{\pi}\big(\boldsymbol{\theta}, \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{r}_{0:T+1}\big) = \pi\big(\boldsymbol{\theta}, \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{r}_{0:T}\big) \sum_{c=0}^{z_{T+1}} \frac{(u_T - v_{T+1})!}{(u_T - (v_{T+1} - c))!} \binom{v_{T+1}}{c} \frac{m_T^S!}{(m_T^S - c)!}. \tag{3.4.24}$$

3.4.5.1 Further Simplification of the Adjusted Distribution

Although (3.4.24) shows the adjusted posterior in a simpler form, it can be further simplified by expanding the terms inside the summation. We see from the calculations so far that the adjusted distribution takes the form

$$\tilde{\pi}\big(\boldsymbol{\theta}, \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{r}_{0:T+1}\big) \;=\; \pi\big(\boldsymbol{\theta}, \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{r}_{0:T}\big) \underbrace{\sum_{c=0}^{z_{T+1}} \left\{ \binom{u_T - (v_{T+1} - c)}{c}^{-1} \binom{v_{T+1}}{c} \binom{m_T^S}{c} \right\}}_{W}. \tag{3.4.25}$$

The next step is to try and simplify the summation in (3.4.25). We begin by writing the summation term, W, in terms of factorials:

$$W \;=\; \sum_{c=0}^{z_{T+1}} \frac{(u_T - v_{T+1})!c!}{(u_T - v_{T+1} + c)!} \;\times\; \frac{v_{T+1}!}{c!(v_{T+1} - c)!} \;\times\; \frac{m_T^S!}{c!(m_T^S - c)!}. \tag{3.4.26}$$

Next we can expand the summation and cancel out some terms before finally rewriting the summation in a simpler form:

$$\begin{aligned} W \;&=\; \underbrace{1}_{c\,=\,0} \;+\; \underbrace{\frac{v_{T+1} m_T^S}{1!(u_T - v_{T+1} + 1)}}_{c\,=\,1} + \underbrace{\frac{v_{T+1}(v_{T+1} - 1) m_T^S (m_T^S - 1)}{2!(u_T - v_{T+1} + 1)(u_T - v_{T+1} + 2)}}_{c\,=\,2} + \cdots \\ &=\; \underbrace{1}_{c\,=\,0} \;+\; \underbrace{\frac{(-v_{T+1})(-m_T^S)}{1!(u_T - v_{T+1} + 1)}}_{c\,=\,1} + \underbrace{\frac{(-v_{T+1})(-v_{T+1} + 1)(-m_T^S)(-m_T^S + 1)}{2!(u_T - v_{T+1} + 1)(u_T - v_{T+1} + 2)}}_{c\,=\,2} + \cdots \\ &=\; \sum_{c=0}^{z_{T+1}} \frac{(-v_{T+1})_c (-m_T^S)_c}{c!(u_T - v_{T+1} + 1)_c}, \end{aligned} \tag{3.4.27}$$

where $(x)_k$ is the rising factorial, such that $(x)_k = x(x+1)\cdots(x+k-1)$ for $k \in \mathbb{N}_0$ with $(x)_0 = 1$, this is often referred to as the *Pochhammer symbol.* We can note that

W could be extended to a summation over infinity, although all terms after z_{T+1} will be zero. This can be clearly seen in (3.4.25) where $c > z_{T+1}$ would result in one of the coefficients being zero.

Although this form is more compact it is not immediately any easier to work with. First we need to define a special class of functions called the *hypergeometric functions* (Bailey (1964, Chapter 1)). In particular we shall be focusing on *Gauss' hypergeometric function* which takes the form

$$\begin{aligned} {}_2F_1(x, y, z; a) &= 1 + \frac{xy}{1!z}a + \frac{x(x+1)y(y+1)}{2!z(z+1)}a^2 + \cdots \\ &= \sum_{n=0}^{\infty} \frac{(x)_n\,(y)_n\,a^n}{(z)_n\,n!}, \end{aligned} \tag{3.4.28}$$

where it is assumed that z is positive. This summation converges for all $|a| < 1$ and for $|a| = 1$ if $\mathbb{R}(z - x - y) > 0$. Under special conditions the hypergeometric function can take on a simpler form, the form we will be most interested in is when $a = 1$.

Theorem 2 (Gauss' Hypergeometric Theorem)**.**
Let $\Gamma(n)$ *denote the gamma function, then*

$${}_2F_1(x, y, z; 1) = \frac{\Gamma(z)\Gamma(z-x-y)}{\Gamma(z-x)\Gamma(z-y)} \qquad \textit{if} \quad \mathbb{R}(z-x-y) > 0. \tag{3.4.29}$$

Proof. See Bailey (1964, Chapter 1).

The form of (3.4.27) looks similar to the form of hypergeometric functions in (3.4.28) suggesting that we can re-write this in terms of Gauss' hypergeometric function. Suppose we consider

$$x = -v_{T+1}, \qquad y = -m_T^S, \qquad z = u_T - v_{T+1} + 1, \qquad a = 1, \tag{3.4.30}$$

then we can write

$$\begin{aligned} W &= \sum_{c=0}^{z_{T+1}} \frac{(-v_{T+1})_c(-m_T^S)_c}{c!(u_T - v_{T+1} + 1)_c} \\ &= {}_2F_1\big(-v_{T+1},\ -m_T^S,\ u_T - v_{T+1} + 1;\ 1\big). \end{aligned} \tag{3.4.31}$$

Substituting (3.4.31) into (3.4.25) we find that the adjusted posterior distribution can be written as

$$\tilde{\pi}\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T}\,|\,\boldsymbol{r}_{0:T+1}\big) \;=\; \pi\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T}\,|\,\boldsymbol{r}_{0:T}\big) \times {}_2F_1\big(-v_{T+1},\; -m^S_T,\; u_T - v_{T+1} + 1;\; 1\big). \tag{3.4.32}$$

So far we have only applied a definition, we still wish to further simplify this form to hopefully better understand the distribution from which the adjusted particles comes from. The next step therefore is to check we satisfy the conditions required for Gauss' Hypergeometric Theorem:

$$\begin{aligned}\mathbb{R}(z - x - y) \;&=\; u_T - v_{T+1} + 1 \;+\; v_{T+1} \;+\; m^S_T \\ &=\; u_T + 1 \;+\; m^S_T \;>\; 0.\end{aligned} \tag{3.4.33}$$

Therefore we can use Theorem 2 to simplify the summation in terms of Gamma functions:

$$W \;=\; {}_2F_1\big(-v_{T+1},\; -m^S_T,\; u_T - v_{T+1} + 1;\; 1\big) \;=\; \frac{\Gamma(u_T - v_{T+1} + 1)\,\Gamma(u_T + 1 + m^S_T)}{\Gamma(u_T + 1)\,\Gamma(u_T - v_{T+1} + 1 + m^S_T)}. \tag{3.4.34}$$

We can then use the fact that for $n \in \mathbb{N}$, $\Gamma(n) = (n-1)!$ such that

$$\begin{aligned}W \;&=\; \frac{(u_T - v_{T+1})!\,(u_T + m^S_T)!}{u_T!\,(u_T - v_{T+1} + m^S_T)!} \\ &=\; \frac{(u_T - v_{T+1})!\,(u_T + m^S_T)!\,v_{T+1}!}{u_T!\,(u_T - v_{T+1} + m^S_T)!\,v_{T+1}!} \\ &=\; \binom{u_T}{v_{T+1}}^{-1}\binom{u_T + m^S_T}{v_{T+1}} \\ &=\; \binom{u_T}{v_{T+1}}^{-1}\binom{N_{pop} - m^R_T}{v_{T+1}} \\ &=\; \binom{u_T}{v_{T+1}}^{-1} f\big(N_{pop},\; m^R_T,\; v_{T+1}\big)\end{aligned} \tag{3.4.35}$$

where f is an arbitrary function, not dependent on the particle we are considering.

Altogether we can simplify the adjusted distribution as

$$\tilde{\pi}\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T+1}\,|\,\boldsymbol{r}_{0:T}\big) \;=\; \pi\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T}\,|\,\boldsymbol{r}_{0:T}\big) \times \binom{u_T}{v_{T+1}}^{-1} f\big(N_{pop},\, m_T^R,\, v_{T+1}\big). \tag{3.4.36}$$

Therefore we can relate the original and the adjusted posterior distributions as

$$\pi\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T}\,|\,\boldsymbol{r}_{0:T}\big) \;\propto\; \tilde{\pi}\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T}\,|\,\boldsymbol{r}_{0:T+1}\big)\binom{u_T}{v_{T+1}}. \tag{3.4.37}$$

This is a direct result of N_{pop}, m_T^R and v_{T+1} being fixed for all particles and therefore we are only concerned with the factor of the weight relating to u_T. Consequently this means that if we have a homogeneous population then the posterior distribution, (3.4.5), breaks down as

$$\begin{aligned}\pi\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T+1}\,|\,\boldsymbol{r}_{0:T+1}\big) \;\propto\; & \tilde{\pi}\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T}\,|\,\boldsymbol{r}_{0:T+1}\big)\\ & \times\pi\big(\boldsymbol{r}_{T+1}\,|\,\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T},\, \boldsymbol{r}_{0:T}\big)\binom{u_T}{v_{T+1}}\\ & \times\pi\big(\boldsymbol{i}_{T+1}\,|\,\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T},\, \boldsymbol{r}_{0:T}\big).\end{aligned} \tag{3.4.38}$$

In conclusion we have found that for a homogeneous population the adjustment process can be fully accounted for by reweighting of the particles. In general we will not be working with a homogeneous population and thus this assumption will not hold. However, we expect that the weight calculated will form a reasonable approximation to the true weight.

3.4.6 Particle Weight and Resampling

Returning to our algorithm we now have n particles which have been adjusted to be consistent with the new data. Therefore the next step is to calculate their weight. These weights will then be used to resample the particles.

Considering (3.4.38) the unnormalised weight of particle j is now

$$\tilde{w}_{T+1}^{(j)} \;\approx\; \underbrace{\pi\left(\boldsymbol{r}_{T+1}\,|\,\boldsymbol{\theta}^{(j)},\, \tilde{\boldsymbol{i}}_{\tau:T}^{(j)},\, \boldsymbol{r}_{0:T}\right)}_{w_{T+1}^{(j)}} \times \binom{{u_T}^{(j)}}{v_{T+1}}, \tag{3.4.39}$$

where $w_{T+1}^{(j)}$ is the same as the weight previously defined. Therefore, as previously, for particle j, $w_{T+1}^{(j)}$ is the probability of the removals we have witnessed occurring, whilst the other occult individuals remain infectious:

$$\begin{aligned} w_{T+1}^{(j)} = \prod_{\ell \in \mathcal{V}_{T+1}} & P\big(H = (T+1) - \tilde{i}_T^{\ell,(j)} \mid H > T - \tilde{i}_T^{\ell,(j)}\big) \\ & \times \prod_{\ell \in \tilde{\mathcal{I}}_T^{(j)} \setminus \mathcal{V}_{T+1}} P\big(H > (T+1) - \tilde{i}_T^{\ell,(j)} \mid H > T - \tilde{i}_T^{\ell,(j)}\big) \end{aligned} \tag{3.4.40}$$

where $H \sim g_H$ and $\tilde{\mathcal{I}}_T^{(j)}$ and $\tilde{i}_T^{\ell,(j)}$ depend on the particle, j.

Once the weight, (3.4.39), has been calculated for each particle we can perform a simple random sample to select n particles (see, Section 1.6.2.3). Once completed we will have n particles, each with weight $1/n$. We remove the ~ notation henceforth.

3.4.7 Particle Augmentation

The next step in the algorithm is to augment the resampled particles with the new information sampled at time $T+1$, this relates to (3.4.5c) and also the last term in (3.4.38). Considering the breakdown of the posterior distribution, this new information will be sampled from

$$\pi(\boldsymbol{i}_{T+1} \,|\, \boldsymbol{\theta},\, \boldsymbol{i}_{\tau:T},\, \boldsymbol{r}_{0:T}). \tag{3.4.41}$$

We are therefore updating the infection times to include individuals newly infected at time $T+1$. We generate the new infections in the following way:

1. For each particle, $j = 1, \dots, n$

   (a) For each susceptible at time T, $\ell \in \mathcal{S}_T^{(j)}$

      (i) Generate X_ℓ, a Bernoulli random variable such that

      $$X_\ell \sim \text{Bernoulli}\left(1 - P_T(\ell\,;\, \boldsymbol{\theta}^{(j)})\right),$$

      where $P_T(\ell\,;\, \boldsymbol{\theta}^{(j)})$ is the probability individual ℓ avoids becoming infectious at time $T+1$.

      - If $X_\ell = 1$ then ℓ is newly infected at time $T+1$.
      - If $X_\ell = 0$ then ℓ remains susceptible at time $T+1$.

Those individuals that are newly infected become part of the set of occult individuals within particle j, with infection time $T+1$. We therefore have updated the unobserved data such that $\boldsymbol{y}_{\tau:T+1} = \boldsymbol{i}_{\tau:T+1} = \left\{\boldsymbol{i}^R_{\tau:T+1},\, \boldsymbol{i}^I_{\tau:T+1}\right\}$, where the infection times of those newly removed will now be contained in $\boldsymbol{i}^R_{\tau:T+1}$. Once this step has been completed we will have n samples from approximately the posterior distribution at time $T+1$, $\pi(\boldsymbol{\theta},\, \boldsymbol{i}_{\tau:T+1} \,|\, \boldsymbol{r}_{0:T+1})$.

3.4.8 Moving the Particles

We could choose to finish this iteration of the SMC algorithm here, as we will already have an (approximate) sample from the posterior distribution at time $T+1$. However, as mentioned previously, there will still be particle degeneracy in this algorithm. Additionally, the readjustment will have introduced errors which will be compounded as we apply the algorithm over multiple times steps. This is a result of the structure of the SMC algorithm and if we intended to run it over a shorter period of time this would not be of major concern. However, often our aim will be to sequentially update the samples over the course of many days, as the outbreak is developing. Therefore, these issues may significantly affect the accuracy of the estimates at later time steps.

We aim to counteract these problems by running a short MCMC algorithm on each particle. This has two advantages: firstly, it will allow the particles to move and thus increase their diversity. Secondly, when resampling the particles we used an approximation of the true particle weight. As a result we expect the particles to be close to, but not precisely from, the true posterior distribution. Therefore a short MCMC will allow any movement away from the truth, as a result of the adjustment step, to be counteracted. This shares many similarities to the movement step described in Section 1.6.3, as proposed by Berzuini et al. (1997).

We will use the same MCMC algorithm defined previously in Section 3.3, which was also used to initialise the SMC (Section 3.4.1), and we refer the reader to these sections for further detail. We perform n_p iterations of the MCMC algorithm on each particle, independently. After running n_p iterations we take the final samples in each of the n chains to be the new set of particles. Although we are including an MCMC step the advantage over the full MCMC algorithm is that for each particle the MCMC is applied independently. Consequently this step can be trivially parallelized to decrease the computational intensity of the algorithm.

3.4.8.1 The Removal Times

Recall that previously, to run the MCMC, we required inference of the removal times of the occult individuals, represented by $\boldsymbol{r}^{I}_{0:t}$. At the start of applying the SMC we discarded these values for each particle. This is as ensuring the newly observed removals were consistent with the inferred removal times would have introduced additional complexity. As a result, prior to applying the MCMC, we need to generate these times.

Sampling the removal times is straightforward as they only depend on the infection times for each occult individual, which are contained within each particle. We therefore generate the new removal times (and consequently the infectious periods) in the following way:

1. For each particle, $j = 1, \ldots, n$
   - (a) For each current infective, $\ell \in \mathcal{I}^{(j)}_{T+1}$
     - (i) Generate $h^{\ell,(j)}_{T+1}$ using the conditional probability

       $$f^{\ell,(j)}_{H}(x\,;T+1) = P\left(H = x \,|\, H > T+1-i^{\ell,(j)}_{T+1}\right), \tag{3.4.42}$$

       where H is a random variable with probability mass function g_H.
     - (ii) Set $r^{\ell,(j)}_{T+1} = i^{\ell,(j)}_{T+1} + h^{\ell,(j)}_{T+1}$.
   - (b) Set $\boldsymbol{r}^{I,(j)}_{0:T+1} = \left\{r^{k,(j)}_{T+1} : k \in \mathcal{I}^{(j)}_{T+1}\right\}$ and $\boldsymbol{y}^{(j)}_{\tau:T+1} = \left\{\boldsymbol{i}^{R,(j)}_{\tau:T+1},\, \boldsymbol{i}^{I,(j)}_{\tau:T+1},\, \boldsymbol{r}^{I,(j)}_{0:T+1}\right\}$.

We have used a conditional distribution as we know these individuals are not removed at time $T+1$. This ensures that the occult individuals have removal times that are consistent with the new data. Once this step has been performed we can run the MCMC algorithm on each particle.

3.4.8.2 Tuning the MCMC

We will not adaptively tune the MCMC within the movement step, however, we will adapt the tuning parameters once per SMC iteration, prior to applying the MCMC, to ensure that we maintain a reasonable acceptance rate.

We tune using similar ideas to those discussed in Section 1.4.4.4 and those we described when constructing the MCMC algorithm in Section 3.3. To update the parameters we will often use a RWM proposal step, utilising an estimation of the covariance matrix of the parameters, as discussed in Section 1.4.4.4. When running the full MCMC algorithm we obtain this matrix by repeatedly calculating the covariance matrix of the samples for a pre-defined number of iterations, at the start of the chain. For the SMC method we already expect the samples after the augmentation step to be close to a true representation from the posterior distribution, therefore we choose to instead calculate the covariance matrix from the current particles. This will ensure that as our distribution changes the proposal changes accordingly. We then apply the full MCMC movement step, on each particle, using this covariance matrix.

We next consider the tuning parameters used within the updating steps relating to the augmented data, as described in Section 3.3.2. These will also need to change as the posterior distribution we are interested in evolves. To choose the tuning values we use the fact that the posterior distribution between two times steps will not change dramatically. Therefore we monitor the acceptance rates of the MCMC applied within the SMC, if this drops too low or high then we can change the tuning parameter accordingly, as discussed in Section 3.3.6. This adaptation is performed in an identical manner to the adaptive method used within the full MCMC algorithm previously defined and ensures that we maintain the desired acceptance rate. Again, this adaptation is performed before we apply the movement step.

3.4.9 Summary of the SMC Steps

Once all of the steps described have been completed we will have produced samples from (approximately) the posterior distribution at time $T+1$. This algorithm can then be repeated for as many time steps forward as we desire. In summary the steps of the SMC algorithm are:

1. Generate n particles (see, Section 3.4.1).
    - Using an MCMC algorithm, generate particles from the posterior distribution.
2. Acquire the new data (see, Section 3.4.2).
    - At the next time step obtain the new data, which we aim to incorporate.

3. Adjust each particle (additional step, see Section 3.4.4).
    - Adjust each of the particles sampled, so that they are consistent with the newly observed data.
4. Calculate the weight of each particle (see, Sections 3.4.5 and 3.4.6).
    - For each particle calculate its weight, such that the particles form a properly weighted sample.
5. Resample the particles (see, Section 3.4.6).
    - Using the previously calculated weights resample the particles, once completed each sample will have weight $1/n$.
6. Augment the particles (see, Section 3.4.7).
    - For each particle generate the individuals infected at this time step and add this new information to the particle.
7. Move the particles (see, Section 3.4.8).
    - For each particle run a short MCMC of length n_p, this will add diversity to the samples and ensure the particles are from the target posterior distribution.
8. Return to Step 2.
    - Repeat this process when new data arrives at the next time step.

We provide an illustration of the SMC algorithm developed in Figure 3.2. Additionally, we provide a formal description in Algorithm 11, where the notation follows that discussed in Section 3.2.3.

Algorithm 11: Sequential Monte Carlo Algorithm for an SIR Epidemic

```
1. At time $t = T$ generate $n$ particles from $\pi\left(\boldsymbol{\theta},\, \boldsymbol{i}_{\tau:t} \,|\, \boldsymbol{r}_{0:t}\right)$.
2. for $t = T,\, T+1,\, T+2,\, \ldots$
    (i) Gather the new data at time $t+1$, $\boldsymbol{r}_{t+1}$, and set $\mathcal{V}_{t+1} = \mathcal{R}_{t+1} \backslash \mathcal{R}_t$.
    (ii) for $j = 1, \ldots, n$,   if  $u_t^{(j)} \geq v_{t+1} = |\mathcal{V}_{T+1}|$
        (a) Set $D^{(j)} = \mathcal{V}_{t+1} \backslash \mathcal{I}_t^{(j)}$ and $E^{(j)} = \mathcal{I}_t^{(j)} \backslash \mathcal{V}_{t+1}$.
        (b) for $\ell \in D^{(j)}$
            • Select at random, $a \in E^{(j)}$, set $i_t^{\ell,(j)} = i_t^{a,(j)}$ and then $i_t^{a,(j)} = \infty$.
            • Let $E^{(j)} \longrightarrow E^{(j)} \backslash \{a\}$.
        (c) For each $z = \tau, \ldots, t$ calculate $\mathcal{I}_z^{(j)}$ and $\mathcal{S}_z^{(j)}$.
        (d) Calculate the approximate weight of particle $j$ as
            $w_{t+1}^{(j)} = \binom{u_t^{(j)}}{v_{t+1}} \prod_{\ell \in \mathcal{V}_{t+1}} \frac{P\left(H = t+1 - i_t^{\ell,(j)}\right)}{P\left(H > t - i_t^{\ell,(j)}\right)} \times \prod_{\ell \in \mathcal{I}_t^{(j)} \backslash \mathcal{V}_{t+1}} \frac{P\left(H > t+1 - i_t^{\ell,(j)}\right)}{P\left(H > t - i_t^{\ell,(j)}\right)}.$
    (iii) Resample $n$ particles with probability proportional to their weight, $w_{t+1}^{(j)}$.
    (iv) for $j = 1, \ldots, n$
        (a) for $\ell \in \mathcal{S}_t^{(j)}$
            • Generate $X_\ell^{(j)} \sim \text{Bernoulli}\big(1 - P_t\big(\ell\,;\, \boldsymbol{\theta}^{(j)}\big)\big)$.
            • if $X_\ell^{(j)} = 1$  then $\ell$ is newly infected at time $t+1$.
        (b) Update $\boldsymbol{y}_{\tau:t+1}^{(j)} = \boldsymbol{i}_{\tau:t+1}^{(j)} = \big\{\boldsymbol{i}_{\tau:t+1}^{I,(j)},\, \boldsymbol{i}_{\tau:t+1}^{R,(j)}\big\}$ to incorporate information
            on the new infections and removals.
        (c) For each $\ell \in \mathcal{I}_{t+1}^{(j)}$ generate the infectious period $h_{t+1}^{\ell,(j)} \sim f_H^{\ell,(j)}(\cdot)$
            where
                $f_H^{\ell,(j)}(x\,;\, t+1) \;=\; P\left(H \;=\; x \,|\, H > t+1 - i_{t+1}^{\ell,(j)}\right).$
        (d) Compute the parameter covariance matrix and adapt the tuning
            parameters if the previous movement step had a poor acceptance rate.
        (e) Run an MCMC of length $n_p$ on particle $j$ then discard $h_{t+1}^{\ell,(j)}$ for
            $\ell \in \mathcal{I}_{t+1}^{(j)}$.
```

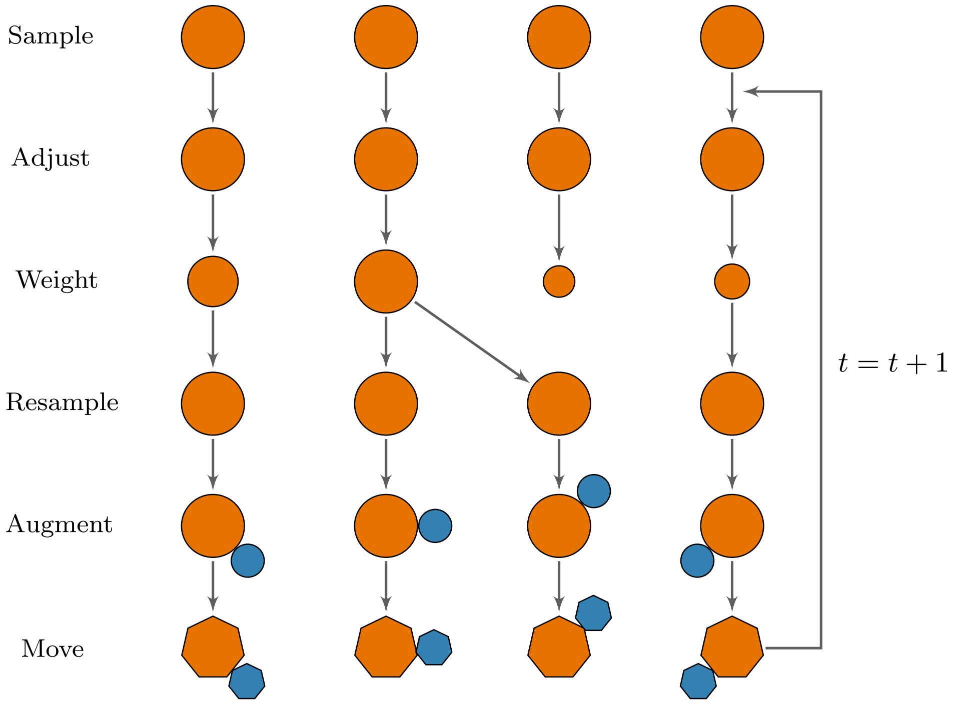


Figure 3.2: Illustration of the SMC algorithm for application to epidemic data.

The sequential Monte Carlo algorithm we have discussed can be applied at each successive time step, as new information is obtained. The particles will continue to update in each iteration to incorporate any newly obtained information and represent the up-to-date posterior distribution.

Throughout we have treated the MCMC and SMC algorithms as alternative methods to the same problem of generating samples from evolving target distributions. However, the SMC algorithm we have constructed can also been seen as a hybrid of the sequential importance resampling and the MCMC methods we discussed in Chapter 1. The strength of the SMC algorithm we have constructed is that it combines elements of both methods to produce a computationally efficient and accurate algorithm. Indeed, if we recall n_p, the length of the MCMC applied to each particle at the end of the SMC algorithm, then this can be see as a parameter controlling how similar to each algorithm our method is. Small n_p produces a method similar to the sequential importance resampling algorithm, whereas larger n_p constructs an algorithm closer to the standard MCMC. This hybrid nature ensures that we have a flexible algorithm that can be tailored to the research question we are interested in.

3.5 Extension: A Non-Uniform Adjustment

Recall that, to generate samples from the target posterior distribution at time T, we require samples for the unobserved process, $\boldsymbol{y}_{\tau:T}$, where for the SMC this contains the infection times of the observed and the occult individuals, such that $\boldsymbol{y}_{\tau:T} = \boldsymbol{i}_{\tau:T}$. Consequently, we also need to infer who is infectious at time T. This can become problematic at the next time step, as we may observe individuals being removed on day $T+1$ who were not inferred to be infectious on day T.

The proposed solution to this problem is to adjust the samples generated so that they are compatible with the new data. We achieve this by randomly allocating the newly removed individuals, who we had not inferred to be infectious at time T, infection times from individuals who we inferred to be infectious at time T but who were not removed at time $T+1$. This method ensures the particles generated at time T are consistent with the new data obtained at time $T+1$. Additionally, this adjustment aims to change each sample as little as possible, preserving the infection times and the number of them—only changing who is infectious at time T.

One potential problem with this adjustment is that it does not consider any properties of the individuals when choosing how to adjust the particles. As such we may, by chance, produce an adjusted particle containing individuals with highly unlikely infectious periods. We illustrate this using the example we have considered previously in Section 3.4.

Example Extension: the original adjustment

Recall that we have a population, $\{A, B, C, D, E, F, G, H, I, J\}$, with observed data up until time T, $\mathcal{R}_T = \{J\}$. At time T, one possible sample is

$$\boldsymbol{i}_{\tau:T}^{I,(j)} = \left\{\{v, w, x, y, z\} : \mathcal{I}_T^{(j)} = \{E, A, B, H, I\}\right\}.$$

However, at time $T+1$ we witness $\mathcal{R}_{T+1} = \{J, E, F, G\}$. Therefore, as individuals F and G were not inferred to be infectious at time T, this particle is not consistent. Using the current adjustment scheme we can fix this particle in the following way:

- With probability $\frac{1}{2}$ select F, then swap them with A with probability $\frac{1}{4}$.

- With probability 1 select G, then swap them with B with probability $\frac{1}{3}$.

Therefore we have the adjusted particle

$$\tilde{\boldsymbol{i}}^{I,(j)}_{\tau:T} = \Big\{ \{v, w, x, y, z\} : \tilde{\mathcal{I}}^{(j)}_T = \{E, F, G, H, I\} \Big\}.$$

Suppose that $(w,\, x,\, y,\, z) = (T,\, T-1,\, T-4,\, T-6)$ and thus we have allocated the newly removed individuals very short infectious periods. If individuals are assumed to have, on average, a long infectious period then it perhaps would have been more appropriate to swap them with individuals H and I, whose infection times are much earlier.

3.5.1 An Alternative Weighting

One possible extension to this idea is to instead select who we switch according to some criteria. The criteria we shall focus on is based on the (discrete) hazard function,

$$h(t) \;=\; P(H \;=\; t \,|\, H \;\geq\; t) \;=\; \frac{P(H \;=\; t)}{P(H \;\geq\; t)}, \tag{3.5.1}$$

where H is some discrete random variable. This function represents the probability of dying at time t, given the individual has survived up to this time. We can recast this by interpreting an individual's infectious period as 'dying' and thus they are no longer infectious. Therefore, continuing with our definition of the infectious period distribution the probability mass function of H is g_H.

Considering the adjustment step, we are interested in deciding which individuals to swap with the newly removed individuals to make the particle consistent. In the previous section we described choosing which individuals to switch at random. However, now we instead propose an alternative method which utilises each individuals hazard. Practically, this now means that for the newly removed individual at time $T+1$, who we did not correctly infer to be infectious at time T, we select their infection time to be that of individual $k \in \mathcal{I}_T \backslash \mathcal{V}_{T+1}$ with probability proportional to

$$h\big(T+1-i^k_T\big) \;=\; P\big(H \;=\; T+1-i^k_T \mid H \geq T+1-i^k_T\big). \tag{3.5.2}$$

The justification behind this is that we are allocating those newly removed infection times which means they had a high chance of being removed. We illustrate this new adjustment scheme by continuing the previous example.

Example Extension: an alternative adjustment

We again suppose that at time T we have the sample,

$$\boldsymbol{i}^{I,(j)}_{\tau:T} = \left\{ \{v, w, x, y, z\} : \mathcal{I}^{(j)}_{T} = \{E, A, B, H, I\} \right\},$$

with $\mathcal{V}_{T+1} = \{E, F, G\}$. Using the new adjustment scheme we can fix this particle in the following way:

- With probability $\frac{1}{2}$ select F and then swap them with A with probability

$$\frac{h(T+1-w)}{h(T+1-w) + h(T+1-x) + h(T+1-y) + h(T+1-z)}.$$

- With probability 1 select G and then swap them with B with probability

$$\frac{h(T+1-x)}{h(T+1-x) + h(T+1-y) + h(T+1-z)}.$$

Therefore we have adjusted particle

$$\tilde{\boldsymbol{i}}^{I,(j)}_{\tau:T} = \left\{ \{v, w, x, y, z\} : \tilde{\mathcal{I}}^{(j)}_{T} = \{E, F, G, H, I\} \right\}.$$

The probability of the adjustment we performed is

$$\left(\frac{1}{2} \times \frac{h(T+1-w)}{h(T+1-w) + h(T+1-x) + h(T+1-y) + h(T+1-z)} \right)$$
$$\times \left(1 \times \frac{h(T+1-x)}{h(T+1-x) + h(T+1-y) + h(T+1-z)} \right).$$

This adjustment scheme works in a similar way to the original, only now we take into account the probability of witnessing certain infectious periods. As a result, the adjusted distribution cannot be related to the true distribution in the same manner as previously discussed (see, Section 3.4.5). We consider in the next section how this will affect the SMC algorithm. For clarity we refer to the previous adjustment method as the 'uniform' adjustment and the new weighting as the 'non-uniform' adjustment.

3.5.2 The New Adjusted Distribution

We begin by considering the relationship between the adjusted distribution and the posterior distribution. This relationship can be initially described in the same way as for the uniform adjustment:

$$\tilde{\pi}\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{r}_{0:T+1}\big) = \sum_{c=0}^{z_{T+1}} \sum_{\boldsymbol{a}_c} P\big(\boldsymbol{a}_c \longrightarrow \tilde{\boldsymbol{i}}_{\tau:T}\big)\, \pi\big(\boldsymbol{\theta},\, \boldsymbol{a}_c \,|\, \boldsymbol{r}_{0:T}\big). \tag{3.5.3}$$

As mentioned previously there are $N_c = {}^{m_T^S}P_c \times \binom{v_{T+1}}{c}$ possibilities $\boldsymbol{a}_c$ could take, such that $P(\boldsymbol{a}_c \longrightarrow \tilde{\boldsymbol{i}}_{\tau:T}) \neq 0$. This comprises the number of ways we could have c incorrect individuals from those newly removed, $\binom{v_{T+1}}{c}$, and the number of different ways susceptible individuals could have been substituted in their place, ${}^{m_T^S}P_c = \frac{m_T^S!}{(m_T^S - c)!}$ (see Section 3.4.5 for further detail).

We assume that we can order the possible (pre-adjusted) samples in some way, such that for $c = 0, \ldots, z_{T+1}$ we have the set

$$\mathcal{A}_c = \left\{ \boldsymbol{a}_c^{i,j} \quad \text{for } i = 1, \ldots, \binom{v_{T+1}}{c};\ \ j = 1, \ldots, {}^{m_T^S}P_c \quad \text{s.t. } P\big(\boldsymbol{a}_c^{i,j} \to \tilde{\boldsymbol{i}}_{\tau:T}\big) \neq 0 \right\}. \tag{3.5.4}$$

Therefore we can write (3.5.3) as

$$\tilde{\pi}\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{r}_{0:T+1}\big) \;=\; \sum_{c=0}^{z_{T+1}} \sum_{i=1}^{\binom{v_{T+1}}{c}} \sum_{j=1}^{{}^{m_T^S}P_c} P\big(\boldsymbol{a}_c^{i,j} \longrightarrow \tilde{\boldsymbol{i}}_{\tau:T}\big)\, \pi\big(\boldsymbol{\theta},\, \boldsymbol{a}_c^{i,j} \,|\, \boldsymbol{r}_{0:T}\big). \tag{3.5.5}$$

To make progress we, once again, assume that the population is homogeneously mixing, therefore

$$\tilde{\pi}\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{r}_{0:T+1}\big) \;=\; \pi\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{r}_{0:T}\big) \sum_{c=0}^{z_{T+1}} \sum_{i=1}^{\binom{v_{T+1}}{c}} \sum_{j=1}^{{}^{m_T^S}P_c} P\big(\boldsymbol{a}_c^{i,j} \longrightarrow \tilde{\boldsymbol{i}}_{\tau:T}\big). \tag{3.5.6}$$

Currently this relationship is the same, regardless of which updating scheme we use. However, we can note that, as in the original adjustment scheme, we do not change the infection times. As such, each set of possible susceptibles who could have been sampled as occults will have the same probability of adjustment (see, Section 3.4.5). We can see this by briefly considering our example: the initial samples,

$$\boldsymbol{i}^{I,(j)}_{\tau:T} = \Big\{\{v,w,x,y,z\} : \mathcal{I}^{(j)}_T = \{E,A,B,H,I\}\Big\}$$

$$\boldsymbol{i}^{I,(j)}_{\tau:T} = \Big\{\{v,w,x,y,z\} : \mathcal{I}^{(j)}_T = \{E,C,D,H,I\}\Big\},$$

both have the same probability of being adjusted to

$$\tilde{\boldsymbol{i}}^{I,(j)}_{\tau:T} = \Big\{\{v,w,x,y,z\} : \tilde{\mathcal{I}}^{(j)}_T = \{E,F,G,H,I\}\Big\}.$$

Therefore, we can rewrite (3.5.6) as

$$\tilde{\pi}\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T}\,|\,\boldsymbol{r}_{0:T+1}\big) \;=\; \pi\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T}\,|\,\boldsymbol{r}_{0:T}\big)\sum_{c=0}^{z_{T+1}} \frac{m^S_T!}{(m^S_T - c)!} \sum_{i=1}^{\binom{v_{T+1}}{c}} P\big(\boldsymbol{a}^{i,\cdot}_c \longrightarrow \tilde{\boldsymbol{i}}_{\tau:T}\big). \tag{3.5.7}$$

Without selecting which individuals to switch uniformly at random, we cannot simplify this any further. However, we can estimate (3.5.7) by noting that the leading order term, when N_{pop} is large, will be when $c = z_{T+1} = v_{T+1}$. Therefore for large N_{pop} we find

$$\tilde{\pi}(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T}\,|\,\boldsymbol{r}_{0:T+1}) \approx \pi(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T}\,|\,\boldsymbol{r}_{0:T})\frac{m^S_T!}{(m^S_T - v_{T+1})!}P(\boldsymbol{a}^1_{v_{T+1}} \longrightarrow \tilde{\boldsymbol{i}}_{\tau:T}), \tag{3.5.8}$$

where we only have a single term as $\binom{v_{T+1}}{v_{T+1}} = 1$.

Overall, we have determined that for large N_{pop}, and therefore an expected larger m^S_T, we can use the following (approximate) relationship

$$\tilde{\pi}\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T}\,|\,\boldsymbol{r}_{0:T+1}\big) \;\propto\; \pi\big(\boldsymbol{\theta},\, \tilde{\boldsymbol{i}}_{\tau:T}\,|\,\boldsymbol{r}_{0:T}\big)P\big(\boldsymbol{a}^1_{v_{T+1}} \longrightarrow \tilde{\boldsymbol{i}}_{\tau:T}\big). \tag{3.5.9}$$

This relationship is pleasing in its simplicity. Although this will only be an approximate to the true weight, we expect that it will be sufficient to account for the adjustment that has occurred. Therefore we only need to compute a single probability to relate the posterior and the adjusted posterior distributions.

3.5.2.1 The New Weighting

Unfortunately, calculation of the required probability is not straightforward. We have seen previously in our example that if we have an initial particle then there are many possible ways to achieve a specific adjusted particle. As such we can write this probability as

$$P\big(\boldsymbol{a}^1_{\upsilon_{T+1}} \longrightarrow \tilde{\boldsymbol{i}}_{\tau:T}\big) \;=\; \sum_{\boldsymbol{\omega}} (\upsilon_{T+1}!)^{-1} P\big(\boldsymbol{a}^1_{\upsilon_{T+1}} \longrightarrow \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{\omega}\big) \tag{3.5.10}$$

where $\boldsymbol{\omega}$ represents the set of possible orderings in which we update the υ_{T+1} new infectives. This is such that $|\boldsymbol{\omega}| = \upsilon_{T+1}!$.

For large υ_{T+1} there will be too many configurations to calculate (3.5.10). However, if we take a random permutation, $\boldsymbol{\omega}' \in \boldsymbol{\omega}$, then

$$\mathbb{E}\big[P\big(\boldsymbol{a}^1_{\upsilon_{T+1}} \longrightarrow \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{\omega}'\big)\big] \;=\; P\big(\boldsymbol{a}^1_{\upsilon_{T+1}} \longrightarrow \tilde{\boldsymbol{i}}_{\tau:T}\big), \tag{3.5.11}$$

and therefore $P\big(\boldsymbol{a}^1_{\upsilon_{T+1}} \longrightarrow \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{\omega}'\big)$ is an unbiased estimator of $P\big(\boldsymbol{a}^1_{\upsilon_{T+1}} \longrightarrow \tilde{\boldsymbol{i}}_{\tau:T}\big)$. Therefore, rather than spending significant time computing (3.5.10), we can instead choose to estimate it by considering a random permutation.

Returning to the adjustment distribution, we find that

$$\tilde{\pi}(\boldsymbol{\theta}, \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{r}_{0:T+1}) \approx c\pi(\boldsymbol{\theta}, \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{r}_{0:T}) P(\boldsymbol{a}^1_{\upsilon_{T+1}} \longrightarrow \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{\omega}'), \tag{3.5.12}$$

for some constant, c. Therefore, we can use the weighting $P(\boldsymbol{a}^1_{\upsilon_{T+1}} \longrightarrow \tilde{\boldsymbol{i}}_{\tau:T} \,|\, \boldsymbol{\omega}')^{-1}$ for each adjusted particle, this will ensure we produce consistent particles from (approximately) the target distribution. The rest of the SMC algorithm will then follow in the same way as previously (see, Section 3.4).

3.5.2.2 Summary

We have adapted the previous SMC algorithm to now incorporate the infectious period distribution when adjusting the particles. The resulting algorithm remains the same, only now we have a slightly different weighting. As previously, we have not been able to calculate the true weight and therefore have had to approximate it. This addition may not be necessary and thus, unless stated otherwise, we will continue to use the uniform weighting throughout.

3.6 Modelling an Agricultural Epidemic

3.6.1 Motivation

The SIR model is one of the most commonly examined compartmental frameworks, however, it is not always the most suitable choice when handling real data.

One extension we shall consider in detail is the case of agricultural epidemics. We are particularly interested in the 2001 UK Foot-and-Mouth (FMD) outbreak, which is further discussed in Chapter 5. This was a severe outbreak which cost the public sector an estimated £3 billion, representing 0.8% of the annual public expenditure (UK National Audit Office (2002)). Using this outbreak as motivation we briefly describe an extension based on the SINR model discussed in Section 2.4. Although we will have in mind the FMD outbreak, we aim to construct a model which can be used more generally for epidemics where

(a) The disease is rapidly transmitted at a farm level.

(b) There is complete culling/quarantining at the farm level.

Models of this form have been discussed in other work, for example, Deardon et al. (2010) and Xiang and Neal (2014) analysed the 2001 UK FMD outbreak and Jewell et al. (2009) considered both the FMD outbreak and a simulated Avian Influenza outbreak.

These outbreaks are characterised as occurring within a population of farms where, rather than modelling each individual animal, we treat each farm as an 'individual' capable of transmitting the disease. For ease we shall still refer to a farm as an 'individual' with the ability to infect others, or be infected itself. Customarily, we declare a farm as infectious once a single animal on that farm has been infected. This is suitable for highly infectious diseases (such as Foot-and-Mouth disease) which will spread rapidly through a farm once a single animal is infected. We note that we could choose to model this outbreak at an individual animal level, we choose not to for two reasons. Firstly, the data we have access to will usually only contain information at the farm level, and secondly for diseases that spread quickly it appears reasonable to represent each farm as a single individual.

3.6.2 The SINR Model

A key feature of this form of agricultural epidemics is that, as we are dealing with farms instead of people, it will often take some length of time before the farm can be fully removed from the population. Usually in this state there are restrictions placed on the farm to try to reduce the size of the outbreak. During this period a farm is infectious, but with a reduced level of infectiousness. As a result, agricultural epidemics are often modelled using an SINR framework.

In summary, first a farm is susceptible (S) and therefore can become infected. If infection occurs then it will become infectious (I) and can pass on the disease to other farms. Once the disease is identified within a farm it becomes notified (N) and may have restrictions placed on it, before it is finally removed (R) once the disease is no longer present or the farm has been quarantined. Once removed a farm plays no further role in the spread of the outbreak. We therefore require the additional notation that there are m_t^N notified individuals at time t with now $m_t^I = u_t + m_t^N$ and $m_t^R \leq m_t^N \leq m_t^I$.

Due to both the notification and removal times being observed we now will have access to two sets of data with which to form the model, where the notification times are now a sufficient indicator of which farms are infectious. As in the SIR example underlying these two observed sets of data is an unobserved process defined by the infection times.

3.6.3 The Posterior Distribution

We construct the posterior distribution in the same way as for the SIR model, with

$$\pi(\boldsymbol{\theta},\, \boldsymbol{y}_{\tau:t}\,|\,\boldsymbol{x}_{0:t}) \,\propto\, L(\boldsymbol{\theta}\,;\, \boldsymbol{y}_{\tau:t},\, \boldsymbol{x}_{0:t})\,\pi(\boldsymbol{\theta}), \tag{3.6.1}$$

where $\boldsymbol{x}_{0:t}$ contains the observed data and $\boldsymbol{y}_{\tau:t}$ the unobserved data. We now need to add to our previous definitions, to include times relating to the notifications. For the SINR model $\boldsymbol{n}$ will represent the notification times and '0' is now the time of the first observed notification. If i^k, n^k and r^k represent the infection, notification and removal time of individual k respectively then

$$i_t^k = \begin{cases} i^k & \text{if } \; i^k \leq t \\ \\ \infty & \text{if } \;\; i^k > t \end{cases}, \quad n_t^k = \begin{cases} n^k & \text{if } \; i^k \leq t \\ \\ \infty & \text{if } \;\; i^k > t \end{cases}, \quad r_t^k = \begin{cases} r^k & \text{if } \; r^k \leq t \\ \\ \infty & \text{if } \;\; r^k > t \end{cases}, \tag{3.6.2}$$

and we define

$$\begin{aligned} &\boldsymbol{i}_{\tau:t}^{I} = \{i_t^k : k \in \mathcal{I}_t\}, && \boldsymbol{i}_{\tau:t}^{N} = \{i_t^k : k \in \mathcal{N}_t\}, && \boldsymbol{i}_{\tau:t}^{R} = \{i_t^k : k \in \mathcal{R}_t\}, \\ &\boldsymbol{n}_{0:t}^{I} = \{n_t^k : k \in \mathcal{I}_t\}, && \boldsymbol{n}_{0:t}^{N} = \{n_t^k : k \in \mathcal{N}_t\}, && \boldsymbol{n}_{0:t}^{R} = \{n_t^k : k \in \mathcal{R}_t\}, \\ & && && \boldsymbol{r}_{0:t}^{R} = \{r_t^k : k \in \mathcal{R}_t\}, \end{aligned} \tag{3.6.3}$$

where $\mathcal{N}_t$ represents those individuals in state N at time t. We note that now the removal times take a slightly different form. This is because we do not model the notification period (time between notification and removal). We now define the observed and unobserved data to be

$$\boldsymbol{x}_{0:t} = \left\{\boldsymbol{n}_{0:t}^{N},\, \boldsymbol{n}_{0:t}^{R},\, \boldsymbol{r}_{0:t}^{R}\right\}, \qquad \boldsymbol{y}_{\tau:t} = \left\{\boldsymbol{i}_{\tau:t},\, \boldsymbol{n}_{0:t}^{I}\right\} = \left\{\boldsymbol{i}_{\tau:t}^{N},\, \boldsymbol{i}_{\tau:t}^{R},\, \boldsymbol{i}_{\tau:t}^{I},\, \boldsymbol{n}_{0:t}^{I}\right\}. \tag{3.6.4}$$

In practice we will treat $\boldsymbol{i}_{\tau:t}^{N}$ and $\boldsymbol{i}_{\tau:t}^{R}$ together, as the infection times of those known to be infected, and $\boldsymbol{i}_{\tau:t}^{I}$ as the infection times of the occult individuals, as before. Additionally, the key observed data is now $\boldsymbol{n}_{0:t}^{N}$ and $\boldsymbol{n}_{0:t}^{R}$, as the notification times represent when individuals are first known to be infected, taking the role of the removal times in the SIR outbreak. As such, although we observe the removal times they do not hold the same weight of information as in the SIR example.

The methods of using data augmentation to construct the likelihood are the same as those discussed previously, therefore the likelihood takes the same form as in equation (3.2.11),

$$L(\boldsymbol{\theta};\, \boldsymbol{y}_{\tau:t},\, \boldsymbol{x}_{0:t}) \;=\; \prod_{s=\tau}^{t-1} \left\{ \prod_{\ell \in \mathcal{S}_{s+1}} P_s(\ell\,;\boldsymbol{\theta}) \prod_{\ell \in \mathcal{S}_s \setminus \mathcal{S}_{s+1}} \big(1 - P_s(\ell\,;\boldsymbol{\theta})\big) \right\} \prod_{j \notin \mathcal{S}_t} g_H\big(h_t^j\,;\boldsymbol{\theta}\big) \tag{3.6.5}$$

where $P_t(\ell\,;\boldsymbol{\theta})$ is the probability of individual ℓ avoids transmission at time t. One key difference is that now the infectious period refers to the time between notification and infection:

$$h_t^j = \begin{cases} n_t^k - i_t^k & \text{if} \quad i^k \leq t \\ 0 & \text{if} \quad i^k > t \end{cases} . \tag{3.6.6}$$

Therefore, the infectious period is the time spent in state I and the notification period is the time spend in state N. Thus, to avoid infection an individual must avoid transmission from each infectious and each notified individual, therefore

$$P_t(\ell\,; \boldsymbol{\theta}) = \prod_{j \in \mathcal{I}_t} (1 - q_t(\ell, j)) \prod_{j \in \mathcal{N}_t} (1 - \kappa q_t(\ell, j)) \tag{3.6.7}$$

where κ represents the reduction in infectiousness when an individual becomes notified. We now therefore can interpret $q_t(\ell, k)$ as the probability individual k infects individual ℓ with no restrictions.

3.6.4 Extending the SMC Algorithm

The focus of the SMC algorithm defined has been on the application to data which follows a SIR compartmental framework. This was in the interest of clarity and the majority of the theory presented transfers over, the main difference is that the notification times take the place of the removal times. The notification times then inform us as to who is newly infectious and the removal times then act as an supplementary piece of information, which we can easily incorporate.

In general as we observe both the notification and removal times we will not be interested in modelling the length of this period, only how effective the notification stage is in reducing the infectiousness of the farm before full removal can occur.

This will be further discussed in Chapters 4 and 5, where we apply the SMC algorithm to simulated and real data from SINR-type outbreaks.

3.7 Discussion

We began this chapter with a discussion of the stochastic, discrete-time epidemic model which shall be the focus of our analysis. Although we required some assumptions, the final model is relatively general. As such the ideas presented in this chapter can be easily applied to a variety of outbreaks.

The first algorithm we considered was an adaptive MCMC, which uses data augmentation to sample from the posterior distribution of interest. This MCMC has been constructed to efficiently explore the posterior distribution and adaptively tune to ensure we obtain a reasonable acceptance rate. Next we focused on the main aim of this thesis, which was to utilise the ideas underpinning sequential Monte Carlo methods and adapt them for the epidemic setting. We successfully achieved this, constructing a novel method of analysing infectious disease data. The method formed is highly general, with the ability to be applied to outbreaks with a variety of behaviour.

The key advantage of the SMC algorithm when compared to the analogous MCMC algorithm is it is, theoretically, much quicker to compute. Many of the steps within the algorithm are performed on each particle independently, for example: the adjustment step, calculation of the weight, the augmentation step and the movement of the particles. As such each of these steps can be very easily parallelized and performed for each particle simultaneously. Additionally the SMC algorithm only ever updates the particles it started with, thus avoiding restarting the algorithm from the beginning. For these reasons it is a fast alternative to the commonly used MCMC algorithms.

Throughout we have formed the methods without specifying the form of the transmission probability. This is to ensure the SMC algorithm can be used on a variety of outbreaks. Thus when applying it we are free to specify an individual-level model that incorporates specific heterogeneities of interest. Additionally we have not specified the form of the infectious period distribution, again this can be chosen dependent on the outbreak we are interested in modelling.

The SMC algorithm does require tuning and this will need to be further discussed before we can truly state its advantages over any other algorithm. For example how long does the movement step need to be, to ensure we obtain accurate estimates far into the future? Too short a chain and the samples may be far from the true distribution, but too long and the algorithm may become costly. It will be this balancing of accuracy versus efficiency that requires further discussion.

Additionally, we do not currently have an understanding of the impact of the adjustment step performed within the SMC. In the interest of tractability we had to approximate the weight for each particle. This was an appropriate solution, however, we are yet to see how this will affect our inference when we apply the SMC algorithm developed.

Overall we have produced a new method for the study of outbreak data. Although further analysis will be required before we can fully know its strengths and weaknesses, it does provide a potential alternative to MCMC methods.

Chapter 4

A Comprehensive Simulation Study

Having developed the SMC algorithm, in this chapter we demonstrate the application of it to a collection of simulated data sets. As we will have knowledge of the true underlying infection process, and the values of the parameters which drive it, this is a crucial step in validating the SMC algorithm. We will assess the performance of the SMC by comparing it to the analogous MCMC algorithm, as well as the true parameter values used within the simulations.

The SMC algorithm we have constructed is fairly general in its methodology. As a result of this generality, prior to applying it we must make some decisions about the various tuning values within the algorithm. The choices we make could have a considerable impact on the results obtained and therefore must be selected with care. With this in mind, by conducting an intensive study of simulated data sets, in this chapter we also aim to gain a greater understanding of the algorithm we have constructed and how it performs under varying conditions.

4.1 Motivating Questions

Within this chapter our aim is to answer any outstanding questions we may have about the application and performance of the SMC algorithm constructed. Specifically, in the next sections we shall focus on answering the following questions:

Do the particles represent a sample from the target distribution?

This is perhaps the most important question we need to answer. In Section 4.4 we assess the results of the SMC algorithm by comparing it to the current 'gold-standard' of MCMC methods.

Does the performance of the SMC algorithm deteriorate when run for multiple time steps?

Ideally we wish to run the SMC algorithm until the end of the outbreak we are observing. However, as epidemics often occur over long periods of time, we need the SMC algorithm to remain robust when applied over many iterations. We consider this question in Section 4.4.2.

How does the SMC algorithm perform when the new data significantly change the shape of the distribution?

Often we may have a sudden influx of data that rapidly changes the shape or the location of the posterior distribution. We investigate if the SMC remains accurate in this situation in Section 4.4.2.

Is the SMC algorithm computationally faster than the analogous MCMC?

The key benefit of the SMC algorithm is that, in theory, it allows for the fast generation of samples from the target distribution. We check that this is true in Section 4.5, where we consider the computation time of the SMC against the comparable MCMC algorithm.

How many iterations are required in the movement step within the SMC algorithm, to produce accurate results?

The introduction of a movement step ensures that the SMC repeatedly generates samples from the evolving target distribution. In Section 4.6 we investigate the required length of this step, which ensures that (at each time step) we are producing samples from sufficiently close to the truth.

Can the SMC maintain reasonable efficiency, with no further human interaction necessary?

We constructed the SMC algorithm with the intention that it could repeatedly incorporate new observations, without requiring any further tuning to efficiently produce samples from the target distribution. We check that this is indeed the case in Section 4.7, where we consider the acceptance rate of the movement step.

How severe is the particle degeneracy and does this affect the results?
One disadvantage of the SMC algorithm will be the removal of particles during the resampling step, which could impact the accuracy of the final output. We therefore investigate the particle degeneracy of the SMC algorithm in Section 4.8.

Is the adjustment step necessary?
One weakness of the methods developed is that we choose to adjust the particles to ensure that they are consistent with the new data. In Section 4.9 we illustrate why this additional step is required.

How is the accuracy of our analysis affected if we use the wrong transmission model?
When working with simulated outbreaks we are fortunate to know the true underlying transmission mechanism, this is not the case when using real data. Therefore in Section 4.10 we consider the effect of assuming an incorrect transmission model.

Can the algorithm accurately infer the infectious period parameters?
The form of infectious period distribution is often difficult to infer. We aim to test this in Section 4.11, where we estimate the value of the infectious period parameter.

4.2 Simulating Inhomogeneous Epidemic Data

We begin by briefly describing the generation of an inhomogeneous epidemic data set. We will consider a fairly simple population where the differences between individuals is only in their location, with no other heterogeneity incorporated. To simulate an outbreak we assume we have a population of size N_{pop} with, in most of the examples that we consider, individuals uniformly distributed on a (1×1) square.

The outbreak begins with a single infectious individual, chosen at random. Then at each subsequent time step we randomly choose who is newly infected, dependent on their probability of being infected at that time, $1 - P_t(\cdot)$. If infected we then generate their infectious period from the underlying infectious period distribution, with probability mass function $g_H(\cdot)$. We repeat this at each time step, until there are no more infectious or no more susceptible individuals.

We illustrate the simulation of a discrete-time SIR outbreak in Algorithm 12. A similar method can be also be used to generate an SINR epidemic.

Algorithm 12: Simulating a Discrete-Time SIR Epidemic

```
1. Generate the spatial locations of $N_{pop}$ individuals.
2. Choose an individual from the $N_{pop}$ generated to be the initially infectious
   individual. Denote this individual by $v$ and set their infection time as $i^v = 0$.
3. Generate the removal time of the initial infective as $r^v \sim g_H$.
4. Set $\mathcal{S}_0 = \{1, \dots, N_{pop}\} \backslash v$, $\mathcal{I}_0 = v$ and $\mathcal{R}_0 = \emptyset$.
5. Let $t = 0$.
6. while $|\mathcal{S}_t| > 0$ and $|\mathcal{I}_t| > 0$
    (i) for $\ell \in \mathcal{S}_t$
        (a) Generate $X_\ell \sim$ Bernoulli(1, $1 - P_t(\ell)$).
        (b) if $X_\ell = 1$ then
                $\ell$ is newly infected at time $t+1$ and we generate $r^\ell \sim (t+1) + g_H$.
            if $X_\ell = 0$ then
                $\ell$ remains susceptible at time $t+1$.
    (ii) Calculate $\mathcal{S}_{t+1}$, $\mathcal{I}_{t+1}$ and $\mathcal{R}_{t+1}$.
    (iii) Let $t = t + 1$.
7. If desired, shift the times so that $r^v = 0$.
```

4.2.1 Generating the Notification and Removal Times

One choice when simulating epidemic data is the distribution of the time between infection and removal in the SIR model (or infection and notification in the SINR model), represented in the Algorithm 12 by g_H. As we are working in discrete time, in all of the simulations we consider, we will choose infectious period distribution $H \sim \text{Poisson}(a) + 1$ where $(a + 1)$ is the mean length of the infectious period. The '+1' is necessary to ensure that we do not have individuals being removed on the same day that they were infected.

Additionally, when generating an SINR-type outbreak, we assume that the time between notification and removal is of constant length, d. This is fairly realistic as we would expect there to be a standard protocol to follow once an individual becomes notified.

Finally, once the outbreak has concluded we shift the times generated so that $t = 0$ relates to the time of the first observed notification or removal. This is to match with the methods described in the previous chapter. The notification and removal times then form the '*observed*' data.

4.2.2 The Transmission Probability

To simulate an outbreak we need to select the underlying mechanism of transmission. If we have a susceptible individual, ℓ, then we are interested in the probability ℓ avoids infection at time t, denoted by $P_t(\ell) = P_t(\ell\,;\boldsymbol{\theta})$ where $\boldsymbol{\theta}$ are the parameters underpinning the simulation. We define this as

$$P_t(\ell\,;\boldsymbol{\theta}) = \prod_{k\in\mathcal{I}_t}(1 - q_t(\ell,k))\prod_{l\in\mathcal{N}_t}(1 - \kappa q_t(\ell,l)), \tag{4.2.1}$$

where recall $q_t(\ell,k)$ is the probability individual k infects individual ℓ (with no restrictions) and κ denotes the reduction in an individual's infectiousness once they become notified. When $\kappa = 1$ there are no restrictions once notified, therefore this reduces to an extended SIR model. Similarly, when $\kappa = 0$ this is equivalent to there being no notification period, as with probability 1 we avoid infection from someone who has left the infectious state.

We shall assume that the population is only inhomogeneous with respect to the location of the individuals. Additionally, to represent the spatial effect we select an exponential distance kernel, which will be a function of the Euclidean distance. Therefore, for individual k we define

$$q_t(\ell,k) = \begin{cases} (1-p)e^{-\gamma d(\ell,k)} & \text{if } \ k \in \mathcal{I}_t \cup \mathcal{N}_t \\ \\ 0, & \text{if } \ k \in \mathcal{S}_t \cup \mathcal{R}_t \end{cases}, \tag{4.2.2}$$

where $d(x,y)$ is the Euclidean distance between individuals x and y. As such we have chosen a transmission probability dependent on three parameters:

- p, the base probability of avoiding infection,
- γ, a parameter determining how distance affects the probability of infection,
- κ, a parameter controlling how an individual's infectiousness changes once they enter the notification stage,

with $\boldsymbol{\theta} = (p, \gamma, \kappa)$. This infection probability will be used to simulate the outbreaks and be the assumed transmission model within the likelihood, highlighting the benefits of first testing methods on simulated data sets.

4.3 The Simulated Data Sets

In this chapter we will primarily focus on two simulated outbreaks, generated using the method defined in the previous section, with the conditions found in Table 4.1. In both examples we are working with a medium sized population, uniformly spread across a (1×1) square. The infection probability has a strong spatial effect (see Figure 4.1), therefore we would expect to witness infectious individuals infecting those susceptible individuals closest to them.

	N_{pop}	p	γ	κ	a	d	Population Distribution
SIR Simulation	500	0.975	15	-	3	-	$\text{Uniform}(0,1) \times \text{Uniform}(0,1)$
SINR Simulation	300	0.985	10	0.2	4	4	$\text{Uniform}(0,1) \times \text{Uniform}(0,1)$

Table 4.1: The settings used to generate the SIR and the SINR epidemics. The transmission parameters are $\boldsymbol{\theta} = (p, \gamma, \kappa)$, a is the infectious period parameter and d is the length of the notification period in the SINR example.

In Figures 4.2 and 4.4 we show the spread of those who were infected at regular intervals during both of the outbreaks. As we may expect the larger value of γ in the SIR outbreak produces an epidemic with a greater spatial effect. Both epidemics occur over many days, therefore they will be useful for testing if the SMC algorithm can perform well for the entirety of an outbreak. Finally, in Figures 4.3 and 4.5 we show how the number of individuals within each state changes over time. As we can see both of these outbreaks are fairly severe, with a significant portion of the population infected during the course of the outbreaks.

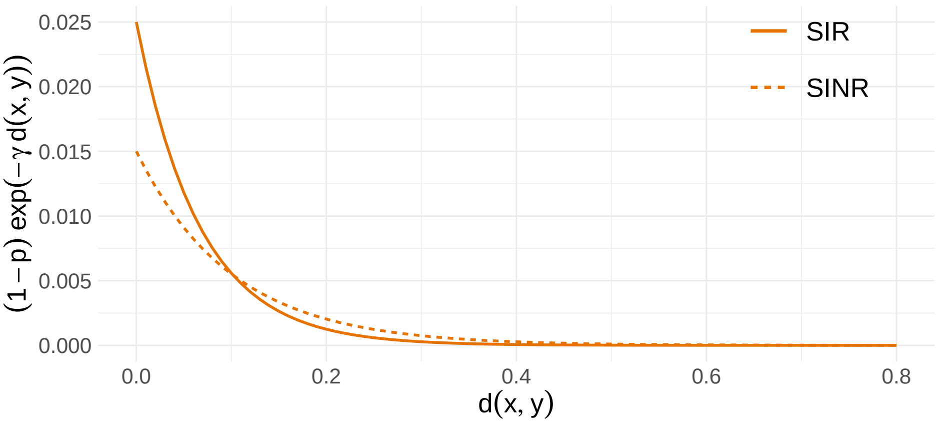


Figure 4.1: The transmission probability evaluated for different values of the Euclidean distance, $d(x, y)$, for both the SIR outbreak, with $\gamma = 15$ and $p = 0.975$, and the SINR outbreak, with $\gamma = 10$ and $p = 0.985$.

4.4 Comparison to MCMC Methods

We will be applying the SMC algorithm described in Section 3.4 to the simulated outbreak data we have generated. The aim is to produce estimates for parameters p and γ in both examples, as well as κ in the SINR example: these form the parameters $\boldsymbol{\theta}$. We will also be interested in the number of occult individuals at each time step, t, denoted by u_t. Throughout this chapter, unless stated otherwise, we assume that the infectious period parameter, a, is known, and therefore we are not interested in inferring it.

In both examples we initialise the SMC algorithm close to the start of the epidemic, denoted by time T. We then will take the algorithm forward to time $T + L$, which will be past the conclusion of the outbreak. Details of the state of the outbreak at the start and end of our application can be found in Table 4.2. We see that, for both examples, we are initialising the SMC algorithm with very few observations (small m_T^N and (or) m_T^R). This may, upon first inspection, appear overly ambitious, however, it will reflect the reality of when these algorithms will be most useful. If we can initiate the algorithm during the initial stages of an outbreak then this will allow for immediate information to be conveyed about the form of the epidemic and thus help inform on any control policies. Similarly there is use in checking the algorithm can be repeatedly applied, far forward in time, as it is likely that an outbreak will evolve and the ability to capture this would prove highly useful.

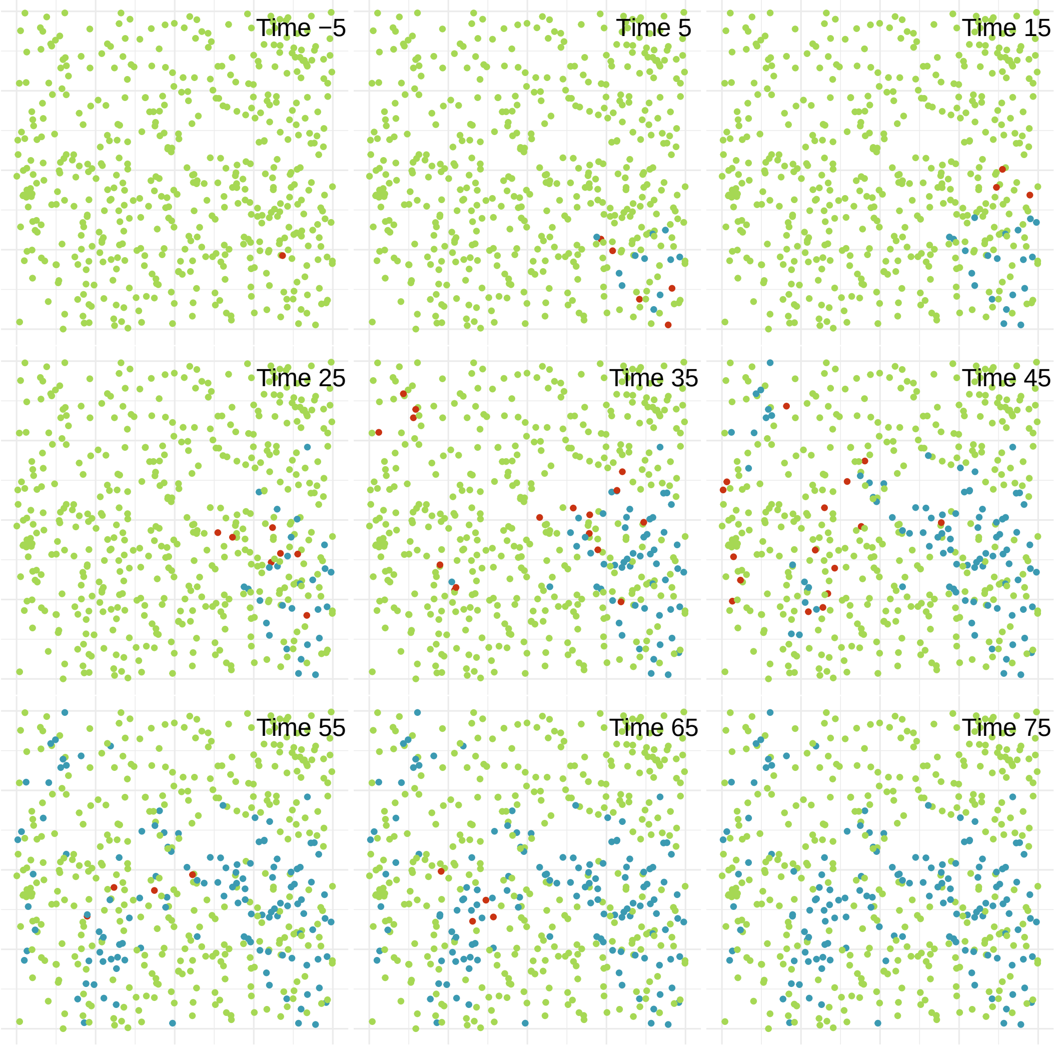


Figure 4.2: The location of each individual within the population on which the simulated SIR outbreak occurs. The colour indicates each individual's status at the current time step (top-right) as either susceptible (green), infectious (red) or removed (blue).

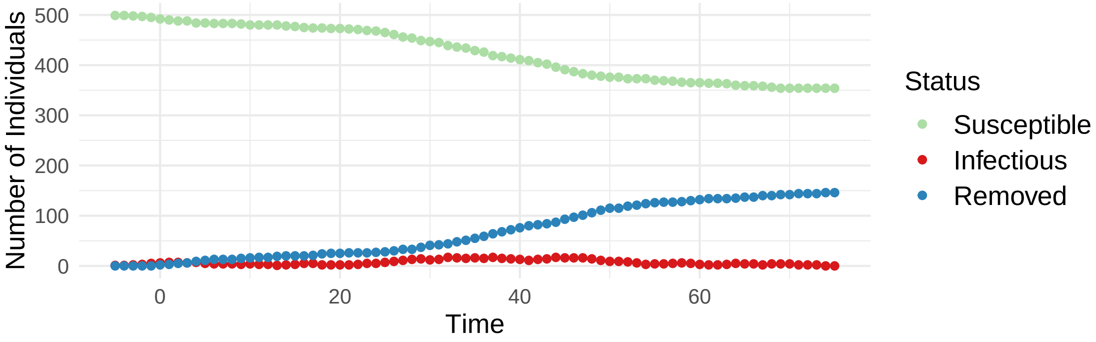


Figure 4.3: The number of individuals in each of the three states: 'Susceptible', 'Infectious', or 'Removed, at each time step of the SIR outbreak.

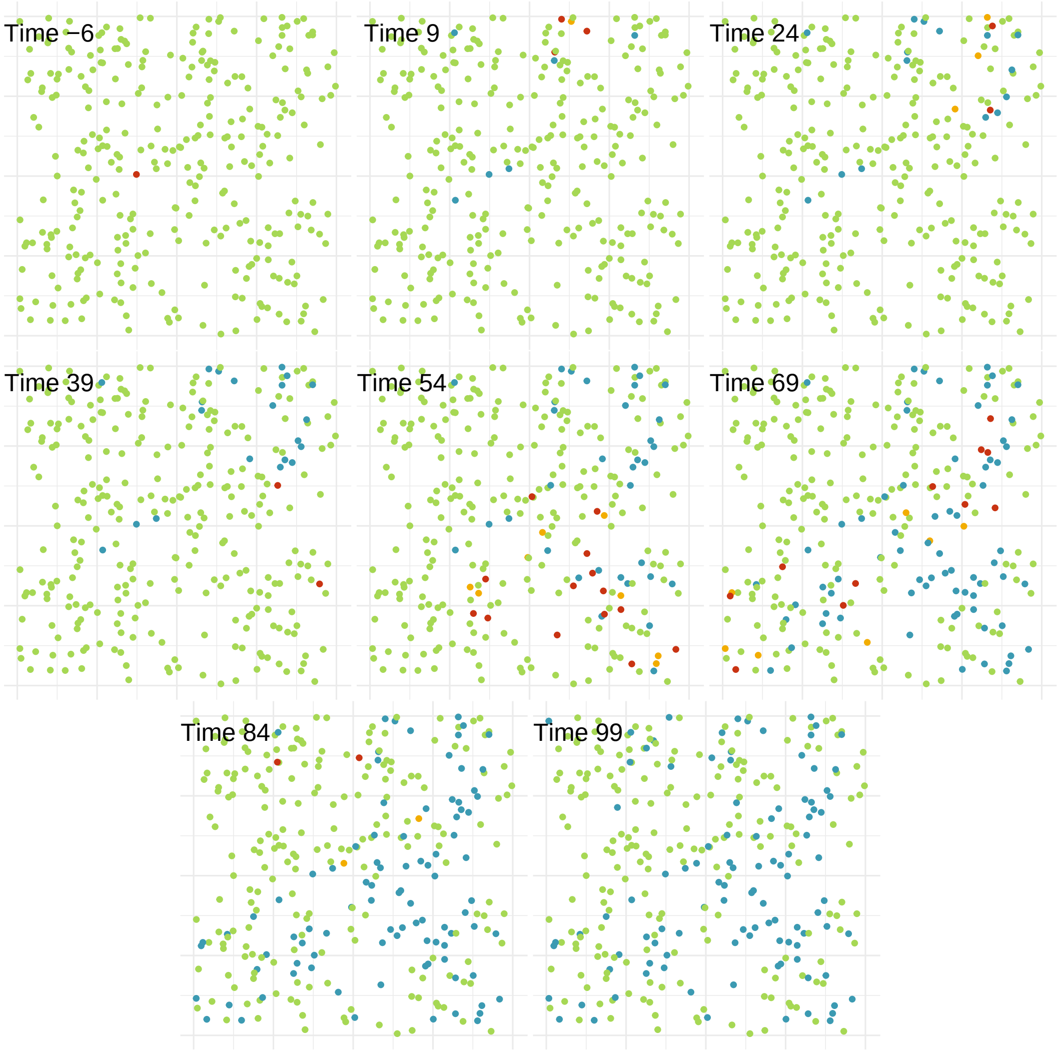


Figure 4.4: The location of each individual within the population on which the simulated SINR outbreak occurs. The colour indicates each individual's status at the current time step (top-left) as either susceptible (green), infectious (red), notified (yellow) or removed (blue).

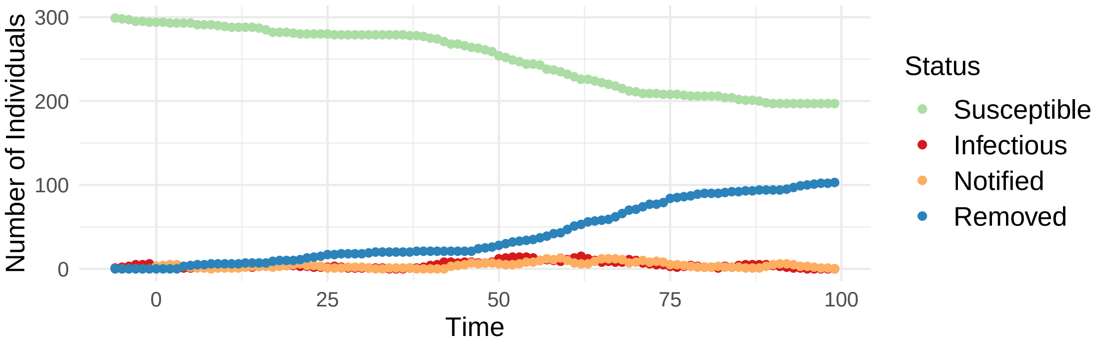


Figure 4.5: The number of individuals in each of the four states: 'Susceptible', 'Infectious', 'Notified', or 'Removed, at each time step of the SINR outbreak.

The first, and possibly the most important, question we need to answer is "*Do the particles represent a sample from the target distribution?*" Answering this is difficult as, by construction, the posterior distributions we are interested in are usually those for which analysis is not straightforward. As such, in order to answer this question, we will compare the SMC algorithm to the current gold-standard of stochastic epidemic modelling: data-augmented MCMC.

To initialise the SMC algorithm we use the MCMC algorithm described in Section 3.3 to generate $n = 1000$ initial particles at time T, using the prior distributions described in Table 4.3. These have been chosen to be fairly uninformative, as we wish to see how well the SMC works when the data is driving our inference.

We choose to update all of the transmission parameters together, using the RWM method from Section 1.4.4.2. Additionally, the MCMC will be adaptively tuned between iterations $b_1 = 100$ and $b_2 = 5000$, following the methods previously described in Section 3.3.6.1. Once generated these are the samples we then feed into the SMC algorithm and repeatedly transform, as we obtain new data. We will consider two SMC algorithms with either $n_p = 25$ or $n_p = 50$ iterations in the movement step. At the end of each iteration we will have $n = 1000$ particles approximately from the posterior distribution at that time step. We then choose to compare the SMC algorithm to an MCMC algorithm, which is provided with all of the data up to that time step and is applied with the same conditions as those used to initialise the SMC. Additionally for the comparable MCMC we use a burn-in of $b = 10000$ in both the SIR and SINR examples.

	T	m_T^N	m_T^R	$T+L$	m_{T+L}^N	m_{T+L}^R	T_{max}^N	T_{max}^R
SIR Simulation	3	-	6	80	-	146	-	74
SINR Simulation	3	5	0	105	103	103	95	99

Table 4.2: Information relating to the start and end points of when we apply the SMC algorithm to the SIR and SINR outbreaks. Here m_t^N and m_t^R denote the number of notified and removed individuals, respectively, at time t. Additionally we denote by T_{max}^N and T_{max}^R the time of the last notification and removal, respectively.

	p	γ	κ
SIR Simulation	Uniform(0, 1)	Gamma(1.69, 0.13)	-
SINR Simulation	Uniform(0, 1)	Gamma(2.25, 0.15)	Uniform(0, 1)

Table 4.3: The prior distributions used within the two simulated outbreaks.

4.4.1 The SIR Outbreak

We begin by considering the SIR example. We compare the results of the SMC algorithm to the corresponding MCMC, where the MCMC has been applied every 5 time steps (5, 10, . . .). A summary of the samples generated for each parameter, at various time steps, can be found in Figure 4.6 and Table 4.4.

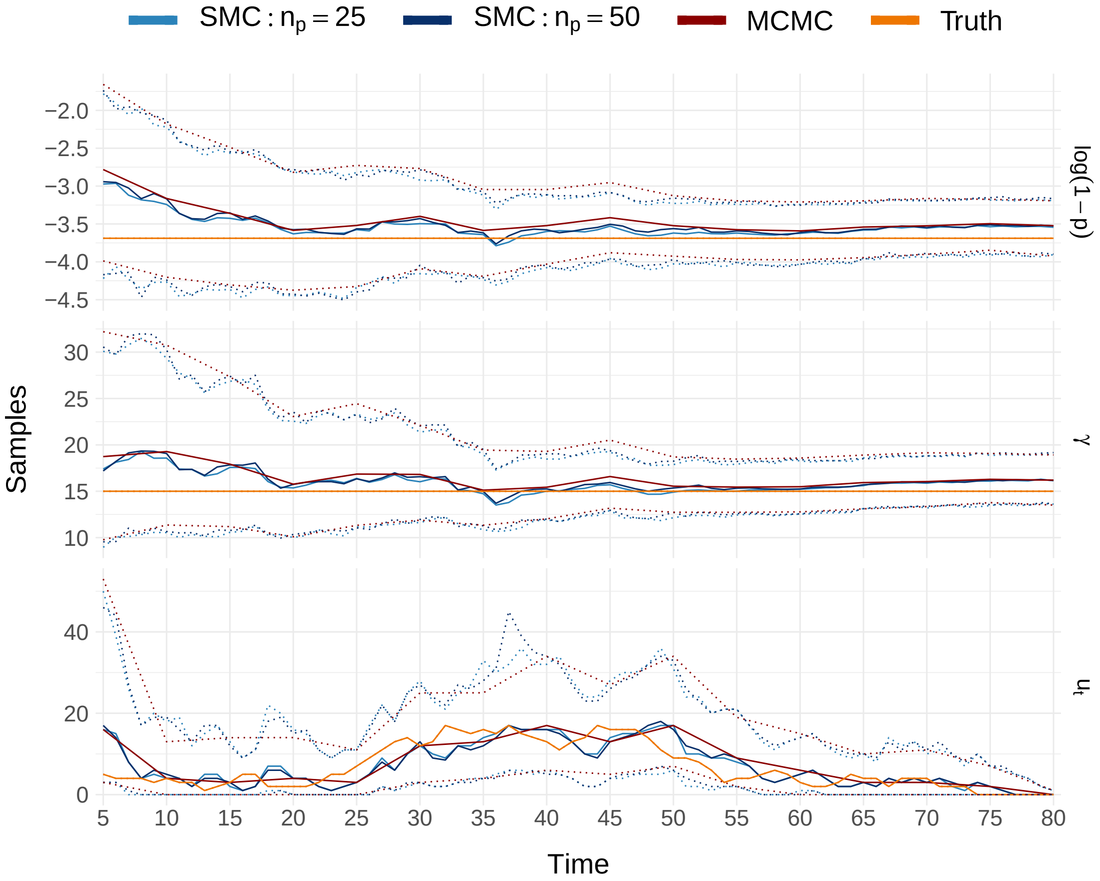


Figure 4.6: Lines showing the median (solid) and the lower (2.5%) and upper (97.5%) quantiles (dotted) of the samples, at each time step, generated using two runs of the SMC algorithm (blues) and an MCMC algorithm (red), as applied to the SIR outbreak. The SMC algorithm has been applied with movement step of length $n_p = 25$ and $n_p = 50$. Also shown are the true values (orange), which were used to simulate the outbreak. The SMC output is shown every time step and the MCMC every 5 times steps (5, 10, . . .).

		$t = 30$		$t = 55$		$t = 80$	
		Mean	S.D.	Mean	S.D.	Mean	S.D.
p	SMC: $n_p = 25$	0.969	0.00997	0.973	0.00533	0.971	0.00536
	SMC: $n_p = 50$	0.966	0.01100	0.972	0.00566	0.970	0.00544
	MCMC	0.965	0.01140	0.972	0.00575	0.970	0.00551
γ	SMC: $n_p = 25$	16.1	2.54	15.1	1.38	16.2	1.38
	SMC: $n_p = 50$	16.6	2.67	15.4	1.47	16.2	1.37
	MCMC	16.9	2.61	15.5	1.44	16.2	1.39
u_t	SMC: $n_p = 25$	13.4	6.48	8.89	4.88	0.061	0.435
	SMC: $n_p = 50$	13.4	6.06	9.40	4.65	0.058	0.350
	MCMC	12.5	5.64	9.28	4.40	0.079	0.519

Table 4.4: A comparison, at three time steps, of the mean and the standard deviation (S.D.) generated using MCMC and SMC methods on the SIR outbreak, where the latter method has been run with movement steps of length $n_p = 25$ and $n_p = 50$.

Focusing on the spatial parameter, γ, we can see in Figure 4.6 that the median does not change with the incorporation of the additional data. What we do observe, however, is the distribution of the particles beginning to peak (smaller variance), as we become more and more confident in the value γ takes. We can see similar behaviour for parameter p, the more data we have the more peaked the distribution becomes. If we consider u_t we see that this changes throughout time, reflecting the changing number of occult individuals at each time step. For this outbreak the time of the last removal is $t = 74$. We can see that the output is predicting this by estimating a declining number of occult individuals.

As we have simulated the outbreaks we know the true values of the underlying parameters. For γ, Table 4.4 shows that the average generated from both methods at first tends towards the truth ($\gamma^{TRUE} = 15$) before moving slightly away towards the end of the outbreak. Similarly the average for p gets closer in the initial stages before tending away from the truth ($p^{TRUE} = 0.975$) at the end of the outbreak. This is likely as with the construction of our model we expect p and γ to be highly correlated, such that if we have a higher base avoidance probability, p, then γ must be lower and vice versa. Finally, if we consider the value of u_t, we see that when compared to the truth both the SMC and MCMC are fairly successful in estimating the number of occult individuals, at each time step.

For both of the parameters and u_t, the algorithms appear to be in agreement. Additionally, we can see that the longer movement step is unnecessary in this example as $n_p = 25$ is matching very well with the output from the MCMC (see, Table 4.4). Altogether we can see that the SMC algorithm is successfully capturing the changes in the marginal distributions, matching the MCMC at each time step compared.

It is clear from the results displayed that the SMC and the MCMC algorithms are generating samples from approximately the same distribution. What we find most impressive is how closely the SMC and MCMC agree on the number of occult individuals. As this is a dynamic value we would expect it to be the hardest for the SMC algorithm to estimate, at each time step.

4.4.2 The SINR Outbreak

Next we consider the more complicated SINR outbreak. The SIR example was a fairly severe outbreak that infected a large number of individuals, over a relatively short space of time. The SINR outbreak involves fewer new infection cases, but does evolve over a longer period of time. This will challenge the SMC algorithm in a different manner, testing if it can repeatedly incorporate information far into the future.

Our aim is, as in the SIR example, to compare the output of the SMC algorithm to that generated using MCMC methods. There are now three parameters to consider: as well as p and γ, we are now also interested in the parameter κ, which represents the reduction in the infectiousness of an individual when they are notified. Additionally we will again be interested in inferring the number of occult individuals at each time step, u_t.

In Figure 4.7 and Table 4.5 we show a comparison of the MCMC and SMC methods. We can observe that, as in the previous example, the two algorithms agree. This is true even for the tails of the distributions, which we would expect to be harder to capture. Due to the additional notification period this epidemic exhibits different behaviour to the previous example. Nevertheless the SMC is still performing strongly, this is again true for the shorter movement step ($n_p = 25$).

Both the SMC and MCMC produce estimates for the parameters which are close to the truth. This is also true for u_t, whose evolution is correctly estimated. The value of κ is harder to pick up for both algorithms and results in the least agreement between the two methods. This is likely a consequence of how we have generated the simulated data, resulting in there being only a small amount of data provided to either algorithm in which to distinguish the parameters.

Returning to our initial aim, the second question we wish to answer is "*Does the performance of the SMC algorithm deteriorate when run for multiple time steps?*" In this example the SMC algorithm is successfully applied up to the conclusion of the outbreak, over many iterations. We see that the accuracy of the SMC does not deteriorate, even over many successive iterations.

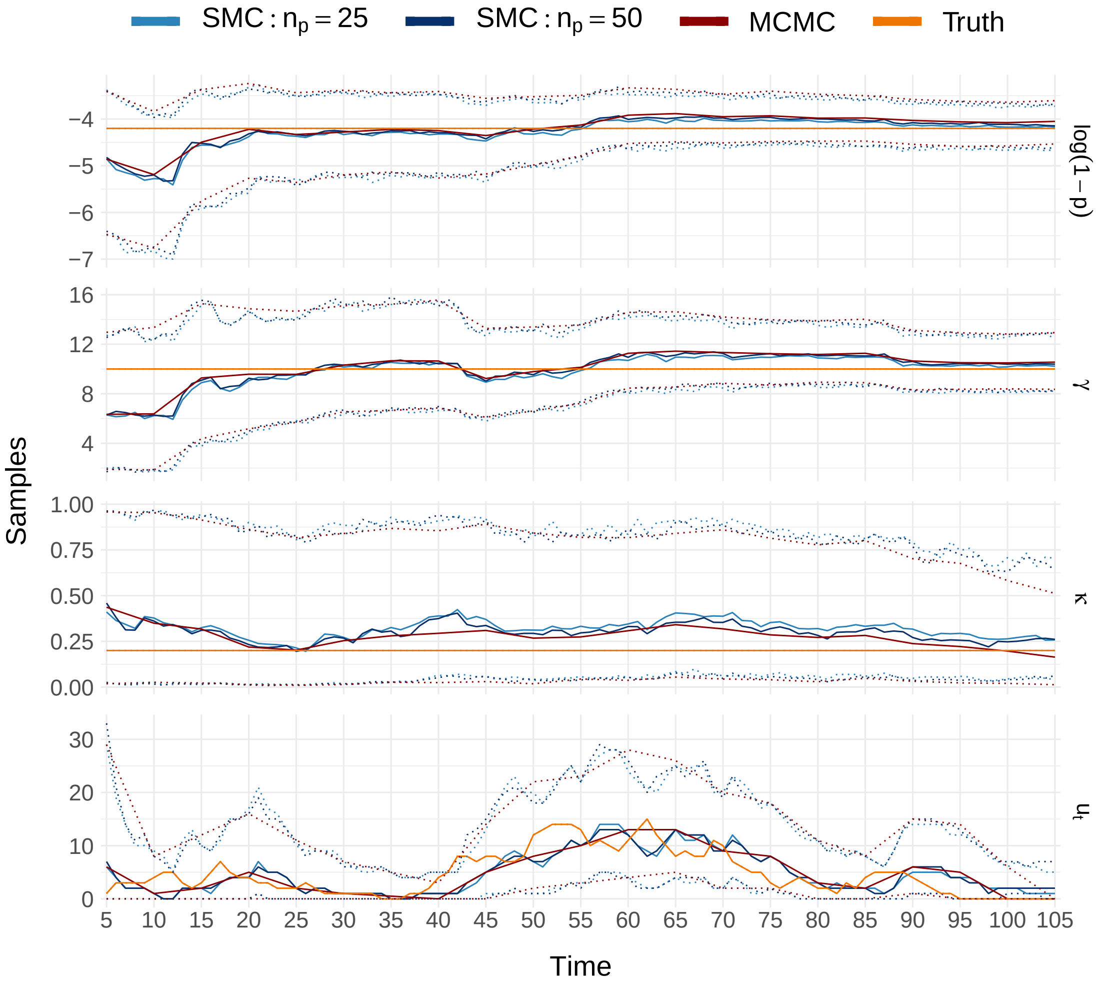


Figure 4.7: Lines showing the median (solid) and the lower (2.5%) and upper (97.5%) quantiles (dotted) of the samples, at each time step, generated using two runs of the SMC algorithm (blues) and an MCMC algorithm (red), as applied to the SINR outbreak. The SMC algorithm has been applied with movement step of length $n_p = 25$ and $n_p = 50$. Also shown are the true values (orange). The SMC output is shown every time step and the MCMC every 5 times steps (5, 10, . . .).

Linked to this we are also interested in answering, "*How does the SMC algorithm perform when the new data significantly change the shape of the distribution?*" In Figure

		$\boldsymbol{t = 25}$		$\boldsymbol{t = 65}$		$\boldsymbol{t = 105}$	
		Mean	S.D.	Mean	S.D.	Mean	S.D.
p	SMC: $n_p = 25$	0.986	0.00686	0.981	0.00525	0.984	0.00378
	SMC: $n_p = 50$	0.986	0.00674	0.981	0.00543	0.984	0.00391
	MCMC	0.985	0.00696	0.979	0.0059	0.982	0.00415
γ	SMC: $n_p = 25$	9.65	2.17	11.0	1.43	10.3	1.13
	SMC: $n_p = 50$	9.64	2.17	11.1	1.45	10.4	1.19
	MCMC	9.71	2.19	11.5	1.48	10.6	1.15
κ	SMC: $n_p = 25$	0.264	0.205	0.431	0.223	0.284	0.166
	SMC: $n_p = 50$	0.256	0.213	0.394	0.217	0.280	0.150
	MCMC	0.258	0.213	0.372	0.212	0.188	0.131
u_t	SMC: $n_p = 25$	3.04	2.90	13.0	5.41	1.240	1.600
	SMC: $n_p = 50$	3.07	3.05	13.2	5.40	2.080	1.750
	MCMC	3.16	3.01	13.6	5.44	0.031	0.358

Table 4.5: Comparison of the mean and the standard deviation (S.D.) generated using SMC methods, with two values for n_p, and MCMC methods, as applied to the SINR outbreak.

4.8 we show the density plots of the final samples generated by the SMC, at the end of the outbreak, and compare them to the samples used to initiate the SMC algorithm. With the incorporation of the new data, we can see that the SMC method has had to account for significant changes in the shape and location of the posterior distribution. This has not affected the accuracy of the output and the SMC performs as well as the MCMC, which is allowed to 'restart' with each new piece of data. This is true for each time step compared.

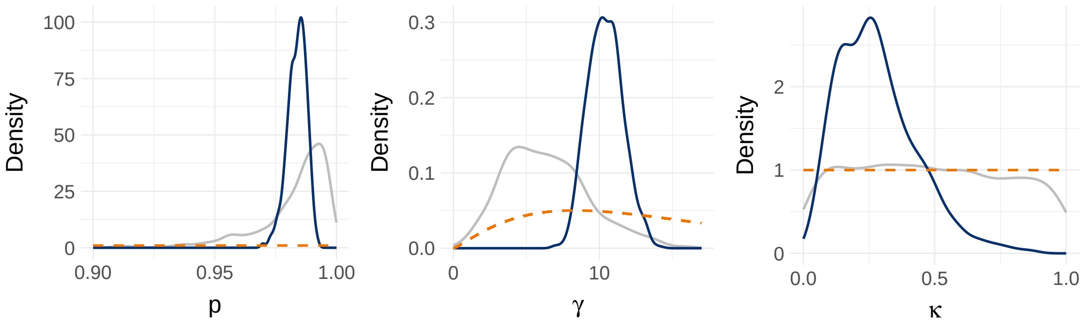


Figure 4.8: The density plots generated at the end of the SINR outbreak, using SMC methods (solid blue) applied from the start of the outbreak. We also include the initial particles used to seed the SMC (solid grey) and the prior distributions used (dashed orange).

4.4.3 Conclusions

In this section we have illustrated that the SMC algorithm can repeatedly generate samples from a target posterior distribution that evolves with time. These samples match the analogous MCMC, which is able to restart and be applied with all of the observed data. This illustrates that the combination of steps we described in the previous chapter are sufficient to transform the samples in such a way that they represent the evolving posterior distribution.

Overall this is a strong start to better understanding the SMC algorithm we have constructed. In the following sections we aim to look closer at how it is working, with the aim of learning of any strengths or weaknesses it may have.

4.5 Comparison of Computation Time

In the previous section we concluded that the SMC produces samples that are comparable to those generated using MCMC methods. The next step therefore is to test the statement we made previously and answer the question: *"Is the SMC algorithm computationally faster than the analogous MCMC?"*

As is the nature of simulation methods, there are many choices about how we apply the MCMC and SMC algorithms, that will impact their computation time. Therefore we stress that throughout this section we aim to only provide an illustration of the comparative computation time of both methods.

4.5.1 Estimating the Computation Time

We choose to estimate the run-time of each method, under the condition that we can run X jobs in parallel. This is to illustrate that often we will not have access to resources that allow for full parallelization, instead we might only be able to split the jobs onto a limited number of processors. In practice, if we had unlimited resources, the maximum number of parallel jobs we could possibly run is $X_{max} = n$, where n is the number of particles we have generated. For our examples $X_{max} = 1000$.

We are interested in the time it takes to sample n particles from the posterior distribution of interest, at time t. One option is to run an MCMC using all of the data observed up to time t. Timing this accurately is difficult as the time it takes to run will depend on various factors. As our interest is in testing if the SMC is quicker we choose to underestimate the time it takes to complete the MCMC and overestimate the time for the SMC. This should allow us to observe the minimum speed increase we could obtain by using SMC methods instead of MCMC. Therefore when timing the MCMC we choose to only record the time it takes to complete each MCMC algorithms burn-in period. Thus the conditions of applying the MCMC are that we have access to resources which allow 1000 independent chains (the maximum) to run in parallel, and we then select the first value from each, once the burn-in has been completed. This is not the true setting in which we would run it, rather this will serve as a lower bound estimate of the true time it would take to run the MCMC algorithm.

Considering the SMC, we recall that it has several steps, some of which are parallelizable and some which are not. For those which can be run in parallel, we assume that we split the n particles into X groups, each with n/X samples. Within each of the groups that step of the SMC is performed in serial, therefore this stage of the SMC is completed when the last group has finished. Additionally, for the SMC we choose to over-estimate the runtime, again in an effort to illustrate the minimum possible speed increase. Therefore, in the estimates to follow we assume that the longest running jobs all occur within the same group. This is unlikely to be the case and as such this will represent an upper bound for the time it takes to complete a single iteration of the SMC.

Suppose we denote by $T_z^{(t)}$ the time it takes to complete step z of the SMC, at time t, where this will be a vector with n elements if this step can be parallelized. We split the timing of the SMC algorithm into the following parts:

$T_{Data}^{(t)}$ Initialise each iteration i.e. the inputting/collecting of the new data.

$T_{Weight}^{(t,j)}$ Adjust and find the weight of particle j, with $T_{Weight}^{(t)} = \left(T_{Weight}^{(t,1)}, \dots, T_{Weight}^{(t,n)}\right)$.

$T_{Resample}^{(t)}$ Resample the particles.

$T_{Augment}^{(t,j)}$ Augment particle j, with $T_{Augment}^{(t)} = \left(T_{Augment}^{(t,1)}, \dots, T_{Augment}^{(t,n)}\right)$.

$T_{Tune}^{(t)}$ Adaptively tune the algorithm, e.g. calculate the new covariance matrix.

$T_{Move}^{(t,j)}$ Move particle j, with $T_{Move}^{(t)} = \left(T_{Move}^{(t,1)}, \dots, T_{Move}^{(t,n)}\right)$.

$T_{Collect}^{(t)}$ Collect together the moved particles.

These steps match those discussed in Section 3.4.9, where we summarised the SMC algorithm. We have split the timing in this way as we have serial components and embarrassingly parallel components, that require no communication between each other. Additionally, we do not incorporate the time required to initially set the code running, as this will be similar for both methods and negligible compared to the computation time of the algorithms.

We define $S^{(t)}_{Weight}$ as the set $T^{(t)}_{Weight}$ placed in descending order, such that $S^{(t,k)}_{Weight}$ is the time it takes to calculate the weight for the particle with the k^{th} longest weight computation time. We define $S_{Augment}$ and S_{Move} in a similar way. Therefore, under our assumption that the longest jobs all occur in the same batch, the total time it takes to complete iteration t of the SMC algorithm is defined as

$$T^{(t)}_{Data} + \sum_{k=1}^{n/X} S^{(t,k)}_{Weight} + T^{(t)}_{Resample} + \sum_{k=1}^{n/X} S^{(t,k)}_{Augment} + T^{(t)}_{Tune} + \sum_{k=1}^{n/X} S^{(t,k)}_{Move} + T^{(t)}_{Collect} \quad (4.5.1)$$

where recall X is the number of parallel jobs we can run. For ease we have assumed $n/X \in \mathbb{N}$, this is not necessary and the calculation can easily extend.

4.5.2 Simulation Examples

We consider the timings for both the SIR and SINR outbreaks discussed previously, the results can be seen in Figure 4.9. As we can see, even overestimating the time the SMC algorithm will take and underestimating the MCMC, with a relatively small X the SMC can be computed much quicker than the corresponding MCMC algorithm. For example, we found previously that $n_p = 25$ is sufficient for accurate analysis and we see that even with $X = 5$ the SMC is quicker than the MCMC. As such we have shown that not only does the SMC algorithm produce similar results to the MCMC algorithm, but it can do so in a much quicker time frame.

The longer the movement step the longer each iteration of the SMC will take, however, this is negligible for larger X. We should note that this is only an estimate of the time-to-compute where, throughout, we have aimed to show the minimum speed increase possible when using SMC methods as opposed to MCMC. In practice the SMC is found to be much quicker, especially when we deal with more realistic data which requires a significantly longer burn-in period.

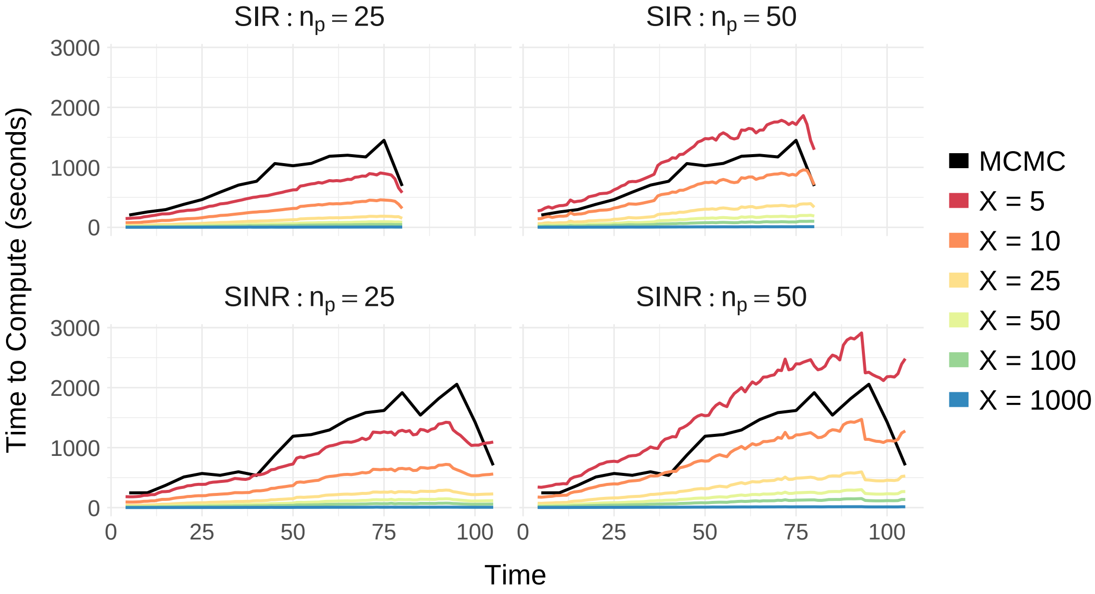


Figure 4.9: A comparison of the (estimated) time taken to apply both the SMC and MCMC methods to the SIR and SINR outbreaks, where the former is applied with two values of n_p. The MCMC (black) is run with a burn-in of $b = 10000$ and has been applied every 5 time-steps (5, 10, ...). The SMC has been applied at every time step and split into X jobs, where different values of X are represented by a different colour.

Finally we note that, as we can see in Figure 4.9, the length of the time to compute both the SMC and MCMC algorithm, on average, increases through time. This is for two reasons: firstly the more data we have the more time it will take to process and, secondly, when calculating the likelihood we need to consider a greater number of time steps. As we can observe there are some deviations from this trend, for example the MCMC algorithms applied at the end of the outbreak. This is as at these time steps we are estimating a smaller number of occult individuals, therefore the dimension of the problem is reduced and thus this results in a quicker computation time. Overall the greater number of occult individuals the longer the calculations will take, however, this is true for both methods.

4.5.3 Summary

In general the movement step of the SMC is the time-limiting factor of this method. Therefore for the SMC to be computationally quicker, we just require $n_p \times (n/X)$ to be less than the number of iterations in the burn-in of the MCMC. This is key to remember, as in later sections we choose n_p to be much higher; however, this still results in a significant speed increase over the corresponding MCMC algorithm, which also requires a longer burn-in.

4.6 The Length of the Movement Step

So far we have illustrated the effectiveness of the SMC algorithm in comparison to its MCMC counterpart. The algorithm appears stable to uninformative choices of prior distribution and the requirement to be run multiple time steps into the future. Next we are interested in how the length of the movement step impacts on the output of the SMC algorithm and its agreement with the MCMC algorithm. Thus we move on to answering the question of "*How many iterations are required in the movement step within the SMC algorithm, to produce accurate results?*" We have shown in the previous sections that a movement chain of length $n_p = 25$ is enough to generate samples from close to the target distribution. In this section we consider if it is possible to use a smaller value of n_p. This is of interest as the movement step of the SMC algorithm is the most computationally expensive, thus reducing it will allow for faster inference.

We consider the SINR outbreak previously discussed. We run the SMC and MCMC algorithms as previously, only now for the SMC we consider $n_p = 1, 5, 10, 25, 50$. A summary of the results can been seen in Figure 4.10.

We can see from Figure 4.10 that with $n_p = 1$ the parameter outputs produced using the SMC and MCMC algorithms are not in agreement for many time steps. However, u_t is well estimated by the SMC algorithm run with a shorter movement step, matching the MCMC algorithm even with very small values of n_p. This is perhaps as the assumptions we have had to make within the SMC algorithm will impact the transmission parameters rather than the number of occults we infer, which we aimed to keep the same. For the parameters we see that, as we expect, the increase in n_p leads to the SMC and MCMC having greater agreement. Even so, we can see that with $n_p = 5$ the estimates from both algorithms are fairly close.

Figure 4.10 highlights the hybrid nature of the SMC algorithm we have developed. The smaller value of n_p produces an algorithm closer to the sequential-importance-resampling algorithm, whereas larger n_p produces an algorithm closer to an MCMC algorithm. We observe that the shorter movement steps appear to begin well, and it is only after multiple iterations that the accuracy deteriorates. This is likely as any error

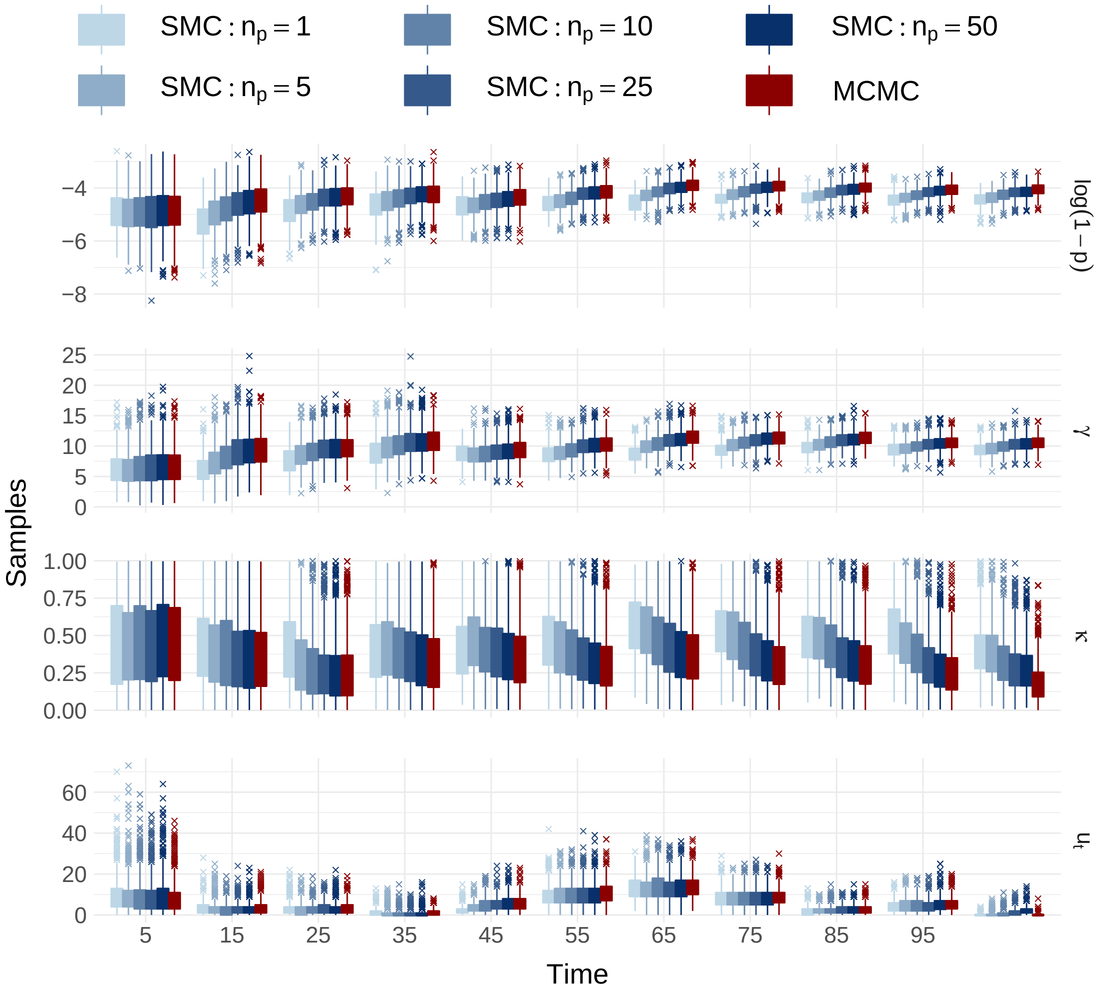


Figure 4.10: A comparison of the particles generated using SMC methods with different values of n_p (blues) and MCMC methods (red), as applied to the SINR outbreak.

in our weighting will be compounded the more iterations that are run. Additionally the particle degeneration associated with sequential-importance-resampling algorithms will be worse for smaller n_p. In conclusion, we can see that the accuracy of the SMC algorithm is highly dependent on the length of the movement step. Thus to apply the methods one must balance an increase in accuracy with the increased cost of running the algorithm.

4.7 Monitoring the Acceptance Rate

Another fundamental question we wish to answer is "*Can the SMC maintain reasonable efficiency, with no further human interaction necessary?*" Ideally the SMC algorithm will continue to incorporate the new data and produce accurate output, with little need for further (human) intervention.

Of particular interest is tracking the acceptance probability of the movement step within the SMC algorithm. From previous discussions, we deemed a ‘reasonable’ acceptance rate to be around 25%. For the parameters this is achieved using the Gaussian random-walk Metropolis algorithm defined previously. This requires estimation of the covariance matrix of the parameters, which for the SMC was recalculated prior to the movement step. Additionally, for the unobserved data, if the acceptance rate drifted too low or too high (outside of $(0.25 - 0.15, 0.25 + 0.15)$), the algorithm is designed to automatically update the tuning parameters in order to improve the acceptance rate. Therefore in this section we check to see if this method of automating the SMC has been successful.

As we have n different MCMC runs, each with n_p iterations, we track the average acceptance rate across all $n \times n_p$ iterations. Recall that there are four updating steps in the MCMC algorithm, therefore we monitor the acceptance rate for each:

- AR_θ, updating the parameters.
- AR_i, updating the infections times of the observed removals.
- AR_{o_1}, updating the times of the occult individuals.
- AR_{o_2}, updating which individuals are the occults.

The average acceptance rates across each particles movement step, at each time step, for the SIR and the SINR outbreaks, can be found in Figure 4.11. As we can see the acceptance rates for each updating step mostly stay within the desired rates. The rate for the changing of the occult individuals is often quite high; however, this is unavoidable as even if we update every occult individual we obtain a high acceptance rate. This is likely due to the fact that the estimated number of occult individuals, u_t, is fairly low throughout these outbreaks, therefore updating all of the occults is unlikely to change the likelihood significantly.

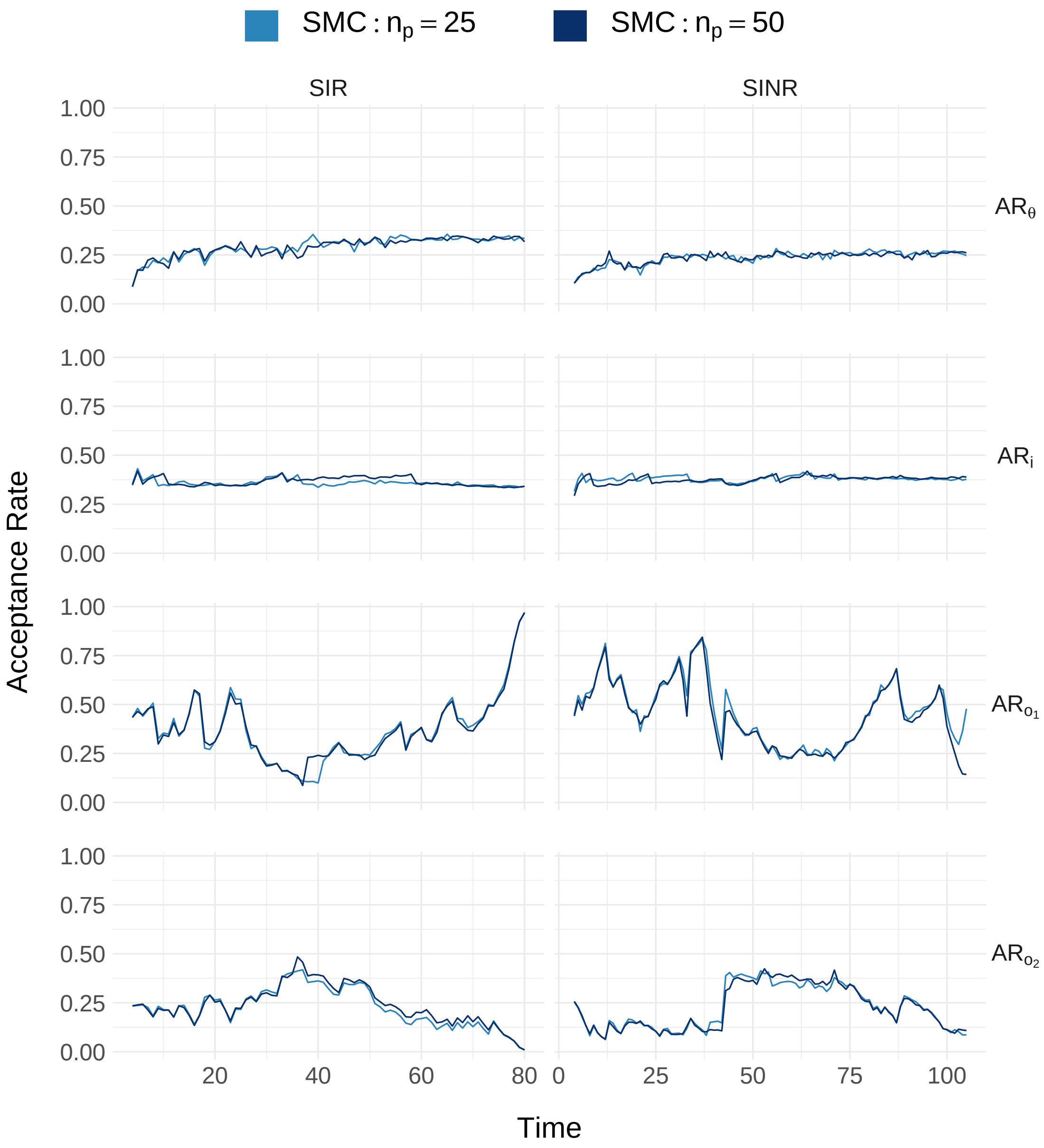


Figure 4.11: The average acceptance rates for each proposal step in the MCMC movement step applied to each particle within the SMC algorithm. We consider both the SIR and the SINR simulated outbreaks.

4.8 Particle Degeneracy

The next question we wish to answer is "*How severe is the particle degeneracy and does this affect the results?*" One way of monitoring this is to record the number of unique particles produced after the resampling step, in each iteration of the SMC algorithm. We display this in Figure 4.12, for both the SIR and SINR outbreaks. We see that the number of unique samples remains steadily high throughout the iterations. We also can see that the increased length of the movement step does not appear to reduce the degeneracy, once we introduce some movement (here $n_p > 5$).

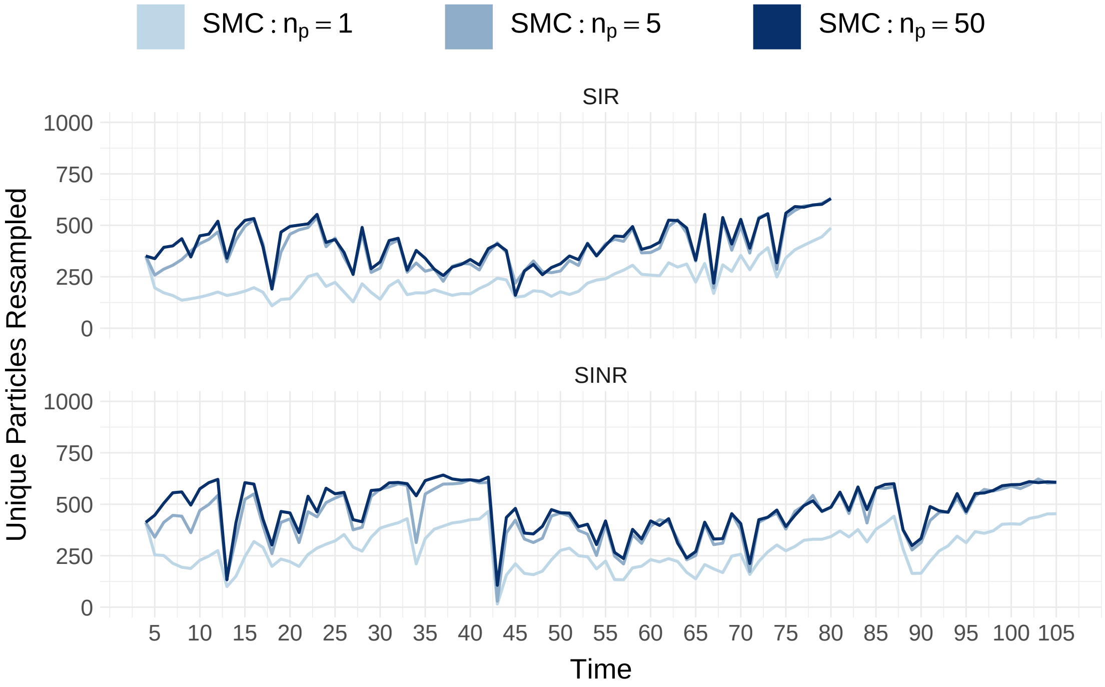


Figure 4.12: The number of unique particles resampled in each iteration of the SMC algorithm during the resampling step, when different lengths of the movement step (n_p) are considered.

Achieving a high number of particles resampled at each iteration is important for ensuring we have a diverse set of samples and thus, together with the movement step, do not suffer from severe particle degeneracy.

4.8.1 Relationship to the Number of New Observations

For the simulations considered, we find that the number of unique particles sampled is directly related to the number of new observations. If we consider $n_p = 50$ then, for the SIR outbreak, there is a correlation of -0.87 between the number of new removals observed and the number of unique particles sampled. Similarly for the SINR outbreak there is a correlation of -0.88 between the number of new notifications and the number of unique particles sampled.

These results are to be expected as, firstly, with more data we expect a greater change to the posterior distribution and thus we are more likely to produce particles with a very small weight. Secondly, with a greater number of new observations we will need to make more significant adjustments to each particle to ensure that they are consistent. This may also result in a greater number of particles with smaller weights. Overall, this implies that if we need to incorporate large amounts of new data, within a single iteration, then we are likely to suffer from a greater level of particle degeneracy. Although the movement step can reduce the impact of this, a high level of particle degeneracy is likely to result in particles which do not represent the posterior distribution.

4.9 The Adjustment Step

Previously in Sections 3.4.3–3.4.5 we introduced the adjustment step, which ensures the particles are consistent with the new data. In this section we consider the question *"Is the adjustment step necessary?"* To answer this we consider the SIR outbreak and apply the same methods as discussed previously, only without the adjustment step within the SMC algorithm.

The results are shown in Figure 4.13, where we have applied the algorithm up to the time step at which it fails (no particles match the new data), immediately illustrating the necessity of the adjustment step. Additionally, we can see that we achieve very poor matching between the SMC and the comparable MCMC. At each time step, the SMC estimates u_t as much higher than the MCMC estimate. This is likely as those particles most likely to match with the data are those with the highest number of proposed occults. Thus we resample an increasing number of particles with a higher value of u_t.

The poor matching between the two methods has been compounded by the tuning that occurs within the SMC algorithm, which occurs after resampling and thus can result in very small proposal steps within the movement step, if the variance of the particles is low. We can see this when considering the estimation of the parameters produced by the SMC: these suddenly converge to a single point, as in the previous iteration we only resampled a single unique particle. We could reduce this problem by choosing a more appropriate RWM algorithm, however in practice we found that the SMC still performed poorly.

In Figure 4.14 we can see that there is significant correlation between the number of unique particles resampled and the number of new observations. Additionally, we can see that in many iterations we lose a significant number of the particles during the resampling step of the SMC. This again illustrates the necessity of the adjustment step.

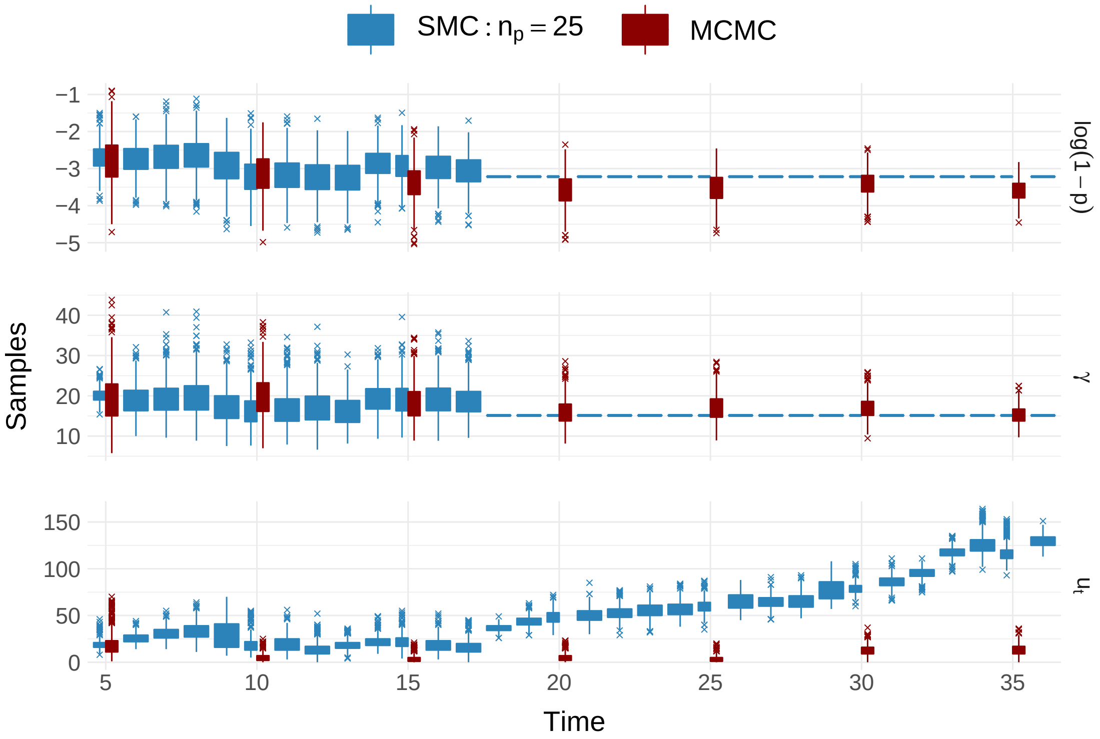


Figure 4.13: A comparison of the output generated using MCMC (red) and SMC methods with $n_p = 25$ (blue), where the latter no longer has the particle adjustment step.

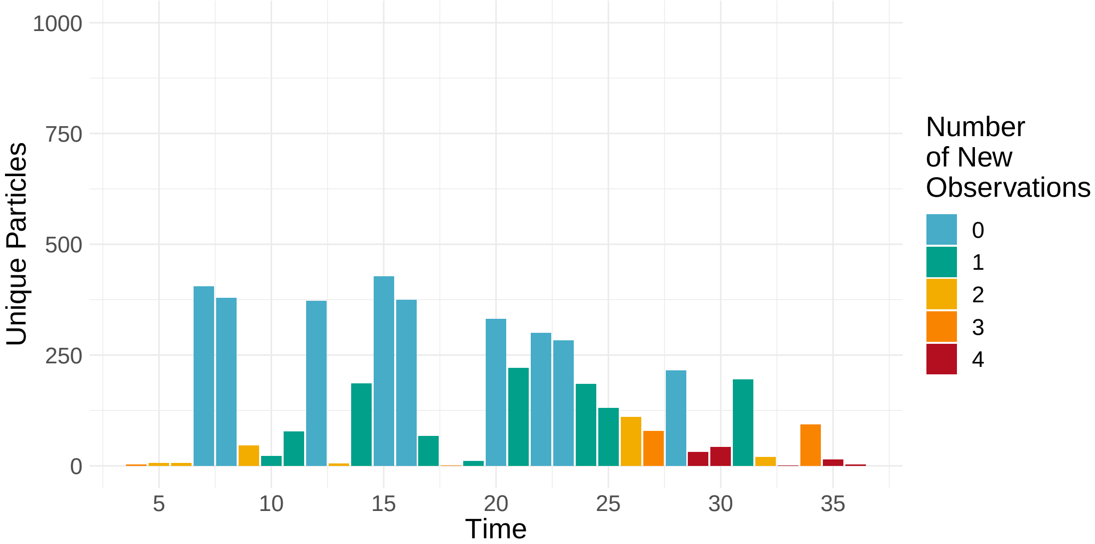


Figure 4.14: The number of unique particles resampled in each iteration, when using the SMC algorithm with no adjustment step and $n_p = 25$. The colour of each bar represents the number of new removals observed at that time step.

Overall, the key issue with choosing to not adjust the particles so that they are consistent with the new data is that the SMC will, almost always, fail at some time step, t. This makes it a method which is unsuitable for most applications. Additionally, this simulation is only a simple example, in epidemics when we have many new cases per day the degeneracy will be even more severe and failure is likely to occur much sooner.

4.10 The Wrong Kernel

Throughout, when modelling the outbreaks, we have used the same transmission mechanism when forming the posterior distribution, as we did in the simulation. However, when considering real data we will not know the true mechanism, therefore in this section we are concerned with answering the question "*How is the accuracy of our analysis affected if we use the wrong transmission model?*"

Throughout we have used an exponential distance kernel, $\exp\{-\gamma d(i,j)\}$, to both simulate and model the epidemic. In this section we instead consider the following:

- We simulate the epidemic with an exponential kernel, but model it assuming a Gaussian kernel of the form $\exp\{-(\gamma d(i,j))^2\}$.
- We simulate the epidemic with an Gaussian kernel, but model it assuming an exponential kernel.

We expect the parameters p and γ to be dependent on how we model the outbreak, however, what will be interesting is how the value of u_t is affected by using the wrong model.

We consider two simulated SINR outbreaks, which will be different to those we have considered thus far. This is as we wish to test how important the distance kernel is on outbreaks which are highly spatial. The new outbreaks each occur on a population existing on a 10×1 square, to exaggerate the spatial aspect of the simulated outbreaks. For both examples we use the same prior distributions for κ and p as in the previous SINR example, however, due to the population existing on a space of greater area, we now use a Gamma$(0.9, 0.3)$ prior for γ in both examples.

4.10.1 True Exponential Kernel

The first outbreak we consider has a true exponential kernel, therefore it is the same as the previous simulations with underlying transmission probability

$$q_t(\ell,k) = \begin{cases} (1-p)e^{-\gamma d(\ell,k)} & \text{if } \; k \in \mathcal{I}_t \cup \mathcal{N}_t \\ 0, & \text{if } \; k \in \mathcal{S}_t \cup \mathcal{R}_t \end{cases}. \tag{4.10.1}$$

We simulate this outbreak with $p = 0.9$, $\gamma = 3.25$ and $\kappa = 0.2$, additionally this epidemic occurred on a population of size $N_{pop} = 100$. We will compare the output of two SMC algorithms, with movement step $n_p = 25$. The first assumes an exponential kernel (correctly matching the truth) and the second assumes a Gaussian kernel. We display a comparison of the result in Figure 4.15.

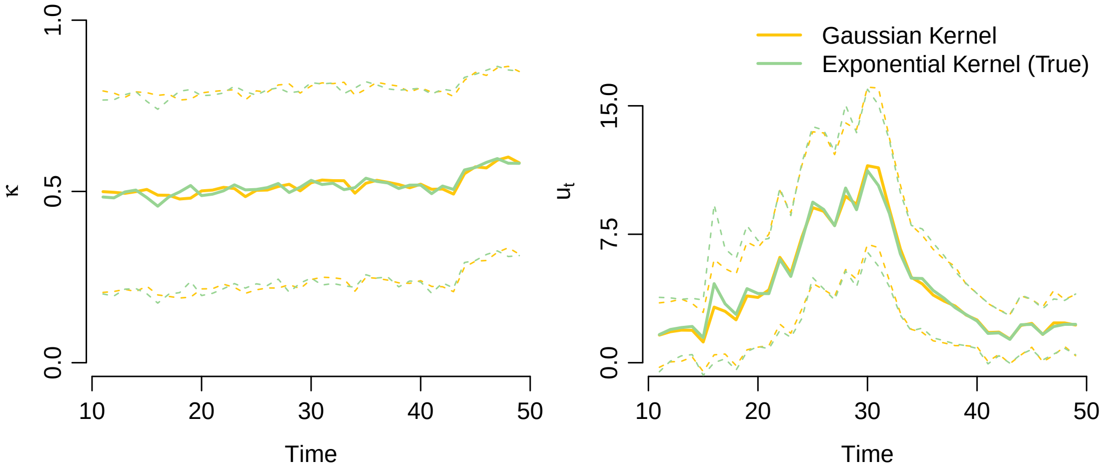


Figure 4.15: The mean values (solid) for each parameter generated using the SMC algorithm which uses either an exponential or a Gaussian distance kernel. Also shown is the mean ± the standard deviation (dashed).

We have not included the estimates for the values of γ and p as these depend on which kernel we are using and therefore are not directly comparable. However, what is surprising is that the value of κ and u_t match very well for the two different kernels used. Therefore, despite this being a spatial epidemic, the wrong model has not affected our estimation of the other parameters.

4.10.2 True Gaussian Kernel

We simulate the second outbreak using the underlying transmission probability

$$q_t(\ell, k) = \begin{cases} (1-p)e^{-(\gamma d(\ell,k))^2} & \text{if } \ k \in \mathcal{I}_t \cup \mathcal{N}_t \\ 0, & \text{if } \ k \in \mathcal{S}_t \cup \mathcal{R}_t \end{cases}. \tag{4.10.2}$$

This is therefore the same as in the previous simulations, only now the transmission probability is a function of the squared Euclidean distance between individuals.

We simulate this outbreak with $p = 0.991$, $\gamma = 1$ and $\kappa = 0.2$, additionally this outbreak occurred on a population of size $N_{pop} = 250$. We analyse this outbreak in the same way as the first example, and the results are shown in Figure 4.16.

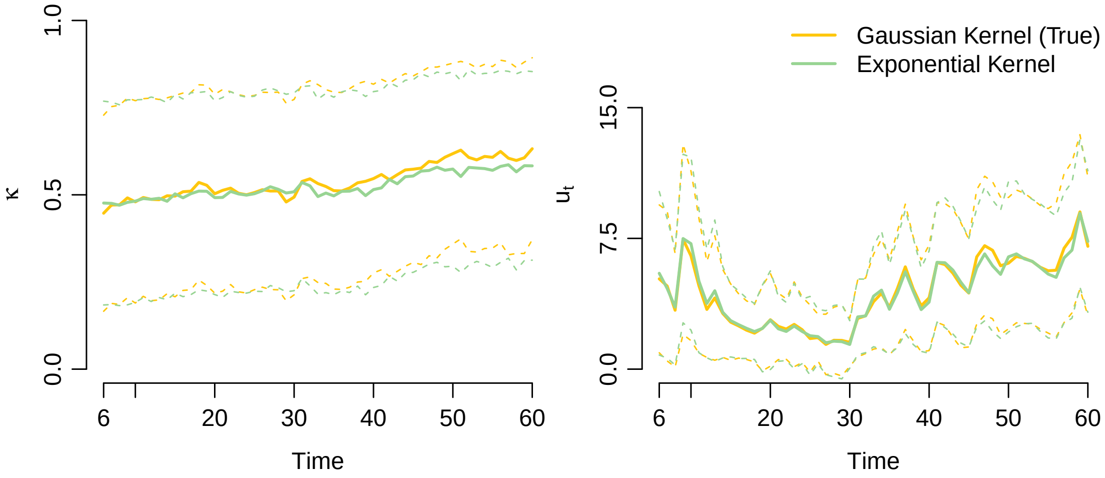
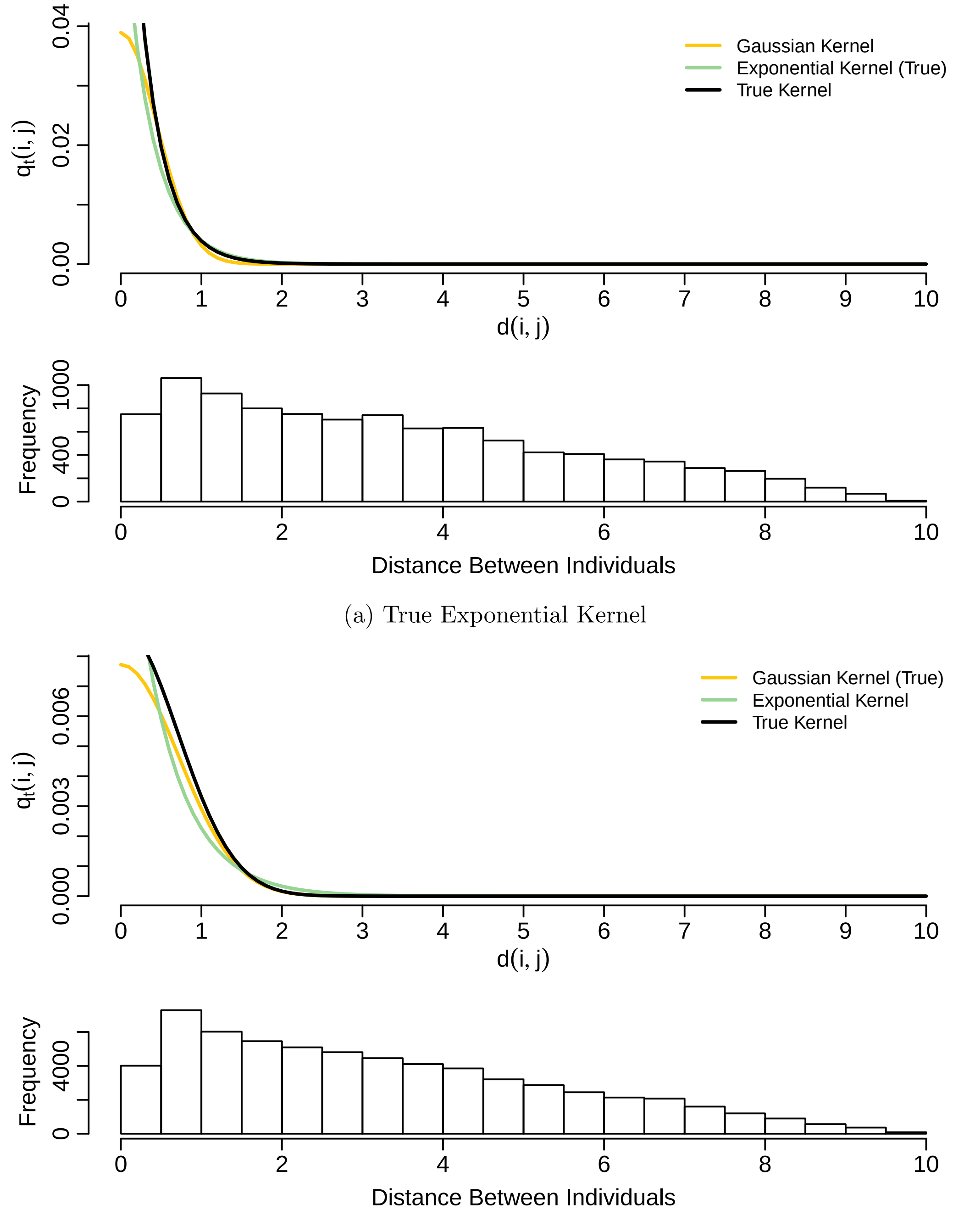


Figure 4.16: The mean values (solid) for each parameter generated using the SMC algorithm which uses either an exponential or a Gaussian distance kernel. Also shown is the mean ± the standard deviation (dashed).

As in the previous example, the values of p and γ depend on which model we have used, however, we can once again see that κ and u_t do not. This is reassuring as it means that the choice of distance kernel will not significantly impact on the final results.

4.10.3 The Transmission Probability

Finally, we can consider the overall transmission probability, q_t, and how this is estimated when we use the wrong model. We display this in Figure 4.17. We can see that, for both examples, the value of p and γ change to produce transmission probabilities close to the truth, even when the wrong kernel is selected.



(a) True Exponential Kernel

(b) True Gaussian Kernel

Figure 4.17: The transmission probability generated using the mean values outputted using SMC methods (taken at the last day of analysis), under the two different transmission models. Shown in black is the true transmission kernel used to simulate the outbreak. Also shown under each plot is the distribution of the distances between each pair of individuals within the population.

Overall we can answer our original question and say that our analysis, specifically the estimation of the number of occults and the overall transmission probability, is not highly dependent on the choice of spatial kernel, as long as it has similar properties (e.g. decaying as individuals move further apart). In this example we have not considered significantly different distance kernels. This is because currently we are concerned with determining if the power we raise distance to affects the estimation of the other transmission parameters. In the future it would be interesting to compare significantly different distance kernels and see the effect this has, especially within the context of model selection problems.

4.11 The Infectious Period Distribution

So far we have assumed that the infectious periods come from a Poisson distribution with known underlying parameter. The assumption of a known parameter is an appropriate one as often the length of an infectious period may be well studied from observing cases and utilising expert knowledge. This is in contrast to who infects whom which is, firstly, often unobservable and secondly, will depend significantly on the population the outbreak is occurring in. However, it would be useful if we could allow the infectious period parameter to be determined, along with the other parameters. With this in mind we consider our final question, *"Can the algorithm accurately infer the infectious period parameters?"* Recall that we assume that each individual's infectious period comes from a Poisson(a) + 1 distribution, therefore we are now interested in seeing if we can determine the infectious period parameter, a.

4.11.1 Updating a

We will denote the parameters of interest to now be $\boldsymbol{\theta} = (p, \gamma, \kappa, a)$. Considering the MCMC algorithm, we will not update the infectious period parameter, a, with the transmission parameters, p, γ and κ. This is as from the construction of the posterior distribution we can instead use a different proposal step for a.

We begin by noting that as we have used a Poisson(a) + 1 distribution for the infectious periods

$$g_H(x; a) = \frac{a^{x-1}e^{-a}}{(x-1)!}, \tag{4.11.1}$$

where recall g_H is the probability mass function for the infectious period distribution. Additionally if we consider the likelihood function (see Section 3.2.4), then the (marginal) posterior distribution of a is

$$\pi(a\,|\,\boldsymbol{\theta}_{\text{-}a},\,\boldsymbol{y}_{\tau:t},\,\boldsymbol{x}_{0:t}) \;\propto\; \left\{\prod_{j\notin\mathcal{S}_t} g_H(h_t^j;a)\right\}\pi(a) \tag{4.11.2}$$

where $\boldsymbol{\theta}_{\text{-}a} = \boldsymbol{\theta}\backslash\{a\}$ and $\pi(a)$ is the prior distribution of a. Next, we can note that

$$\prod_{j\notin\mathcal{S}_t} g_H(h_t^j;a) \;\propto\; a^{\sum\limits_{j\notin\mathcal{S}_t}(h_t^j-1)} e^{-m_t^I a}, \tag{4.11.3}$$

which we can see follows a Gamma $\left(\sum_{j\notin\mathcal{S}_t}(h_t^j-1)+1,\; m_t^I\right)$ distribution. Therefore we can choose to use a conjugate prior (Section 1.2.3), here we choose a Gamma(λ_a, μ_a) prior, such that the posterior distribution for a now has the form

$$\begin{aligned}\pi(a\,|\,\boldsymbol{\theta}_{\text{-}a},\,\boldsymbol{y}_{\tau:t},\,\boldsymbol{x}_{0:t}) \;&\propto\; \left(a^{\sum\limits_{j\notin\mathcal{S}_t}(h_t^j-1)} e^{-m_t^I a}\right)\times\left(a^{\lambda_a-1}e^{-\mu_a}\right)\\ &=\; a^{\sum\limits_{j\notin\mathcal{S}_t}(h_t^j-1)+\lambda_a-1} e^{-(m_t^I+\mu_a)a}.\end{aligned} \tag{4.11.4}$$

Therefore, by using a conjugate prior, we find

$$\pi(a\,|\,\boldsymbol{\theta}_{\text{-}a},\boldsymbol{y}_{\tau:t},\boldsymbol{x}_{0:t}) \;\sim\; \text{Gamma}\left(\sum_{j\notin\mathcal{S}_t}(h_t^j-1)+\lambda_a,\; m_t^I+\mu_a\right). \tag{4.11.5}$$

As a result of the marginal distribution for a taking a simple form we instead choose to use the Gibbs sampler discussed in Section 1.4.3 to update it.

4.11.2 Simulation Example

To test the accuracy of the SMC algorithm with a unknown, we will use a different simulated outbreak than those discussed so far. This is as, from our experience, a is increasingly difficult to determine for larger values. This could be a result of the adjustment and weighting steps within the SMC, which are highly dependent on the parameter a.

	N_{pop}	p	γ	κ	a	d	Population Distribution
SIR Simulation	500	0.978	20	-	7	-	$\text{Uniform}(0,1) \times \text{Uniform}(0,1)$

Table 4.6: The settings used to generate the SIR epidemic, for which we aim to estimate the value of the infectious period distribution parameter, a.

We simulate an SIR outbreak with the parameters defined in Table 4.6. This outbreak resulted in 94 individuals being infected with the last removal occurring at time $t = 126$; this outbreak has different dynamics to those we considered previously due to its longer infectious period. We used the same prior as previously for p, however now we use a $\text{Gamma}(4, 0.2)$ prior for γ. Additionally, to understand the impact of the prior distribution for a we choose to consider four different choices:

$$\text{Gamma}(0.49, 0.07) \qquad \text{Gamma}(16, 4) \qquad \text{Gamma}(49, 7) \qquad \text{Gamma}(81, 9),$$

which we illustrate in Figure 4.18.

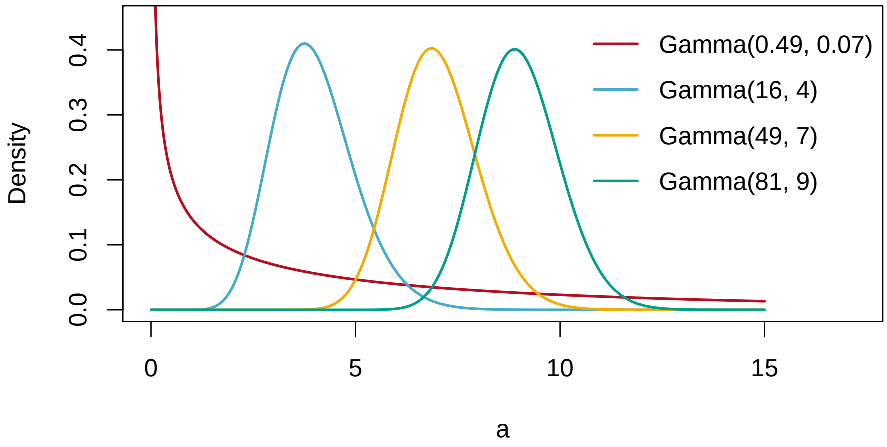


Figure 4.18: The prior distributions we place on the infectious period parameter, a.

The aim is the same as in our previous analysis, to the estimate the parameters underpinning the observed outbreak. We will compare the output from an MCMC, to that generated using an SMC algorithm applied with $n_p = 50$. In Figure 4.19 we show the density plots at two times steps, of the samples generated using each method, for each of the prior distributions for a.

Firstly, we can note that the distribution of parameter a is highly dependent on the prior chosen and this appears to be regardless of the amount of data we use to construct the likelihood. This suggests that the infectious period parameter is difficult to determine given the limited data we have. The other parameters appear fairly robust to different values of a. However, we can see that the number of occult individuals at each time step, u_t, does appear to change slightly, dependent on the prior distribution for a.

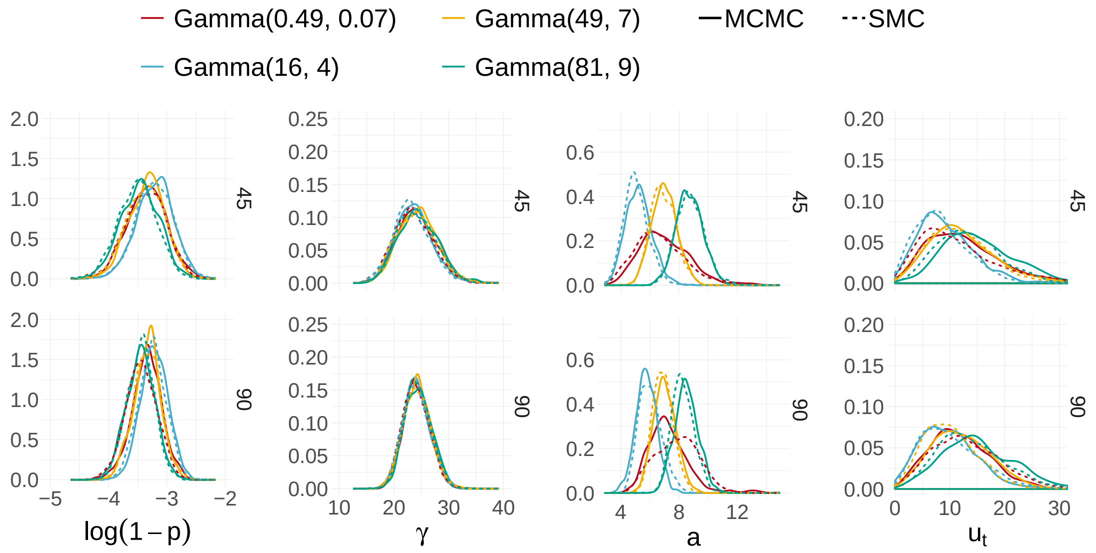


Figure 4.19: The density plots of the particles generated for each parameter using MCMC (solid) and SMC (dashed) methods, compared at two time steps: $t = 45, 90$. Each colour represents a different prior distribution for parameter a. We use a movement step of length $n_p = 50$ within the SMC.

Overall we see that even when the infectious period parameter is free the MCMC and SMC are in good agreement, however, they agree the least when considering the flatter prior for a. The infectious period parameter is highly influenced by the choice of prior placed on it, this suggests that if we are to include a as a parameter the prior must be chosen with care. Similarly if we fix it then we must also do so keeping in mind the effect this may have on the estimation of the other underlying parameters. Altogether we find that although the data can provide some insight about the infectious period parameter, it is not significant enough to overcome the effect of the informative priors that we have considered.

4.12 Extension: Non-Uniform Adjustment

In this chapter we have witnessed the SMC successfully generate samples from the target distribution. Additionally, we have answered some of the outstanding questions we had from developing the algorithm in Chapter 3. In the final two sections of this chapter we consider two extensions to the 'vanilla' SMC algorithm and test how they perform.

Throughout we have been using the uniform weighting scheme when adjusting the particles. In this section we consider the non-uniform weighting scheme introduced in Section 3.5. For clarity we will refer to the 'uniform' weighting as U-SMC (also known as just SMC) and the 'non-uniform' weighting as NU-SMC. Our aim is to compare the performance of the U-SMC to the NU-SMC.

We choose to use the same simulated SIR outbreak as described previously in Section 4.3. Recall that this was an epidemic with two parameters of interest: p, the base rate of infection and γ, the spatial parameter. Additionally we are again interested in the number of occult individuals at each time step, u_t.

We can apply the NU-SMC algorithm in the same way as we did the original (uniform) algorithm (U-SMC). We choose to consider a mixing of $n_p = 25$ and we will compare the output of the U-SMC and the NU-SMC to the MCMC generated every 5 time steps. The results are displayed in Figure 4.20.

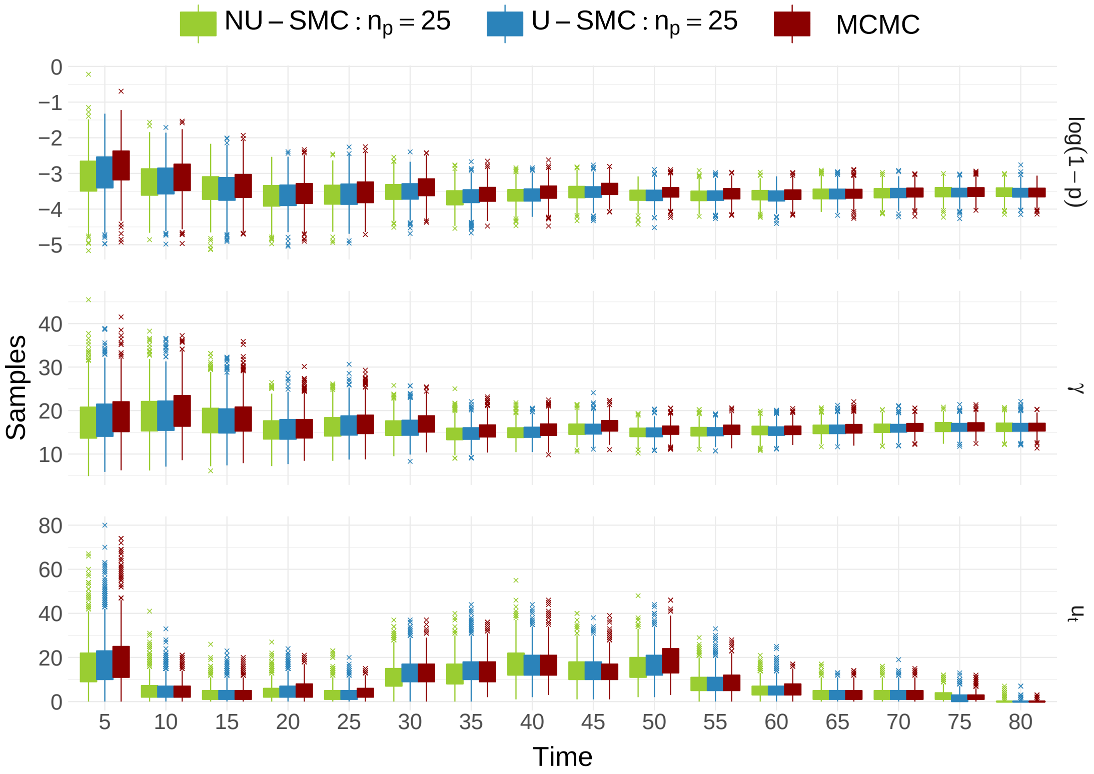


Figure 4.20: A comparison of the particles generated using the SMC algorithm with both the original, uniform, weighting (U-SMC) and the extension with non-uniform weights (NU-SMC). We additionally show the results from the analogous MCMC algorithm. We compare the samples generated at every 5 times steps.

We can see that the two SMC algorithms are in agreement with the MCMC. This is true for both parameters and the value of u_t. We should note that previously in this example we did not have any trouble in matching the MCMC. As such, we are only testing that the alternative weighting scheme produces samples from the evolving target distributions—which we see it does.

4.12.0.1 Unique Particles

Another measure we can consider is the number of unique particles resampled during each iteration of the SMC algorithm. We display the results in Figure 4.21. We can see that the alternative weighting regime has increased the number of unique particles sampled, at almost every time-step considered. This is likely due to the fact that we are proposing a more sensible adjustment, and therefore we will not lose as many 'good' particles as a result of the 'bad-luck' of a poor adjustment. This may prove important for applying the SMC to outbreaks which produce many new events each day, as we found previously that the severity of the degeneracy was highly linked to the intensity of the outbreak (see, Section 4.8).

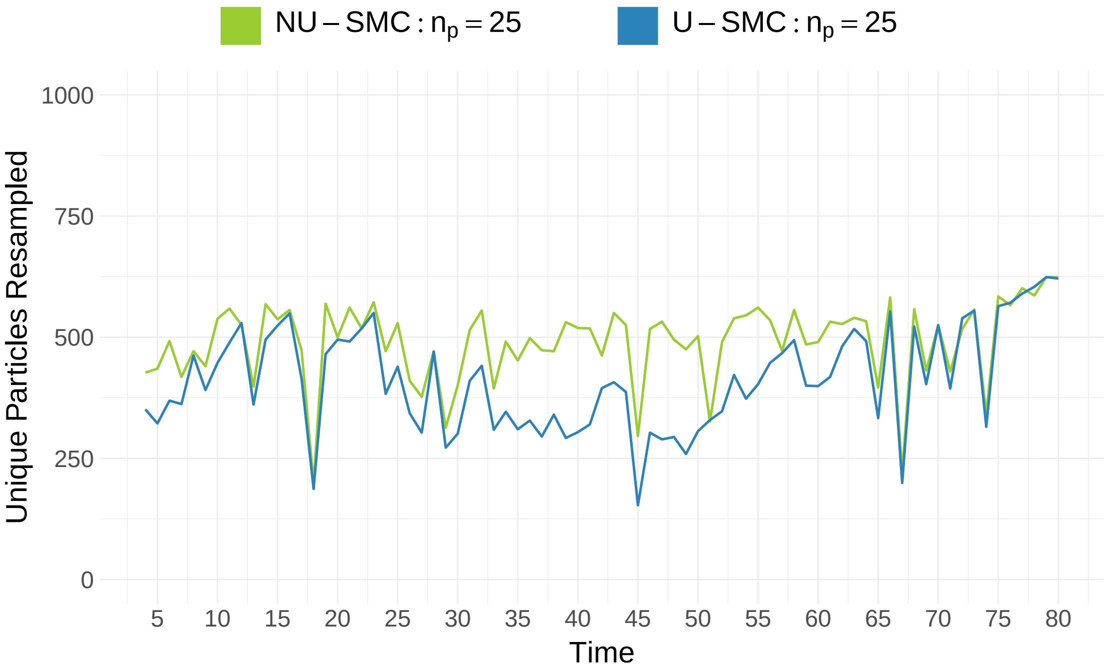


Figure 4.21: A comparison of the number of unique particles resampled at each iteration of the SMC algorithm, when the two different weighting schemes are used.

In Figure 4.21 we note that the improvement is most noticeable during the middle of the outbreak. This is when we are observing the most new observations each day and thus having to make more significant adjustments. Overall this new adjustment has improved the particle degeneracy, without affecting the parameter estimates.

Overall this suggests that the alternative scheme is also producing weights which ensure we are sampling from the evolving target distributions.

4.13 Extension: Duplication Step

The methods we have developed have been successfully applied to the simulated data set. In this section we briefly consider another extension to the SMC algorithm, which may also help to reduce the particle degeneracy. We again assume that we are working with the SMC defined in Section 3.4, which uses the 'uniform' adjustment scheme.

As we illustrated in Section 3.4.4, when working with large data sets we will find that in the particle adjustment step there will be many possibilities as to how the particle is changed. This means that the same particle, by random chance, could be adjusted in several different ways, which may then result in very different weights. This randomness may result in particles being lost, that under different adjustments may have been likely to be resampled. For this reason we propose an addition to the SMC algorithm, a *duplication step*; before we adjust our particles we duplicate each one n_d times resulting in a set of $n \times n_d$ particles. These particles are each then adjusted using the methods of Section 3.4.4. We then calculate the weight of each of these particles and resample n of them from the set of size $n \times n_d$. This addition is illustrated in Figure 4.22.

We hope that this will reduce the loss of 'good' particles due to chance, whilst not affecting those particles that are poor candidates: they will still have a low weight regardless of their adjustment. The aim of the duplication step is not to directly improve the accuracy of the estimates, but rather aid in the reduction of particle degeneracy possibly induced by the adjustment step. In the next section we consider adding this step to the SMC algorithm and seeing how it performs when applied to a simulated outbreak.

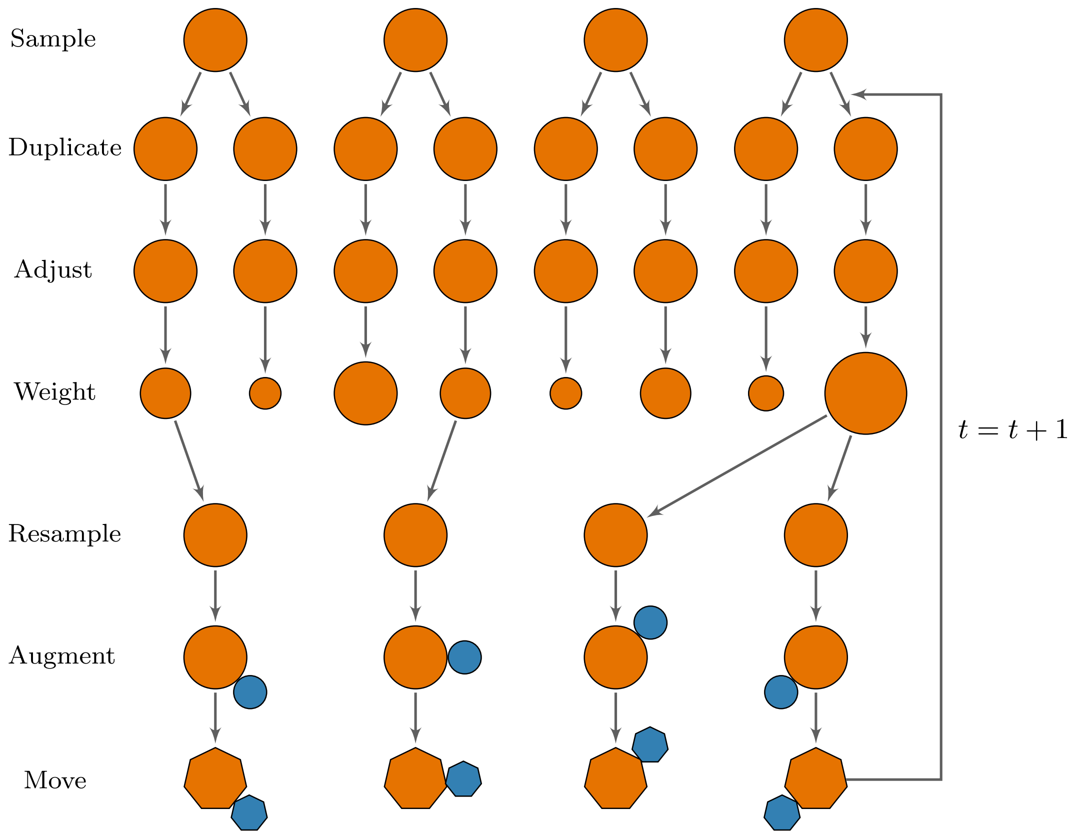


Figure 4.22: Illustration of the SMC algorithm for outbreak data, with the addition of the duplication step.

4.13.1 Simulation Example

We consider the SIR simulation example discussed previously in Section 4.4. We analyse this outbreak as previously, with $n_p = 25$ iterations in the movement step. However, we now consider $n_d = 1, 5, 10, 20$, where $n_d = 1$ matches the previous analysis. This example did not suffer from severe particle degeneracy hence this extension is not required for this outbreak: however, any improvement to the number of unique particles will only serve to strengthen the SMC algorithm.

The two questions we are interested in answering are:

- Does duplicating particles reduce the particle degeneracy?
- Does duplication decrease the accuracy of the parameter estimates?

To test the first question we consider the number of unique particles resampled in each iteration of the SMC algorithm. Now we define a unique particle as a unique set of the parameters, thus selecting two different duplications of the same original particle would only count as a single unique particle. In Figure 4.23 we display the number of unique particles, as we see increasing n_d leads to an increase in the number of unique particles resampled. However, increasing n_d further has little effect and $n_d = 5$ produces a similar number of unique particles to $n_d = 20$.

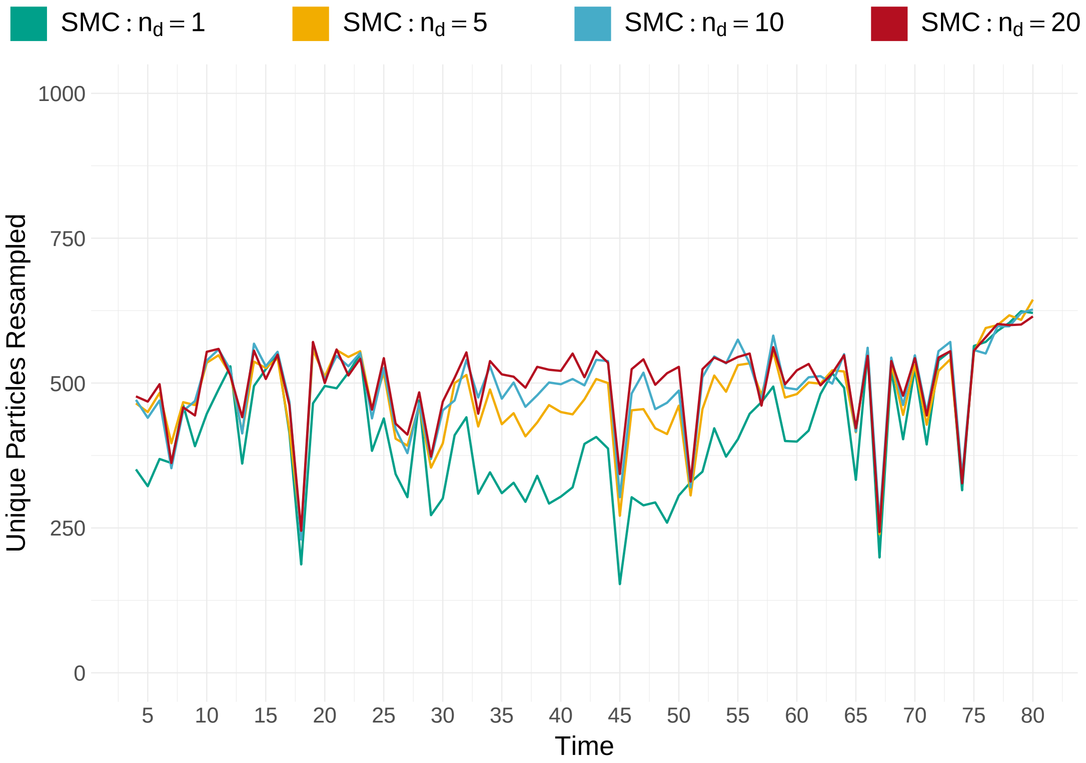


Figure 4.23: The number of unique particles resampled at each step of an SMC algorithm with movement $n_p = 25$ and different values for n_d, the number of duplications. This example is the SIR simulated data considered in Section 4.4.

The second question is of greater importance, does duplication reduce the quality of the output of the SMC algorithm? To test this we consider the distribution of the particles generated, for each value of n_d. This is illustrated in Figure 4.24 where we see that the distribution of the samples generated for each value of n_d appear to be the same. Thus we see that we have successfully improved the particle degeneracy, without suffering a loss in accuracy.

Although we are introducing an additional component the weighting step is significantly quicker to perform than the movement step. As such this does not increase the computation time significantly. We illustrate this in Figure 4.25 where we show that even with full parallelization the time it takes to adjust and weight the additional particles is negligible when compared to the movement step.

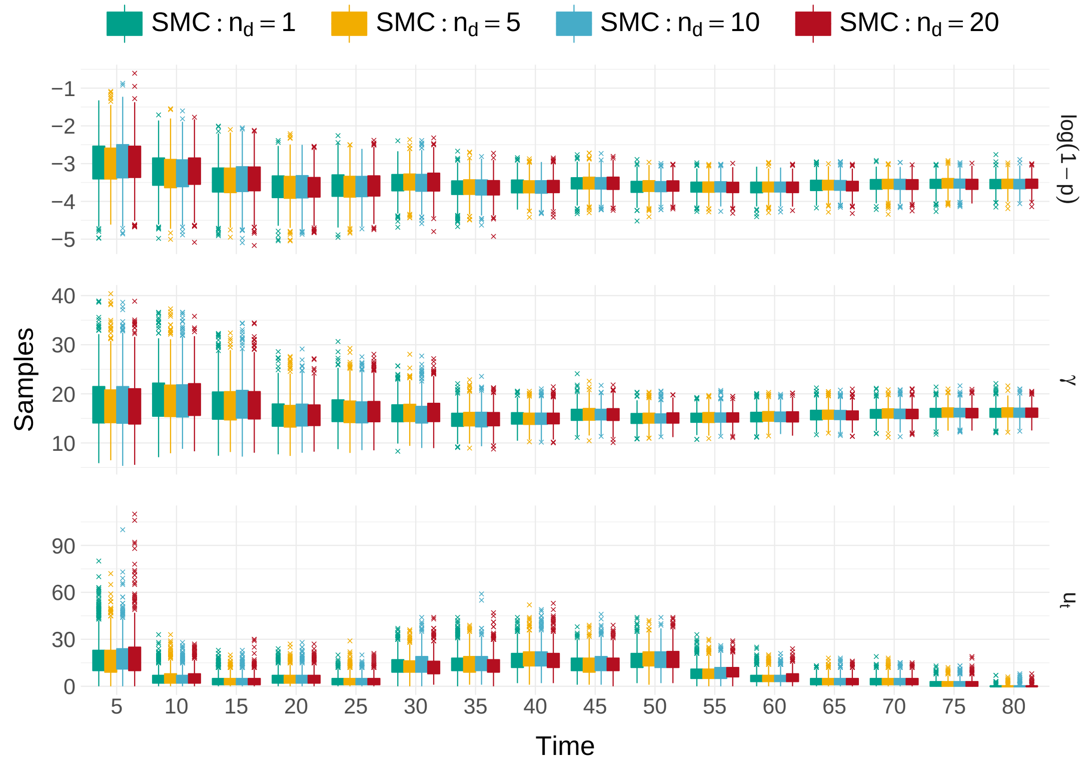


Figure 4.24: A comparison of the output produced using the SMC algorithm ($n_p = 25$), shown every 5 time steps. We have chosen different values for the duplication parameter, n_d, where $n_d = 1$ matches the previous analysis performed on this data set.

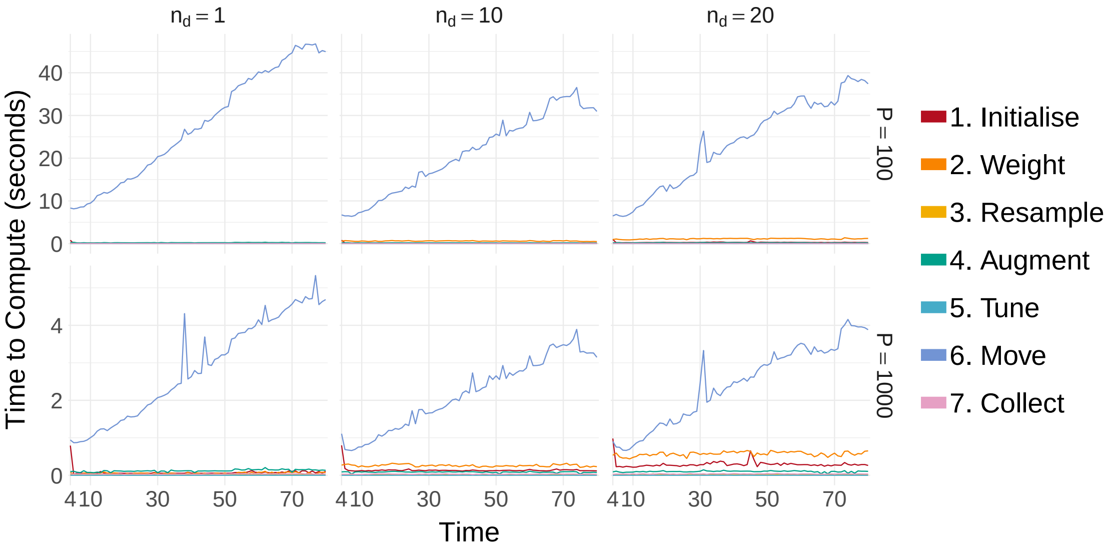


Figure 4.25: A comparison of the time taken to complete each stage of the SMC algorithm, as applied to an SIR outbreak, with $n_p = 25$. We have considered varying levels of duplication, $n_d = 1, 10, 20$, and different values for the number of parallel jobs: $P = 100, 1000$. Here the 'weight' steps includes both the adjustment and weighting of the particle (see Section 4.5 for further details).

4.13.2 Conclusions

Overall this addition to the algorithm is likely to prove useful, especially in examples with a large amount of particle degeneracy. The main benefit of this extension is in its simplicity, it can aid in reducing the issues typically associated with SMC algorithms, without a significant computational burden. We should note that throughout we shall use $n_d = 1$, unless stated otherwise.

4.14 Discussion

In this chapter we have applied the SMC method developed in Chapter 3 to multiple simulated data sets. We have seen that the SMC algorithm matches the accuracy of the MCMC algorithm, even when we use a relatively small movement step. This is reassuring as it means the approximation of the weight is reasonable and therefore the SMC produces samples from (approximately) the target distribution. We have also illustrated that the SMC algorithm continues to be accurate when applied over many iterations, successfully incorporating the new data throughout.

We have also considered extensions to the epidemic model, such as the application when we have infectious periods whose underlying parameters are unknown, as well as additions to the algorithm itself in the form of a duplication step and an alternative weighting. This illustrates the flexibility of the SMC algorithm we have constructed and we expect there to be many more extensions we could incorporate to further the capabilities of this method.

Within this chapter we have only discussed a handful of simple, simulated epidemics, however, this is not when the SMC algorithm is most appropriate. The SMC algorithm we have developed will be of greatest use when we have epidemics for which analysis usually takes longer than 24 hours i.e. longer than the time it takes to obtain the new data. This is often the case when analysing epidemics using MCMC methods which, especially with large amounts of missing data, can take a long time to converge. As such, in the next chapter we aim to apply the methods developed to a real data set, for which online analysis would prove highly beneficial. In this scenario the properties of the SMC algorithm would make it superior to the corresponding MCMC algorithm.

Overall we have shown that SMC methods prove to be a viable alternative to MCMC methods, that can be used in real-time analysis of infectious disease data, repeatedly incorporating new data when it is received. We expect with further consideration and study these methods may grow to be the algorithm of choice for conducting real-time statistical analysis of an epidemic.

Chapter 5

UK Foot-and-Mouth Disease Outbreak (2001)

The sequential Monte Carlo methods we have developed produced impressive results when applied to simulated data sets in Chapter 4. However, as this is data that we have simulated, the underlying mechanism is simple and mathematically well described, this is in contrast to a real data set. For this reason in this section we aim to apply the methods developed in Section 3.4 to a real-world infectious disease outbreak, for which data was collected during its progression. This will present more of a challenge than the simulated data sets: as such this will be a true test of the SMC methods capabilities.

The data set we will be investigating contains the notification and removal times of the farms involved in the 2001 UK Foot-and-Mouth Disease (FMD) outbreak. This data set is of continual interest due in part to the severity of the outbreak, as well as the richness of the data. The outbreak spread very quickly suggesting it is an ideal candidate for the application of the SMC algorithm which can make fast, on-line inference.

5.1 Aims

Within this chapter we aim to achieve the following:

1. Apply the SMC methods to a real data set.

Applying the SMC algorithm developed to a real data set will truly test if it can produce results comparable to those obtained via alternative methods.

2. Demonstrate applying the SMC algorithm in real time.

Ideally the algorithm developed will be applied as the epidemic evolves and new cases are observed. As such, testing it on data that was collected during the progression of the FMD outbreak will be key in checking that the algorithm constructed is appropriate for real-time analysis.

3. Estimate key parameters underpinning the Foot-and-Mouth disease outbreak.

We wish to gain a greater understanding and insight into which factors where important in determining the spread of this disease. Additionally, we will be interested in comparing our inference to that previously performed on this data set.

5.2 Background

Prior to the construction of the FMD epidemic model, we will consider the key characteristics of this outbreak. This will aid in our decision of which features are important to include within the model.

5.2.1 Foot-and-Mouth Disease

Foot-and-Mouth disease is a virus which affects cloven-hoofed animals such as sheep, cattle, pigs, goats and deer. It is rarely fatal to the animal, nevertheless, it will often cause permanent weight loss, as well as a reduction in the quality of their milk. The virus itself spreads extremely quickly and once a farm is infected all trading with it will cease, causing a substantial loss in income. The virus can be transmitted in multiple ways, for example: animal-to-animal contact, contaminated vehicles, items of clothing, contaminated water supplies and animal feed. It is this ease of transmission which allows FMD to rapidly spread, once introduced into a population.

One of the most noticeable symptoms of the virus are vesicles. These are similar in form to blisters but they will often pop quickly, making them difficult to diagnose. The virus has an incubation period of around 2–14 days, although this can vary between species (OIE (2013)).

5.2.2 Timeline of the 2001 UK Foot-and-Mouth Disease Outbreak

We shall be focusing on a specific outbreak of FMD within the UK in 2001. We begin by describing the key moments of the 2001 UK outbreak, as contained within the NAO report (UK National Audit Office (2002)). A timeline of the key events is shown in Figure 5.1.

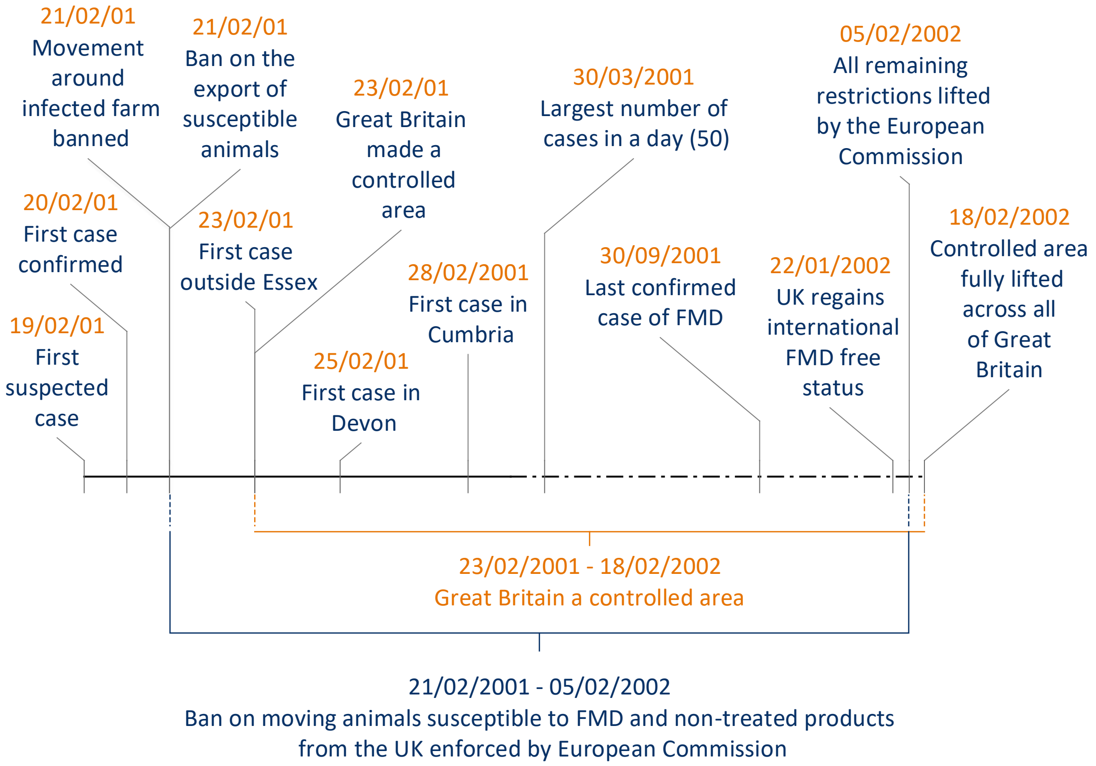

21/02/01
Movement around infected farm banned
21/02/01
Ban on the export of susceptible animals
23/02/01
Great Britain made a controlled area
30/03/2001
Largest number of cases in a day (50)
05/02/2002
All remaining restrictions lifted by the European Commission
20/02/01
First case confirmed
23/02/01
First case outside Essex
28/02/2001
First case in Cumbria
30/09/2001
Last confirmed case of FMD
22/01/2002
UK regains international FMD free status
18/02/2002
Controlled area fully lifted across all of Great Britain
19/02/01
First suspected case
25/02/01
First case in Devon
23/02/2001 - 18/02/2002
Great Britain a controlled area
21/02/2001 - 05/02/2002
Ban on moving animals susceptible to FMD and non-treated products from the UK enforced by European Commission

Figure 5.1: Summary of the key events during the 2001 UK Foot-and-Mouth epidemic. The dates are taken from the UK National Audit Office (2002). Note that, as given by UK National Audit Office (2002), the definition of controlled area is "*The area affected by general control on movement of susceptible animals*".

The first suspected case of FMD during the 2001 outbreak occurred at an abattoir in Essex on 19th February 2001. This was officially confirmed the following day, however, by this time there were already an estimated 57 farms infected and it is likely the initial farm had been infected for weeks. The disease was able to spread so successfully as it had a very short incubation period and animals could infect others before clinical signs of the disease could be seen. Additionally, the disease was primarily spread by sheep as the sheep industry has multiple nationwide animal movement networks, coupled with the fact that symptoms are often difficult to observe in sheep. In Figure 5.2 we can see that this outbreak was confined to specific pockets of the UK, this is as the control

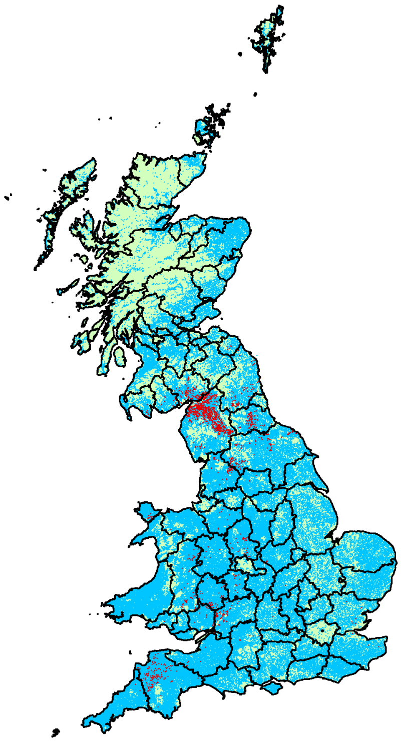


Figure 5.2: The farms within Britain that avoided culling (blue) and the farms which were culled during the 2001 FMD outbreak (red). The large patch of red shows the severity of the outbreak in Cumbria. This is plotted using ArcGIS.

measures put in place prevented the disease from spreading to high risk dairy and pig farming regions (UK National Audit Office (2002)).

In total 2026 premises in Britain were declared infectious, with the worst hit area being Cumbria which had a total of 893 confirmed infectious farms; we can see this severity in Figure 5.2. Also severely affected were the regions of Dumfries and Galloway (176 cases) and Devon (173 cases). The highest number of cases in a single day was 50, with the largest number of cases in a single week being 299 (see, Figure 5.3). Overall the epidemic lasted a total of 32 weeks with the last confirmed case on 30th September 2001. However, it was not until 22nd January 2002 that the UK officially re-obtained its FMD-free status and some restrictions on exports were still in-place until 5th February 2002.

In total the outbreak cost the public sector over £3 billion and the private sector over £5 billion. This demonstrates why this epidemic has remained of such interest to researchers. It was a fast moving outbreak, which caused significant cost to the public and private sector. Additionally the control measures put in place were found to be

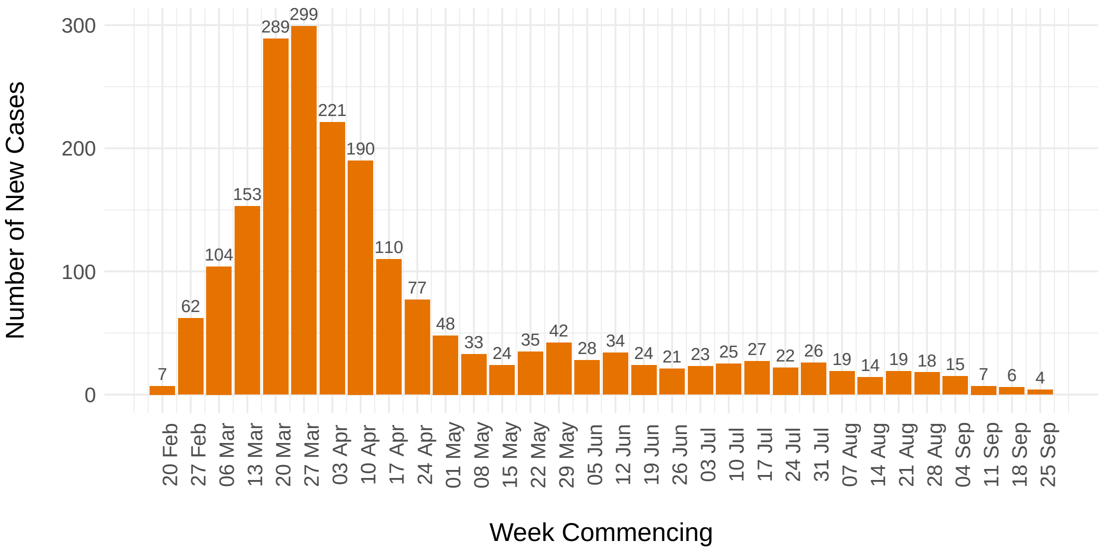


Figure 5.3: The number of new confirmed cases per week, taken from UK National Audit Office (2002).

inadequate (see, UK National Audit Office (2002, Section 2.13)) and as such improving our understanding of how this outbreak spread is of great importance.

5.3 Previous Work

5.3.1 Summary of the Key Findings

There has been a substantial amount of research into FMD, with considerable efforts placed in trying to determine its transmission mechanism and pathogenesis. Alexandersen et al. (2003) performed a general review of FMD, within which they found that mass culling is required to stem any outbreak, as animals can carry the disease without showing symptoms. Also of note they found that within infected sheep there are two levels of infectiousness: an initial period of 7-8 days during which the sheep is highly infectious followed by 1-3 days where it is less so.

Considering the 2001 UK outbreak, Keeling et al. (2001) found that cattle are more infectious and more susceptible to the disease than sheep, however, this is often balanced by the fact that sheep are greater in number. They also found that culling is of much greater effect in limiting the spread than vaccination and therefore stricter culling schemes are the key component in successfully managing an outbreak. This was similarly reported in Ferguson et al. (2001a) who simulated future incidents, finding that ring culling and ring vaccination regimes are required to bring the infections rapidly under control, with culling again being found to be more effective than vaccination.

Again considering the transmission of FMD within the UK, Ferguson et al. (2001b) found that smaller farms are significantly less infectious and less susceptible than larger farms. In agreement with other results they also found cattle farms to be the most susceptible. Interestingly, they additionally concluded that fragmented farms resulted in an increase in transmission, as vehicles often move between the separated farms. Diggle (2006) also concluded that cattle are more infectious and susceptible to infection than sheep, they also found that the relationship between farm size and infectivity and susceptibility is predicted as sub-linear.

5.3.2 Previous Model Assumptions

Before we can apply the SMC algorithm we need to construct the epidemic model. Therefore in this section we will look closer at the model assumptions made in previous work conducted on this data set. We will relate back to this section when defining our own epidemic model in Section 5.5. We provide only a brief overview as motivation for the construction of our model; for a thorough discussion of the models and methods used to analyse this epidemic we refer the reader to Keeling (2005) and Kypraios (2007, Chapter 3).

5.3.2.1 Keeling et al. (2001) (Cambridge-Edinburgh Model)

We begin by focusing on the work performed by Keeling et al. (2001), which much subsequent analysis has been compared against (for example, Diggle (2006), Kypraios (2007) and Jewell et al. (2009)). Their model treated farms as individuals due to the rapid nature of this outbreak—an assumption which the majority of subsequent work also makes. They used a spatial, individual-level, SEIR model (see, Section 2.4.2) assuming a fixed incubation period of 5 days followed by an infectious period of 4 days until the farm was reported.

Their model allowed the infectiousness and susceptibility of a farm to vary based on the species present and the size of the farm, specifically the number of cattle and sheep on a farm. Additionally, this model treated the relationship between farm size and infectiousness and susceptibility as linear. They used a distance kernel which was a function of the Euclidean distance and produced by DEFRA by tracing the path of the infection.

Keeling et al. (2001) used two methods to conduct analysis, firstly they constructed the discrete-time likelihood and used maximum-likelihood methods. However, they found this to be inadequate due to biases in the data. Therefore, for their second form of analysis they used their initial estimates obtained via ML and 'refined' them using least-squares methods, measuring the difference between the observed and expected number of daily cases. Using these methods they found that large farms are more susceptible to the disease and that cattle are both more infectious and more susceptible than sheep. They concluded that, as well as a biological factor, this may be a result of the contact networks the two different species were involved in. Overall they concluded that cattle probably contributed more per capita to the spread of FMD, however, this is balanced by the far greater number of sheep.

5.3.2.2 Deardon et al. (2010)

The work performed by Keeling et al. (2001) was continued in Deardon et al. (2010) where several extensions were proposed, including: a non-linear relationship between farm size and a farms infectivity and susceptibility, the ability to have spontaneous infection and a more general spatial kernel. The methods used by Deardon et al. (2010) required knowledge of the status of all farms at all times, these were estimated if unknown. Additionally, like Keeling et al. (2001), the model assumed a constant period of 9 days between infection and reporting, split into a latent and infectious period.

Deardon et al. (2010) conducted analysis using a Bayesian framework and explored the posterior distribution using MCMC methods. The authors found that the kernel used by Keeling et al. (2001) overestimates short distance risks, whilst underestimating the risk of long-distance infections. They additionally matched the results of previous work, showing that cattle are more infectious than sheep as well as being more susceptible. However, they found that the assumption of the infectivity and susceptibility of a farm scaling linearly, as made in Keeling et al. (2001), to be dubious.

5.3.2.3 Jewell et al. (2009)

The next work we shall focus in on is the continuous-time SIR model used by Jewell et al. (2009) and applied to the subset of the data belonging to Cumbria. Following the work of Keeling et al. (2001) this model also only focuses on the transmission mechanism as a function of the number of cattle and sheep, additionally the authors allow for a non-linear relationship between a farm's size and its infectiousness and susceptibility. They also define the environmental transmission as a function of the Euclidean distance between farms k and ℓ, denoted by the kernel $K(k, \ell, \gamma)$ where γ is to be inferred. For this model a Cauchy type kernel is used.

To model this outbreak Jewell et al. (2009) assumed that an infectious farm, k, makes infectious contact with a susceptible farm, ℓ, at points of a time-inhomogeneous Poisson process with rate

$$\beta_{k,\ell} = \underbrace{(\beta_1(c_k)^\chi + (s_k)^\chi)}_{\substack{\text{Infectiousness}\\ \text{of } k}} \underbrace{(\beta_2(c_\ell)^\chi + (s_\ell)^\chi)}_{\substack{\text{Susceptibility}\\ \text{of } \ell}} \underbrace{K(k, \ell, \gamma)}_{\substack{\text{Environmental}\\ \text{transmission}}}, \tag{5.3.1}$$

where c_x and s_x denote the number of cattle and sheep respectively, on farm x. The parameters β_1 and β_2 denote the relative infectiousness and susceptibility of cattle to sheep and χ represents how the size of a farm affects the rate of transmission. The spatial effect now encapsulates the effect of environmental factors on transmission, such as the disease being spread by rodents, birds, people etc.

In contrast to the work of Keeling et al. (2001) and Deardon et al. (2010), the authors additionally allow the farms' infectious periods to be random, assuming that the time between infection and removal follows a Gamma(a, b) distribution where $a = 4$ and b is some unknown parameter to be determined.

With the model constructed, Jewell et al. (2009) conducted inference in a Bayesian setting using MCMC methods. They additionally consider a partially non-centered MCMC algorithm (see Papaspiliopoulos (2003, Chapter 7)), with 25% of the infection times non-centered in each iteration.

The authors find that over 4km there is little infectious pressure exerted. Additionally the results obtained showed that cattle are more susceptible and more infectious than sheep, matching previous results. In agreement with other work, the authors also found there to be a sub-linear relationship (χ between 0.25 and 0.4) between the rate of infection and the number of animals on each farm. Finally, the authors find that the infectious period is on average between 6.5 and 9 days, this result is slightly different to the assumptions made in other work (for example Keeling et al. (2001) and Deardon et al.

(2010)), however, this model did not incorporate an exposure period so this is to be expected.

5.3.2.4 Xiang and Neal (2014)

The final model we consider is that used by Xiang and Neal (2014), which shares many similarities to the epidemic model used by Jewell et al. (2009). They applied an SIR model to the Cumbria data set, making the assumption of Gamma(a, b) infectious periods. Similarly, they used MCMC methods to learn about the form of the parameters, also applying non-centering methods.

The authors results matched well with those produced in Jewell et al. (2009). Interestingly, they also found that the mean length of the infectious period is highly dependent on the value of a. If $a = 1$ the average infectious period is 5.4, for $a = 4$ it is 7.9 and for $a = 20$ it is 9.5. This shows two things: firstly the results do not directly match those of Keeling et al. (2001), which used a fixed infectious period, and secondly the data itself can have difficulty in determining the true infectious period. However, overall the authors found that the average total infectious pressure exerted by a farm throughout the outbreak is not highly dependent on the infectious period, as many of the transmission parameters are robust to changes in its form. This matches with our observations in Section 4.11 where we found that although the form of the infectious period was difficult for the methods to determine, the transmission parameters were largely unaffected by its value.

5.4 The Data Set

With our discussion of the previous work on this outbreak completed, in this section we describe the data set we will be using and consider its interesting features.

5.4.1 Cleaning the Data

We shall be performing analysis on the 2001 UK Foot-and-Mouth outbreak, restricted to those farms located in the county of Cumbria. This region was one of the most severely affected during the course of the outbreak (see, Figure 5.2) and, as such, is where the focus of our analysis shall lie. Specific analysis of the outbreak within Cumbria has been discussed in Diggle (2006), Jewell et al. (2009) and Xiang and Neal (2014).

To simplify the model we will begin by cleaning the data, as stated we shall only be keeping the data on those farms that fall within the traditional county boundaries for Cumbria. We will also omit those farms which are removed at the time of the first observed notification, these farms were removed 22 days before any other observed notification or removal time, thus we assume that they were not involved in the major outbreak within Cumbria. During this outbreak there were farms culled preemptively, without their infection status being confirmed. To match the work performed by Jewell et al. (2009) and Kypraios (2007) we do not use this information, focusing only on data relating to those farms culled as infectious premises during the outbreak. Finally we only consider farms that contained some cattle or sheep, as these species were the primary carriers of FMD (see, Keeling et al. (2001)).

5.4.2 Summary Statistics

The Cumbria data set comprises of 7876 farms, of which 891 were culled as infectious premises by the end of the epidemic. The progression of FMD within Cumbria was rapid and showed a strong spatial component.

In Figure 5.4 we show the total number of reported cases over time. We can see that this outbreak had an initial period of rapid spread, followed by a longer period of lesser intensity. We will focus on using the methods we have developed near the start of the outbreak, as this is when fast analysis is of greatest use. However, this may be difficult for the SMC algorithm as we will have many new observations to incorporate each day.

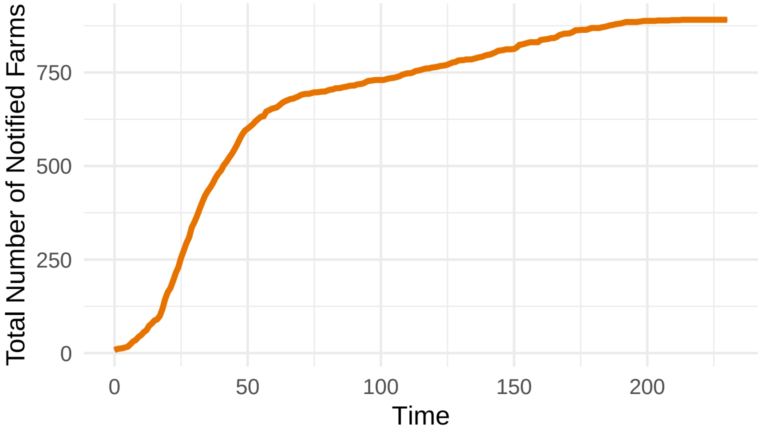


Figure 5.4: The total number of farms confirmed as infectious premises, from the time of the first observed case.

In Figure 5.5 we can again see this rapid spread. Additionally we can observe the highly localised nature of this outbreak, with the majority of the infected farms residing in the north of Cumbria. This suggests that the outbreak did have a large spatial element, as we would expect. We can also see how this outbreak might be problematic for analysis using the SMC algorithm, with a large number of new cases occurring in a very short space of time.

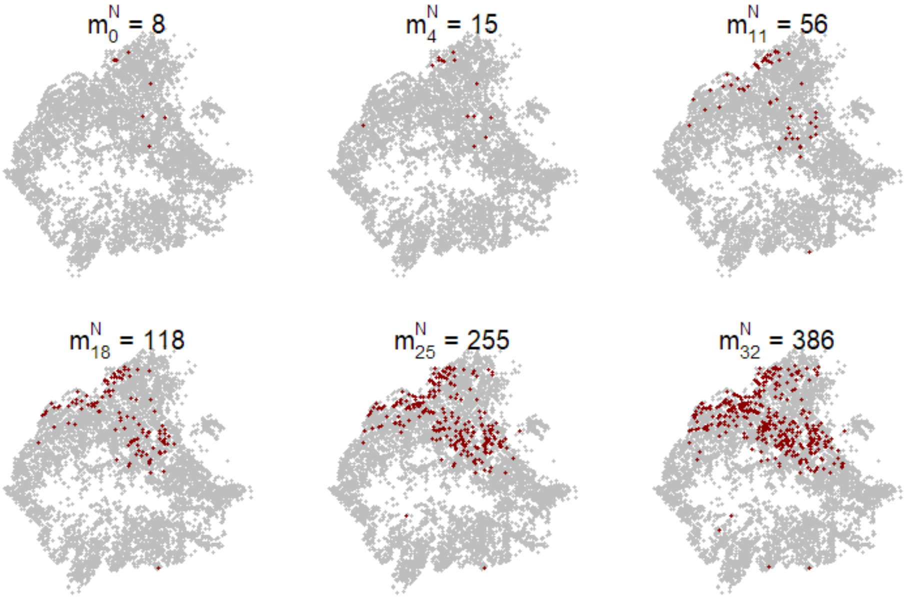


Figure 5.5: The number of notified cases (m_t^N) recorded every 7 days, from time $t = 4$ up to time $t = 32$, these will be the times at which we compare the SMC and MCMC methods later in Section 5.6. For comparison, we have also included the state of the outbreak at the time of the first notification, $t = 0$. In grey we show every farm and in red we display those confirmed as infectious at time t.

Another potential issue highlighted by Figure 5.5 is that we may have some impact from boundary effects. For example there are infections occurring to the south of Cumbria, which may be more likely to have occurred from outside the county borders. Throughout we will not investigate the impact of this, however it is worth bearing in mind. One future solution would be to consider the work of Diggle (2006) and incorporate a chance of spontaneous infection. This would be suitable for accounting for infections coming into Cumbria from outside the boundary we have considered.

We may also be interested in the composition of the farms, as summarised in Figure 5.6. We can note that within this population the majority of farms held both sheep and cattle. However, we can see in Figure 5.7 that overall there is a far greater number of sheep than cattle within the county of Cumbria.

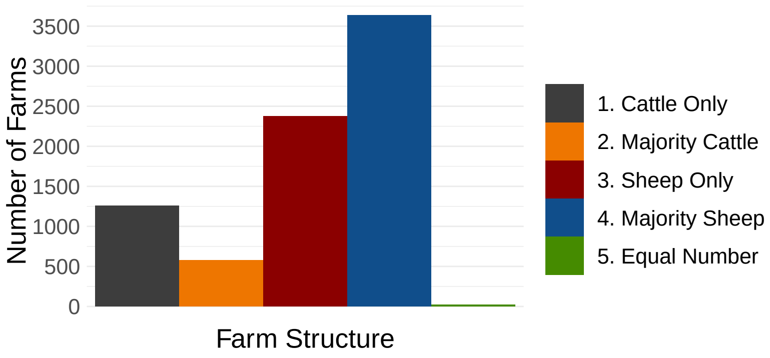


Figure 5.6: The proportion of farms within Cumbria with only cattle, majority cattle and some sheep, only sheep, majority sheep and some cattle or equal number of cattle and sheep.

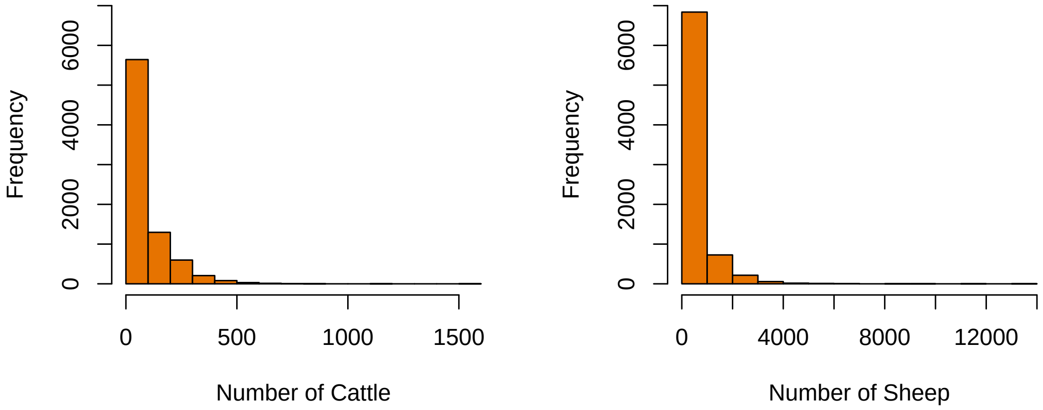


Figure 5.7: Histogram of the total number of cattle and sheep on each of the farms within Cumbria.

To analyse this outbreak, we have access to two pieces of data: the notification and removal times. Respectively these times represent when a farm is first known to carry FMD (notified) and when the animals have been culled (removed). We display the distribution of the notification period in Figure 5.8. We see that it will be useful to incorporate this period as the majority of farms have a delay between identification and the removal of all of the infected animals.

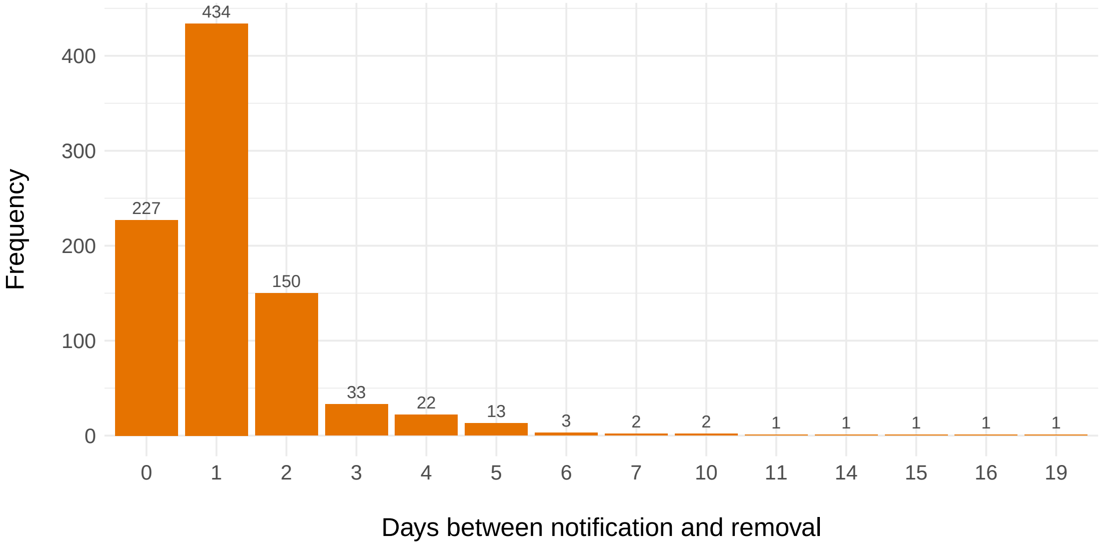


Figure 5.8: The time between notification and removal, for the farms infected within Cumbria during the 2001 UK FMD outbreak.

5.5 Constructing the Model

Now that we have explored the data that we are interested in analysing, we can begin construction of our model. We will use the form of the data set, as well as the work previously performed, to inform on our choices.

5.5.1 Disease Progression

Following the discussion in the previous section, we choose to model this outbreak using an SINR model. This is in contrast to much of the previous work on this data set, which focuses either on SIR models, such as Jewell et al. (2009) and Xiang and Neal (2014), or on SEIR models, such as Keeling et al. (2001) and Deardon et al. (2010). We choose this model as from Figure 5.8 we can see that the period between notification and removal is not trivial, as such we aim to utilise both pieces of data we have access to. It will be interesting to see if we can pick out the behaviour occurring during the notification period as, for most farms, it is very short.

5.5.2 Spatial Component

Given the data it is evident that we will require a spatial component within any model we construct. This will take the form of a distance kernel, $K(d(i,j))$, where $d(i,j)$ is a distance metric which defines how we quantify the distance between two farms.

5.5.2.1 Distance Metric

When previously designing our epidemic model we always utilised the Euclidean distance between individuals when defining the spatial component. This need not be the case, therefore we briefly justify why using the Euclidean distance between farms is an appropriate metric for the 2001 UK FMD outbreak. This has been previously discussed in Jewell et al. (2009).

Savill et al. (2006) showed that for each sub-region they considered, the Euclidean distance and the quickest route were equally good predictors of risk for the 2001 UK FMD outbreak. One of the regions they considered was Cumbria. The authors noted that even when accounting for the unique features of this region, for example the M6 motorway running through it, a Euclidean based transmission kernel was sufficient to model transmission between farms on either side of it—despite this feature separating them. This is possibly as there are many routes through which the disease can be transmitted (e.g. private farmers tracks, footpaths etc.) thus the Euclidean distance is a better representation of the spatial effect than other simplifications: for example the shortest route via roads. As such we will continue to use the Euclidean metric to represent the spatial separation between farms.

5.5.2.2 Spatial Kernel

There has been significant interest in choosing an appropriate distance kernel for the FMD epidemic. For example Keeling et al. (2001) used a highly tailored distance kernel produced by DEFRA, which used tracing data and expert knowledge to deduce the most likely source of infection. As a result of its construction, which was based on (subjective) estimation by veterinarians on the most likely source of infection, this kernel was found to overestimate the effects of short-distance transmissions, whilst underestimating those occurring over a longer distance (Deardon et al. (2010)). This was improved by Deardon et al. (2010) who produced a kernel with a change point such that short distances were explained by a constant, whilst long distances were described by a geometric kernel. Both of these kernels were used to represent the spatial transmission across the whole of the UK epidemic. These may not be as appropriate for our context, where we are only interested in the outbreak occurring within Cumbria.

The Cumbria data was discussed by Diggle (2006) who used a distance kernel which allowed direct transmission over short distances, whilst also allowing spontaneous transmission to account for the longer distances. Also using the Cumbria data, a Cauchy kernel was used by Jewell et al. (2009) due to its heavy tail.

We will choose to keep the exponential distance kernel used previously in the simulated outbreaks. This will allow us to see how the general model performs, as ideally we wish to create a method which can be quickly and easily applied to data from different outbreaks. Additionally, as we are working with a reduced data set, the heavy tailed distribution is not as necessary, unlike the work which considers the outbreak across the whole of the UK. This is illustrated in Figure 5.5 where we can see that the outbreak appears to be dominated by short-range infections. In the future it would be interesting to consider the impact of using a heavy-tailed distance kernel, as in Jewell et al. (2009). Although relative to the whole of the UK Cumbria is small, the region could be still considered large and therefore a heavy-tailed kernel may result in different conclusions.

Finally, we note that in the previous chapter the wrong choice of distance kernel did not severely impact the estimation of the parameters. Thus, we might expect that this would also hold for the FMD dataset: as such, we continue to use the exponential distance kernel.

5.5.3 The Infectious Period

The infectious period is often assumed to be fixed in previous work (see, Keeling et al. (2001) and Deardon et al. (2010)), however, this is highly restrictive. Therefore we allow the infectious periods to be variable. As we are working in discrete time a Poisson$(a) + 1$ distribution seems appropriate, the $+1$ is required as we assume that a farm cannot be notified on the day it is infected. We additionally choose a to be fixed at $a = 5$, due to the difficulty of inferring the infection parameter from the data alone (see Section 4.11). Additionally, previous work has suggested that the inference of the transmission parameters is relatively unaffected by the infectious period, see Xiang and Neal (2014) and the simulated example in Section 4.11. We choose an average of 6 days ($a = 5$) for the infectious period as, together with the notification stage, this agrees with the length of the infectious period found by Kypraios (2007), Jewell et al. (2009) and Xiang and Neal (2014).

In many cases the distribution of the infectious period is relatively well-known and can be observed by other means (e.g. previous outbreaks or laboratory analysis). However, what is often unknown is what is driving the spread of the disease. This will be case dependent and as such our focus will remain on the transmission parameters.

5.5.4 Additional Model Parameters

For the other model settings we choose to follow previous work and incorporate the number of cattle and sheep that reside on each farm into the infectiousness and susceptibility of a farm. Additionally—due to the work by Diggle (2006), Jewell et al. (2009) and Deardon et al. (2010)—we choose to incorporate the possibility of a non-linear relationship between the probability of infection and the size of the farms. In Diggle (2006) and Jewell et al. (2009) this was controlled by a single parameter, however, we will choose to consider two parameters to see if the scaling for susceptibility and infectiousness are distinct (see, for example, Deardon et al. (2010)).

5.5.5 The Posterior Distribution

With the model assumptions made we can begin construction of the posterior distribution. We briefly describe the set-up for an SINR model, as previously discussed in Section 3.6. For an SINR model, we define the form of the posterior distribution as

$$\pi(\boldsymbol{\theta},\ \boldsymbol{y}_{\tau:t} \mid \boldsymbol{x}_{0:t}) \ \propto L(\boldsymbol{\theta};\ \boldsymbol{y}_{\tau:t},\ \boldsymbol{x}_{0:t})\pi(\boldsymbol{\theta})$$

where $\boldsymbol{x}_{0:t} = \{\boldsymbol{n}^N_{0:t},\ \boldsymbol{n}^R_{0:t},\ \boldsymbol{r}^R_{0:t}\}$ and $\boldsymbol{y}_{\tau:t} = \left\{\boldsymbol{i}_{\tau:t},\ \boldsymbol{n}^I_{0:t}\right\} = \left\{\boldsymbol{i}^N_{\tau:t},\ \boldsymbol{i}^R_{\tau:t},\ \boldsymbol{i}^I_{\tau:t},\ \boldsymbol{n}^I_{0:t}\right\}$. Recall that now the infectious period refers to the time between notification and infection:

$$h^j_t = \begin{cases} n^k_t - i^k_t & \text{if} \quad i^k \leq t \\ 0 & \text{if} \quad i^k > t \end{cases}. \tag{5.5.1}$$

Additionally the time between notification and removal is referred to as the notification period.

We can then define the likelihood function to be

$$L(\boldsymbol{\theta};\ \boldsymbol{y}_{\tau:t},\ \boldsymbol{x}_{0:t}) = \prod_{s=\tau}^{t-1} \left\{ \prod_{\ell \in \mathcal{S}_{s+1}} P_s(\ell\,;\,\boldsymbol{\theta}) \prod_{\ell \in \mathcal{S}_s \backslash \mathcal{S}_{s+1}} \big(1 - P_s(\ell\,;\,\boldsymbol{\theta})\big) \right\} \prod_{j \notin \mathcal{S}_t} g_H\left(h_t^j\,;\,\boldsymbol{\theta}\right) \quad (5.5.2)$$

where $P_t(\ell\,;\,\boldsymbol{\theta})$ is the probability that farm ℓ avoids infection at time t and g_H is the infectious period density function. For the SINR model we define the probability of avoiding infection to be

$$P_t(\ell\,;\,\boldsymbol{\theta}) \ = \ \prod_{k \in \mathcal{I}_t} (1 - q_t(\ell, k)) \prod_{l \in \mathcal{N}_t} (1 - \kappa q_t(\ell, l)), \quad (5.5.3)$$

where we can interpret $q_t(\ell, k)$ as the probability farm k infects farm ℓ with no restrictions. We refer to this as the *transmission probability.* We additionally denote by κ the reduction in infectiousness when a farm becomes notified. This is the basic structure of the SINR model we use throughout.

5.5.5.1 The Transmission Probability

The next step in constructing the posterior distribution is to define the form of the transmission probability, this will allow us to define $P_s(\ell\,;\,\boldsymbol{\theta})$ in the likelihood equation, (5.5.3).

We construct this by considering the parameters we have decided to incorporate, additionally we choose to follow closely to the model described in Jewell et al. (2009), as displayed in Section 5.3.2.3. We denote by c_x and s_x the number of cattle and sheep respectively on farm x, then for a farm $\ell \in \mathcal{S}_t$ we define the unrestricted probability that it is infected by farm $k \in \mathcal{I}_t / \mathcal{N}_t$, at time t, as:

$$q_t(\ell,\, k) = 1 - \exp\Big\{ - \beta_0 \underbrace{(\beta_1 c_k + s_k)^{\chi_1}}_{\substack{\text{Infectiousness} \\ \text{of } k}} \underbrace{(\beta_2 c_\ell + s_\ell)^{\chi_2}}_{\substack{\text{Susceptibility} \\ \text{of } \ell}} \underbrace{\exp\{-\gamma d(\ell,\, k)\}}_{\substack{\text{Environmental} \\ \text{transmission}}} \Big\}, \quad (5.5.4)$$

where $d(x, y)$ is the Euclidean distance (measured in meters) between farms x and y. The parameters incorporated are:

- γ, the parameter controlling the importance of two farms' proximity in the infection probability. We use an exponential spatial component to represent the decay in the transmission probability when considering farms that are separated by increasing distance.

- β_0, the base-rate of infection.

- β_1 and β_2, the relative contribution a cow makes compared to a sheep in the infectiousness and susceptibility, respectively, of a farm.

- χ_1 and χ_2, parameters which determine how the infectiousness and susceptibility, respectively, of a farm changes with size.

With this choice of transmission probability we can see that for an infectious farm, $k \in \mathcal{I}_t$, and a susceptible farm, $\ell \in \mathcal{S}_t$,

$$\begin{aligned} &\text{as } d(\ell,k) \longrightarrow \infty && q_t(\ell,k) \longrightarrow 0, \\ &\text{as } d(\ell,k) \longrightarrow 0 && q_t(\ell,k) \longrightarrow 1 - \exp\left\{-\beta_0 \left(\beta_1 c_k + s_k\right)^{\chi_1} \left(\beta_2 c_\ell + s_\ell\right)^{\chi_2}\right\}. \end{aligned}$$

Therefore the further away two farms are the less likely transmission is. However, even if the farms are very close to each other, transmission is not guaranteed and will still be dependent on the structure of the farms.

We have already chosen the form of the infectious period distribution in Section 5.5.3, where we decided to fix the infectious period parameter. Therefore the parameters we are interested in inferring are

$$\boldsymbol{\theta} = (\kappa,\, \gamma,\, \beta_0,\, \beta_1,\, \beta_2,\, \chi_1,\, \chi_2). \tag{5.5.5}$$

Additionally, we again assume independent priors, such that

$$\pi(\boldsymbol{\theta}) \;=\; \pi(\kappa)\,\pi(\gamma)\,\pi(\beta_0)\,\pi(\beta_1)\,\pi(\beta_2)\,\pi(\chi_1)\,\pi(\chi_2). \tag{5.5.6}$$

For κ we assign a Beta prior distribution whilst for all other parameters, which are defined on the positive half-line, we assign a Gamma prior distribution.

5.6 Analysis

With the model defined we can begin our analysis of the Cumbria FMD data set. The form of analysis will follow the same outline as the analysis performed on the simulated outbreaks discussed in Chapter 4.

5.6.1 Algorithm Conditions

We will begin our analysis at time $T = 4$, four days after the first notification has been observed. By this time there has only been $m_4^N = 15$ notifications and $m_4^R = 6$ removals observed. Thus, as in the simulated examples of Chapter 4, we are initialising the SMC at close to the start of the outbreak.

To begin we need to generate the $n = 1000$ initial particles; we achieve this using the same MCMC algorithm discussed in Chapter 3. This algorithm will additionally be used for comparison and for the movement step of the SMC. One modification is that, to help improve the acceptance rate, we choose to scale the covariance matrix in the parameter proposal distribution by $1/d^2$ rather than $1/d$ (see, Section 1.4.4.4). For the MCMC algorithm which incorporates a burn-in (i.e. not the MCMC within the movement step) we choose to adaptively tune between iterations $b_1 = 2000$ and $b_2 = 10000$ and use a burn-in of 500000.

In Table 5.1 we display the prior distributions we will be using for each parameter. To reflect our lack of knowledge about the values of the parameters, we have used uninformative prior distributions. We have chosen for β_0 a prior with a small mean, this is to reflect that the per-contact transmission probability is relatively low. Similarly the prior for γ also has a low mean, a result of the distance kernel being a function of meters. For β_1 and β_2 we choose a prior with mean 1 to reflect the initial premise that cattle and sheep contribute equally to both a farm's infectiousness and susceptibility. Finally we have chosen the priors for χ_1 and χ_2 to have an expected value of 0.5, as from previous work (see, Diggle (2006)) it is found that the scaling is sub-linear. Additionally in Deardon et al. (2010) the estimated values for the comparable parameters varied between 0.1 and 0.9.

Once the initial particles have been generated we will then run the (uniform) SMC algorithm forward 28 days to time $t = 32$, which coincides with the epidemic ending

Parameter	κ	γ	β_0	β_1
Prior	Uniform$(0,1)$	Gamma$(1,1000)$	Gamma$(1,10000)$	Gamma$(1,1)$

Parameter	β_2	χ_1	χ_2
Prior	Gamma$(1,1)$	Gamma$(1,2)$	Gamma$(1,2)$

Table 5.1: The uninformative priors placed on each parameter used to model the FMD data set.

its period of rapid growth, with $m_{32}^N = 386$ and $m_{32}^R = 361$. We will then assess the results of the SMC algorithm by comparing it to the output of an MCMC algorithm, as in Chapter 4.

We shall compare the results of the SMC algorithm applied with movement steps of length $n_p = 200$ and $n_p = 500$ to the analogous MCMC algorithm. We will refer to the SMC with $n_p = 200$ and $n_p = 500$ as the 'shorter' and 'longer' SMC algorithms, respectively. The movement step is much longer than previously considered in the simulated outbreaks in Chapter 4. This is to account for the fact that the outbreak was severe and had many new cases each day. As we found previously, this is likely to reduce the number of unique particles and in general the SMC algorithm will struggle more. The use of a longer movement step is not an issue as it will still take a fraction of the time of the corresponding MCMC, which can take a long time to converge. To also aid in reducing the particle degeneracy we utilise the duplication step previously described in Section 4.13 and set $n_d = 10$. Finally, as before, the tuning parameters are adaptively chosen in each iteration of the SMC to produce an acceptance rate close to the optimal 25%.

5.6.2 Results

We show in Figure 5.9 a comparison of the density plots produced by the SMC algorithms and the MCMC algorithm, compared at times t = 11, 18, 25, 32 (a week apart). Also included are the initial particles generated at time $t = 4$, which the SMC algorithms are initiated with. We also show in Table 5.2 a summary of the results generated at each time step. When discussing the results it is with implicit reference to Table 5.2 and Figure 5.9. We are only comparing at these time steps to illustrate the results, however, the SMC algorithm has produced output for each time step in between.

Overall we see that the three algorithms are mostly in agreement and, as we would expect, the longer SMC algorithm ($n_p = 500$) is closer to the results produced by the MCMC algorithm than the shorter ($n_p = 200$). In general there is least agreement between the SMC and MCMC when inferring the number of occult individuals, u_t. This is likely as this value is dynamic and highly related to the large number of unknown infection times, corresponding to the occult individuals. This will be further discussed in Section 5.6.2.7.

We show in Figure 5.10 a comparison of the prior and the posterior distributions at time $t = 32$. We can see that, with the exception of β_1, the parameters are insensitive to the choice of prior distribution. We began this outbreak with relatively little data: as a result the posterior distribution at time $t = 32$ is drastically different to that at time $t = 4$. The SMC has therefore successfully handled significant changes in the shape of the target distribution. In the next sections we consider these results in detail.

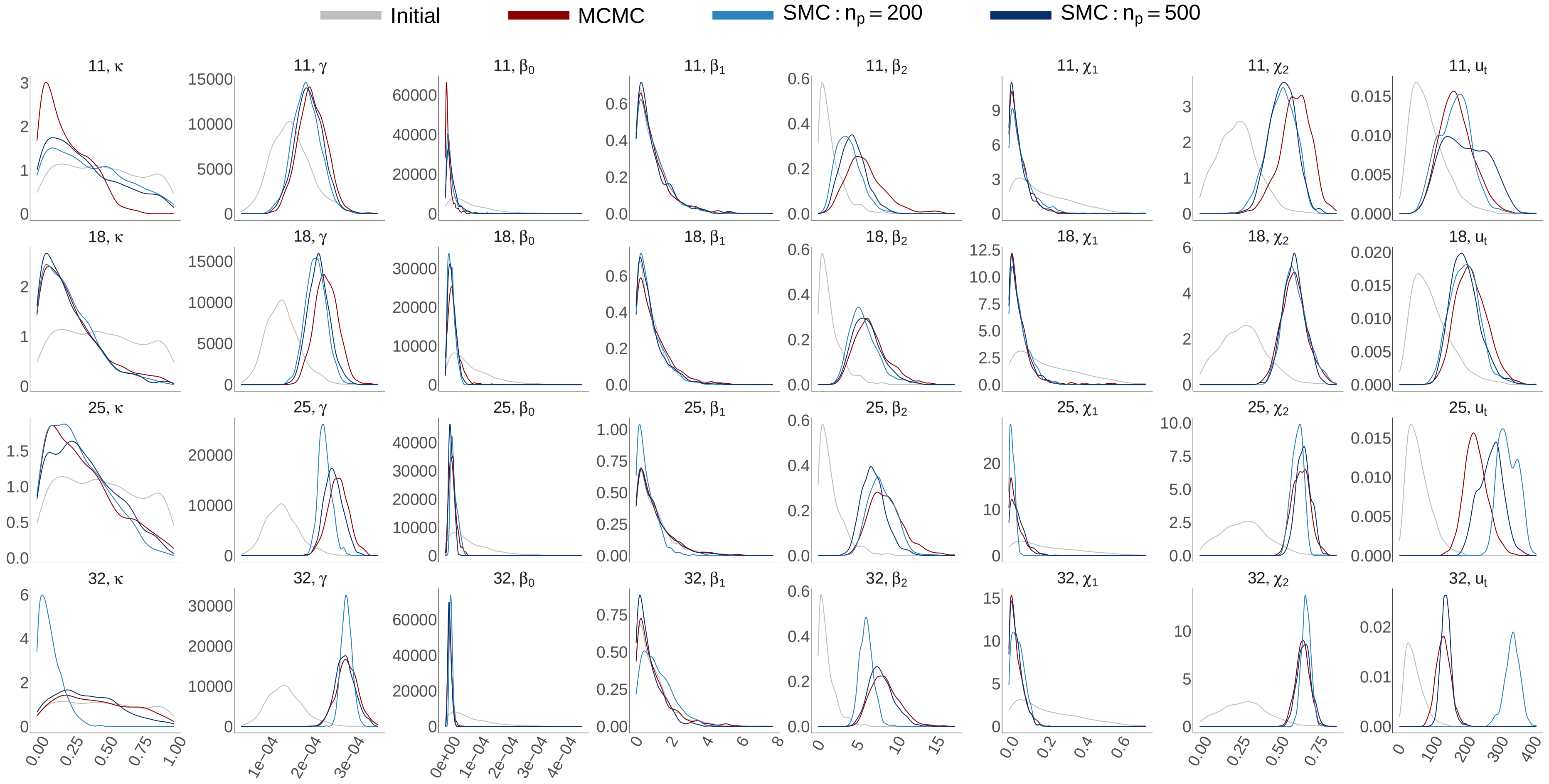


Figure 5.9: The estimated posterior distribution of each parameter, formed using the particles generated using SMC (blue) and MCMC (red) methods. In grey we show the initial particles used within the SMC algorithm.

		$t = 11$		$t = 18$		$t = 25$		$t = 32$	
		Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.
κ	SMC: $n_p = 200$	0.382	0.268	0.242	0.196	0.310	0.212	0.100	0.0789
	SMC: $n_p = 500$	0.344	0.259	0.235	0.199	0.351	0.234	0.368	0.231
	MCMC	0.214	0.164	0.257	0.216	0.337	0.245	0.433	0.262
$\gamma/10^{-4}$	SMC: $n_p = 200$	1.85	0.278	2.34	0.250	2.58	0.163	2.95	0.123
	SMC: $n_p = 500$	1.90	0.283	2.41	0.262	2.80	0.225	2.91	0.229
	MCMC	1.96	0.288	2.60	0.290	2.92	0.255	2.96	0.243
$b_0/10^{-5}$	SMC: $n_p = 200$	2.28	2.30	2.20	1.19	3.20	1.16	2.12	0.512
	SMC: $n_p = 500$	2.48	2.25	2.46	1.35	2.30	0.818	1.71	0.587
	MCMC	1.37	1.68	3.01	2.18	2.72	1.30	1.83	0.763
β_1	SMC: $n_p = 200$	1.02	0.912	0.927	0.813	0.614	0.526	1.21	0.866
	SMC: $n_p = 500$	1.01	1.00	0.971	0.948	0.986	0.907	0.753	0.690
	MCMC	1.03	1.03	1.13	1.09	1.05	1.02	0.991	0.991
β_2	SMC: $n_p = 200$	2.79	1.14	4.26	1.33	5.49	1.13	6.18	0.861
	SMC: $n_p = 500$	3.24	1.16	4.6	1.40	4.79	1.05	8.04	1.58
	MCMC	4.35	1.83	4.97	1.57	5.87	1.47	8.55	1.84
χ_1	SMC: $n_p = 200$	0.0757	0.0732	0.0598	0.0536	0.0200	0.0136	0.0515	0.0334
	SMC: $n_p = 500$	0.0630	0.0773	0.0561	0.0517	0.0491	0.0392	0.0417	0.0334
	MCMC	0.0610	0.0557	0.0595	0.0740	0.0406	0.0393	0.0418	0.0361
χ_2	SMC: $n_p = 200$	0.628	0.107	0.641	0.0821	0.632	0.0370	0.689	0.0293
	SMC: $n_p = 500$	0.637	0.102	0.632	0.0720	0.673	0.0473	0.676	0.0448
	MCMC	0.737	0.117	0.633	0.0849	0.657	0.0563	0.666	0.0441
u_t	SMC: $n_p = 200$	90.8	26.0	95.5	20.5	255	22.6	345	22.7
	SMC: $n_p = 500$	107	35.1	96.0	20.4	214	27.4	144	15.8
	MCMC	93.5	26.5	105	22.1	180	26.3	137	21.8

Table 5.2: The mean and standard deviation (S.D) of the particles generated for each parameter, at time steps $t = 11\,, 18\,, 25\,, 32$. We compare the output from an SMC with $n_p = 200$ and $n_p = 500$, as well as the equivalent MCMC.

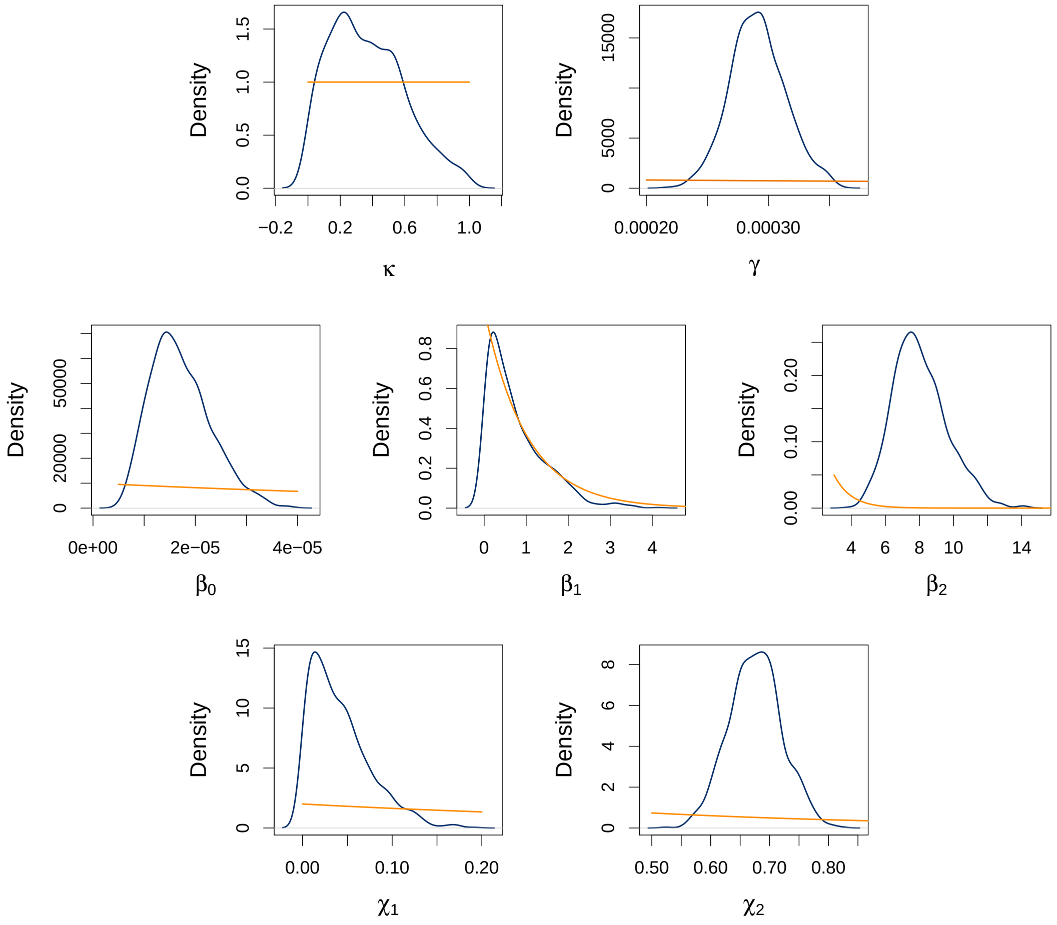
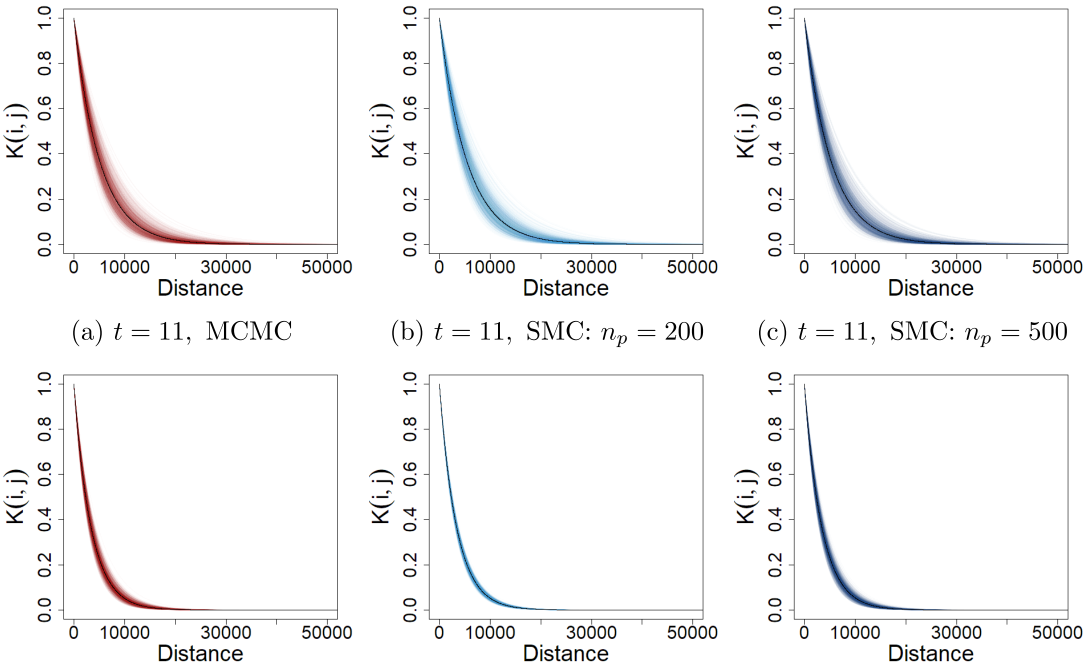


Figure 5.10: The posterior distribution (blue) generated at time $t = 32$ by the SMC algorithm with $n_p = 500$. Also shown is the prior distribution (orange) for each parameter.

5.6.2.1 Spatial Parameter

We begin by considering the spatial parameter, γ. We can see in Figure 5.9 that for this parameter we have reasonable agreement between the algorithms, although the shorter SMC struggles more at the later time steps. In Figure 5.11 we display the distance kernel evaluated at each value of γ generated by the three different methods. Again we can see that the SMC and the MCMC are in close agreement. This is important as quickly understanding the spatial component of a disease outbreak is vital when informing on control policies.



(a) $t = 11,$ MCMC (b) $t = 11,$ SMC: $n_p = 200$ (c) $t = 11,$ SMC: $n_p = 500$

(d) $t = 32,$ MCMC (e) $t = 32,$ SMC: $n_p = 200$ (f) $t = 32,$ SMC: $n_p = 500$

Figure 5.11: The distance kernel, $K(i, j) = \exp(-\gamma d(i, j))$, evaluated at each of the values of γ generated using the three methods, at times $t = 11$ and $t = 32$. We only show a portion of the distance range due to the kernel flattening out for very large distances. In black we show the kernel evaluated using the mean value of γ generated using each method.

From Figure 5.11 we can see that the spatial effect appears significant for farms up to 15km away. This is surprising given the work of Jewell et al. (2009), who found that there was significance only up to 4km away. This is possibly because we have not used a heavy-tailed kernel, thus the fitted kernel has to account for both short and long-distance transmissions. Therefore the ‘significance’ may actually be an artefact of the transmission kernel we have used. We may consider in the future using a thicker-tailed distance kernel, or introducing an additional chance of spontaneous infection (see, Diggle (2006)). Additionally, we can see in Figure 5.11 that the effect of distance becomes more significant with the incorporation of more data. These results suggest that the initial spread is fairly dispersed, with spatial effects becoming more prominent as the outbreak continues. This reflects the increasing restrictions on movement which were enforced, resulting in the transmissions becoming highly localised as the epidemic progressed. We note that this illustrates another strength of the SMC algorithm: it can detect changes in behaviour during the outbreak.

5.6.2.2 The Infectious Farm

We next examine the parameters relating to the structure of a farm, beginning with the contribution to the transmission probability from the infectious farm. Thus we are considering parameter β_1, the relative infectiousness of each cow compared to each sheep, and χ_1, a parameter determining how the contribution from the infectious farm scales with the size of that farm.

For β_1, we see (in Figure 5.10) that this value does not change much from the prior distribution placed on it, suggesting that the data alone cannot distinguish this value. This is not overly concerning as it is likely that, under our model, the difference in the infectiousness of cattle and sheep does not play an important role in the transmission of the disease. Additionally, we find that the average value of β_1 is close to 1 at each time-step considered (see, Figure 5.9). Although not directly comparable, similar overall findings occur in the other work on this data set. For example, although they found cattle to be more infectious, Diggle (2006), Jewell et al. (2009) and Xiang and Neal (2014) estimated that cattle and sheep contribute (almost) equally to the infectiousness of a farm.

If we consider how the infectiousness of a farm scales with the size of the farm, χ_1, the value for this is fairly low (an average < 0.1 for each time step considered), minimising the contribution of the infectious farm in the transmission probability. This is similar to the scaling found in Deardon et al. (2010): although the authors used two parameters so the results are not directly comparable, they found that the infectiousness scaled with sheep by $\chi_1^s \in (0.000, 0.249)$ and by cattle as $\chi_1^c \in (0.147, 0.471)$. We can additionally note that the value of χ_1 provides insight into the results for β_1. As χ_1 is fairly small this will mean there is less contribution to the transmission probability from the infectious farm and thus this may be why we have difficulty in determining the value of β_1 from the data.

Overall for β_1 and χ_1 the SMC and MCMC are in agreement. This is to be expected for β_1 as there has been little change throughout the time steps considered. However, for χ_1 we can see the shape of its distribution underwent significant changes between times $t = 4$ and $t = 11$ which the SMC has captured, even with the shorter movement step.

5.6.2.3 The Susceptible Farm

We next examine the contribution to the transmission probability from the susceptible farm. Therefore we consider parameter β_2, the relative susceptibility of each cow compared to each sheep, and χ_2, a parameter determining how the contribution from the susceptible farm scales with the size of that farm.

In contrast to β_1, we see that β_2 is defined as much greater than 1 (> 3 at each time step considered in Figure 5.9), increasing in value at each time step compared. This suggests that each individual cow contributes more to the susceptibility of a farm than each individual sheep. This matches with the previous work discussed in Section 5.3, for example, Keeling et al. (2001).

Additionally we see that in contrast to χ_1, χ_2 is defined as being around 0.6–0.7. Therefore, although the effect of the farm size on the susceptibility of a farm is sub-linear, it is not negligible. This may relate to why, in contrast to β_1, we can determine the value for β_2, as its contribution to the transmission probability is significant. The value of χ_2 matches with those found by Deardon et al. (2010), who again split this into two parameters with the susceptibility scaling for sheep estimated as $\chi_2^s \in (0.877, 0.941)$ and for cattle as $\chi_2^c \in (0.842, 0.934)$.

As with the infectious farm parameters, overall the MCMC and SMC are in agreement when inferring the values of β_2 and χ_2. This is especially impressive as we see χ_2 in particular has a shift in location, as we obtain new data. This has not been an issue for the SMC and the re-weighting has accounted for this. The shorter movement step struggles at time $t = 32$ to determine β_2, which is likely due to the significant shift in value that this parameter experiences between times $t = 25$ and $t = 32$.

5.6.2.4 Total Transmission Probability

In this next section we consider how the estimated values for the parameters affects the overall transmission probability between farms. We test this by considering the transmission probability:

$$q_t(\ell, k) = 1 - \exp\Big\{ - \beta_0 \underbrace{(\beta_1 c_k + s_k)^{\chi_1}}_{\substack{\text{Infectiousness} \\ \text{of } k}} \underbrace{(\beta_2 c_\ell + s_\ell)^{\chi_2}}_{\substack{\text{Susceptibility} \\ \text{of } \ell}} \underbrace{\exp\{-\gamma d(\ell, k)\}}_{\substack{\text{Environmental} \\ \text{transmission}}} \Big\}. \tag{5.6.1}$$

Following Jewell et al. (2009) we consider four farms as described in Table 5.3. We then consider the transmission probability between farms with this composition using the mean values generated for each parameter, from each method, at time $t = 32$. We plot the results in Figure 5.12.

	Large Farm	Cattle Only	Sheep Only	Small Holding
Sheep	500	0	1000	6
Cattle	50	100	0	2

Table 5.3: The composition of the four 'typical' farm types in the Cumbria data set, as used in Jewell et al. (2009).

From Figure 5.12 it is difficult to see the difference between the transmission probability for the MCMC and the SMC with $n_p = 500$, as they lie on top of each other. This indicates that, although the parameter distributions vary slightly between the SMC and the MCMC, they are estimating similar transmission probabilities. For the shorter chain the transmission probability is not as close to that produced by the MCMC, reflecting the lesser agreement in the individual parameters.

Considering Figure 5.12 in greater detail we can notice some interesting behaviour. Firstly, we can see that the smaller holding is around 10 times less likely to be infected than the larger farm. Thus, as we expected from the parameter values, the contribution from the farms does not scale linearly. Additionally we see that the average cattle only and sheep only farm behave similarly, reiterating that although each individual cattle is more susceptible, this is counteracted by the on average higher number of sheep per farm.

We can also now see even clearer that the transmission probability does not depend on the structure and size of the infectious farm, but is rather fully determined by the

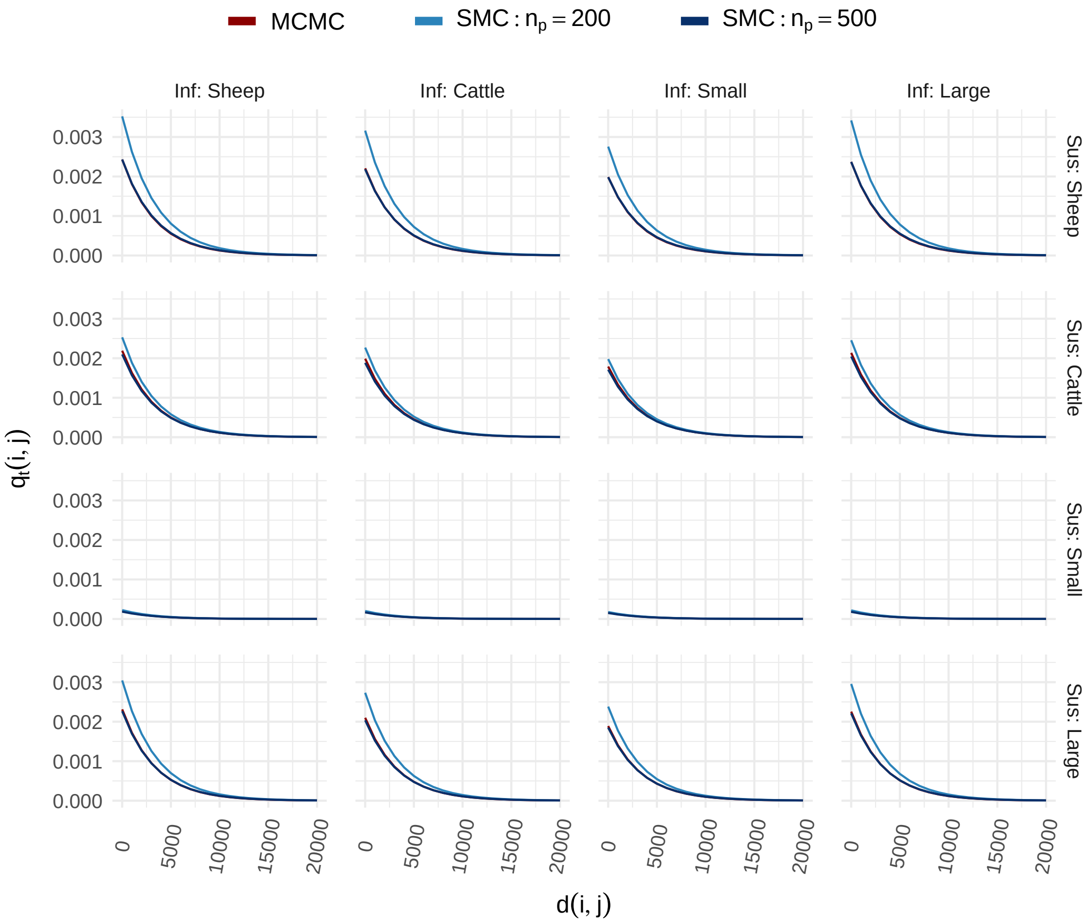


Figure 5.12: The probability of infection, $q_t(i, j)$, against distance, $d(i, j)$, between farms of different sizes (see Table 5.3), where ‘Inf’ represents the infectious farm and ‘Sus’ represents the susceptible farm. $q_t(i, j)$ is evaluated at the average value generated for each parameter, at time $t = 32$, using the three different algorithms.

susceptible farm. This will impact on how we would advise on control policies, with evidence suggesting that it is important to contain all infectious farms, with even small holdings posing a risk, matching the conclusions of Jewell et al. (2009). It also suggests that we should concentrate on reducing movement to those farms most at risk of infection.

Overall the results for the transmission parameters relating to the effect of a farm’s structure on the transmission probability match with work previously performed on this data set i.e. that the infectiousness and susceptibility does not scale linearly and that cattle contribute more per capita to the transmission probability than sheep. We have also additionally found that, in the event of an outbreak, it is the structure of the susceptible farms that plays a key role in how the disease is spread.

5.6.2.5 Notification Parameter

The notification parameter, κ, represents the reduction in the infectiousness of a farm once we become aware of its infectiousness. We can see in Figure 5.9 that we are predicting κ as between 0.2–0.5, suggesting that the control measures placed on a farm once we are aware it is infectious are reasonably successfully in reducing the infectiousness of that farm.

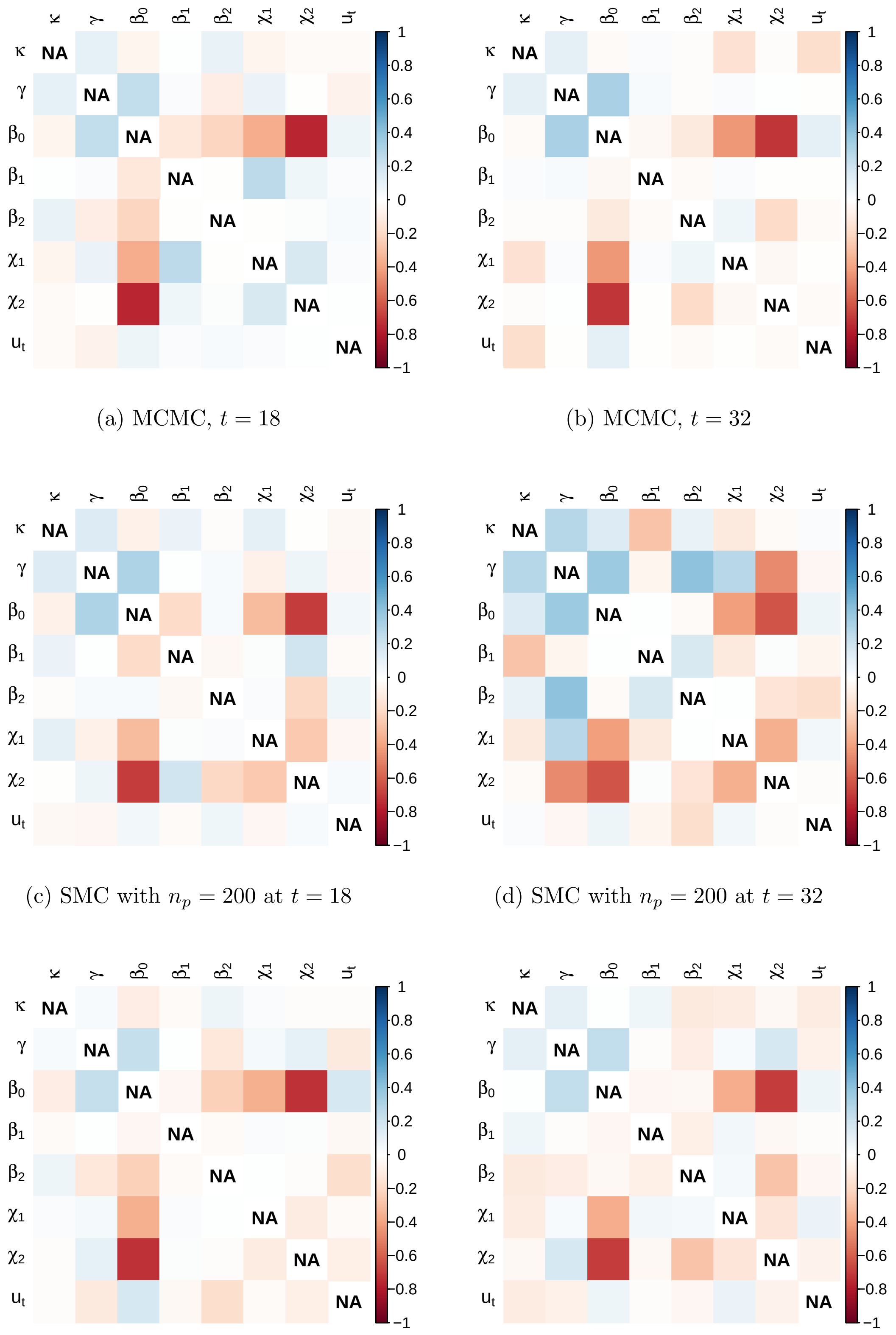
We observed previously in Section 5.4.2 that the response once a farm became notified was fairly rapid. As a result we will have limited data with which to estimate the notification parameter. This is reflected in Figure 5.9 where we can see that there appears to be a large amount of uncertainty around the value of κ. This additionally may be reflecting that κ cannot be accurately described by a point estimate, as the effectiveness of control measures is likely to vary widely by farm.

Also of interest, we find that the the mean value of κ changes between time steps $t = 18$ (in the longer SMC $\hat{\kappa} = 0.235$) and $t = 32$ ($\hat{\kappa} = 0.368$). This may be a result of the posterior distribution evolving as we acquire more data. Additionally this could suggest that there may have been a change in behaviour during the outbreak, which has resulted in a change in the shape of the posterior distribution. From the post-outbreak report from UK National Audit Office (2002) it is stated that there was a backlog in the disposal of the animals such that "*carcasses were sometimes left rotting on farms for days on end and this discouraged prompt slaughter*". This may be what we are witnessing in the behaviour of the parameter value, as a backlog occurs there are less resources to contain the infectious farms and thus there is less reduction in their infectiousness. This once again illustrates the ability of the SMC algorithm to detect potential changes in behaviour that can then be further investigated and then, if needed, acted upon.

5.6.2.6 Parameter Correlation

Next, we check to see if the values produced by the various methods are a result of any correlation between the parameters. In Figure 5.13 we plot the correlation between the parameters generated at times $t = 18$ and $t = 32$, using each method.

The first thing to note is that there is a strong negative correlation between χ_2 and β_0, -0.7 as generated at $t = 32$ by the SMC with $n_p = 500$. This is understandable as if χ_2 increases then β_0 will need to decrease in order to compensate. We can note



(a) MCMC, $t = 18$

(b) MCMC, $t = 32$

(c) SMC with $n_p = 200$ at $t = 18$

(d) SMC with $n_p = 200$ at $t = 32$

(e) SMC with $n_p = 500$ at $t = 18$

(f) SMC with $n_p = 500$ at $t = 32$

Figure 5.13: An illustration of the correlation between the parameters, as computed using the particles generated by the three different algorithms. We display the results for times $t = 18$ and $t = 32$.

that this correlation is stronger than the correlation between β_0 and χ_1 (-0.37). This is likely due to the smaller estimated value of χ_1. Similarly there is a positive correlation between γ and β_0 (0.25). This is because a strong spatial component (high γ) requires a larger value of β_0 to compensate for the rapid decay in infectivity.

We observe that there are stronger relationships between the parameters in the shorter SMC when compared to the MCMC and the longer SMC. Additionally, this effect appears to become worse as we apply the SMC over an increasing number of iterations. However, by utilising a longer movement step within the SMC we appear to be better matching the MCMC in terms of the relationship between parameters.

5.6.2.7 The Number of Occults

Finally, we are interested in inferring the current number of occult individuals, u_t, at each time step. If we recall Figure 5.9 then we find that for $t = 11$ and $t = 18$ the three methods are (mostly) in agreement. This is good to see as the value of u_t is fairly high (u_t estimated as greater than 90 throughout the time period we consider). However, when we consider the later time steps we see that the SMC and MCMC algorithms are in worse agreement. Both SMC algorithms struggle at time $t = 25$, however, the longer SMC matches well with the MCMC at time $t = 32$. The value of u_t at time $t = 25$ (u_t around 200) is estimated higher than at time $t = 32$ (u_t around 140), suggesting that the SMC algorithm has greatest difficulty when there are a greater number of occult individuals. This is perhaps to be expected given the nature of the algorithm, the unknown infections will be difficult to accurately infer.

Overall the longer SMC ($n_p = 500$) provides a reasonable estimate for the value of u_t, at each time step. Additionally, due to there being little correlation between u_t and the transmission parameters (see, Figure 5.13), the other estimates are not significantly affected by its value.

These results demonstrate a key advantage of the SMC: in real-time analysis the current number of infectious individuals can be accurately predicted. We can then use these predictions as indications of the severity of the outbreak, possibly days before formal confirmation. For outbreaks which are spreading rapidly this information is vital for informing on how to best reduce the current spread of the outbreak. We display in Figure 5.14 the estimated number of occults at each time step, highlighting the ability of

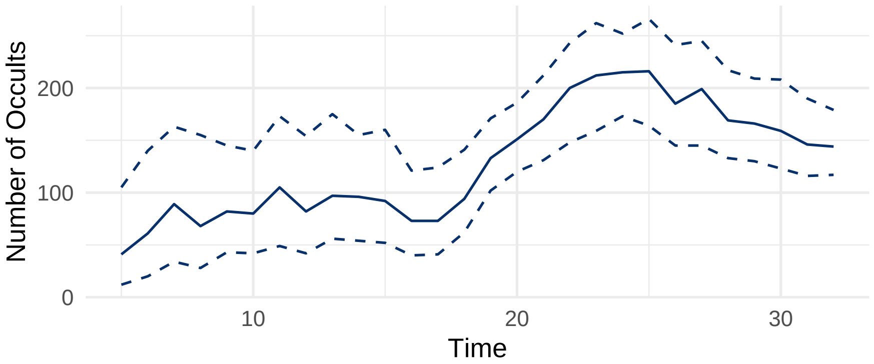


Figure 5.14: The median number of occults (solid) at each time step, generated using an SMC algorithm with $n_p = 500$. Also shown are the upper (97.5%) and lower (2.5%) quantiles represented by the dashed lines.

the SMC to adaptively predict the number of currently infectious individuals, throughout time.

5.6.3 Conclusions

This epidemic was severe, with many new cases observed each day—as displayed in Figure 5.15. Because of this, the SMC algorithm needed to incorporate a large amount of information at each time step. As we might expect, the hardest value to infer was the number of occult individuals: the true value of this will change at each time step and, as such, is the biggest challenge for the methods developed. Overall, however, we have seen that the SMC and MCMC algorithms are in agreement when estimating the key parameters underpinning the FMD outbreak.

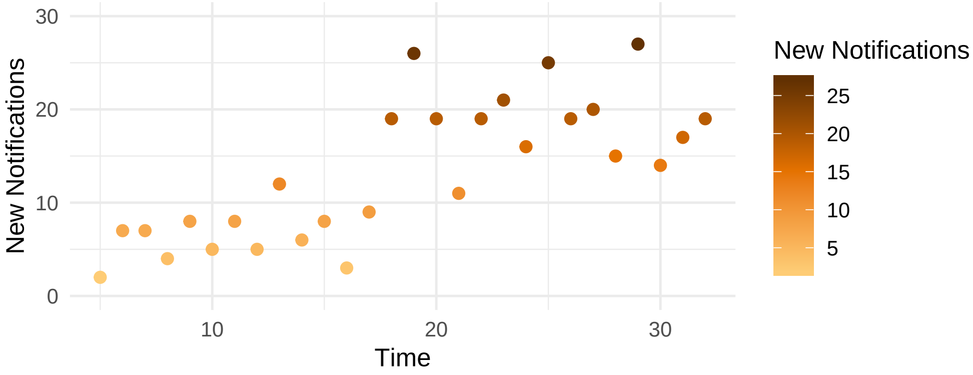


Figure 5.15: The number of new confirmed cases (notifications) observed at each time step, from the start of the SMC at $t = 4$ to the end of the time frame analysed, $t = 32$.

5.7 Non-Uniform Adjustment

We have seen in the previous section that the SMC and MCMC are in reasonable agreement with each other. However, the SMC method did appear to have greatest difficulty in inferring the number of occult individuals, u_t. This may be a result of the adjustment step of the SMC algorithm (discussed in Section 3.4.4), as this directly involves inference about the occult individuals. Therefore in this section we consider the alternative adjustment scheme introduced in Section 3.5, which may result in a greater level of agreement between the output of the SMC and MCMC algorithms.

We are primarily interested in seeing if this alternative adjustment improves the accuracy of the output, specifically with regards to estimating the number of occult individuals, u_t. Previously, although agreement was good, we witnessed some difference between the MCMC and the SMC, especially for the shorter movement step. This is likely as u_t will be highly impacted by the adjustment scheme, despite our best efforts to change the particles as little as possible.

The alternative, 'non-uniform', adjustment scheme may perform stronger as it uses information about the removal process of the individuals when proposing the adjustment. We will again for clarity compare the uniform SMC (U-SMC) to the non-uniform SMC (NU-SMC).

We begin by showing a comparison of the density plots produced using both the uniform and the non-uniform adjustment schemes in Figures 5.16 and 5.17, as well as summarising the results for the NU-SMC algorithm in Table 5.4. As we can see the new adjustment process is a significant improvement on the uniform SMC. This is especially true for the previously most difficult quantity to estimate, u_t. In Figure 5.16 and Table 5.4 we can see that this is true even for the shorter NU-SMC output ($n_p = 200$).

Altogether this suggests that the adjustment step is not moving the particles far from the true posterior distribution and, as such, we obtain accurate estimation of the key parameters. Previously we found that u_t was poorly estimated for the SMC with movement step of length $n_p = 200$, however, this is no longer the case with the alternative adjustment scheme. This is important as the movement step was by far the most time-costly step of the SMC algorithm. Thus, our addition has resulted in more accurate estimates, with a shorter movement step.

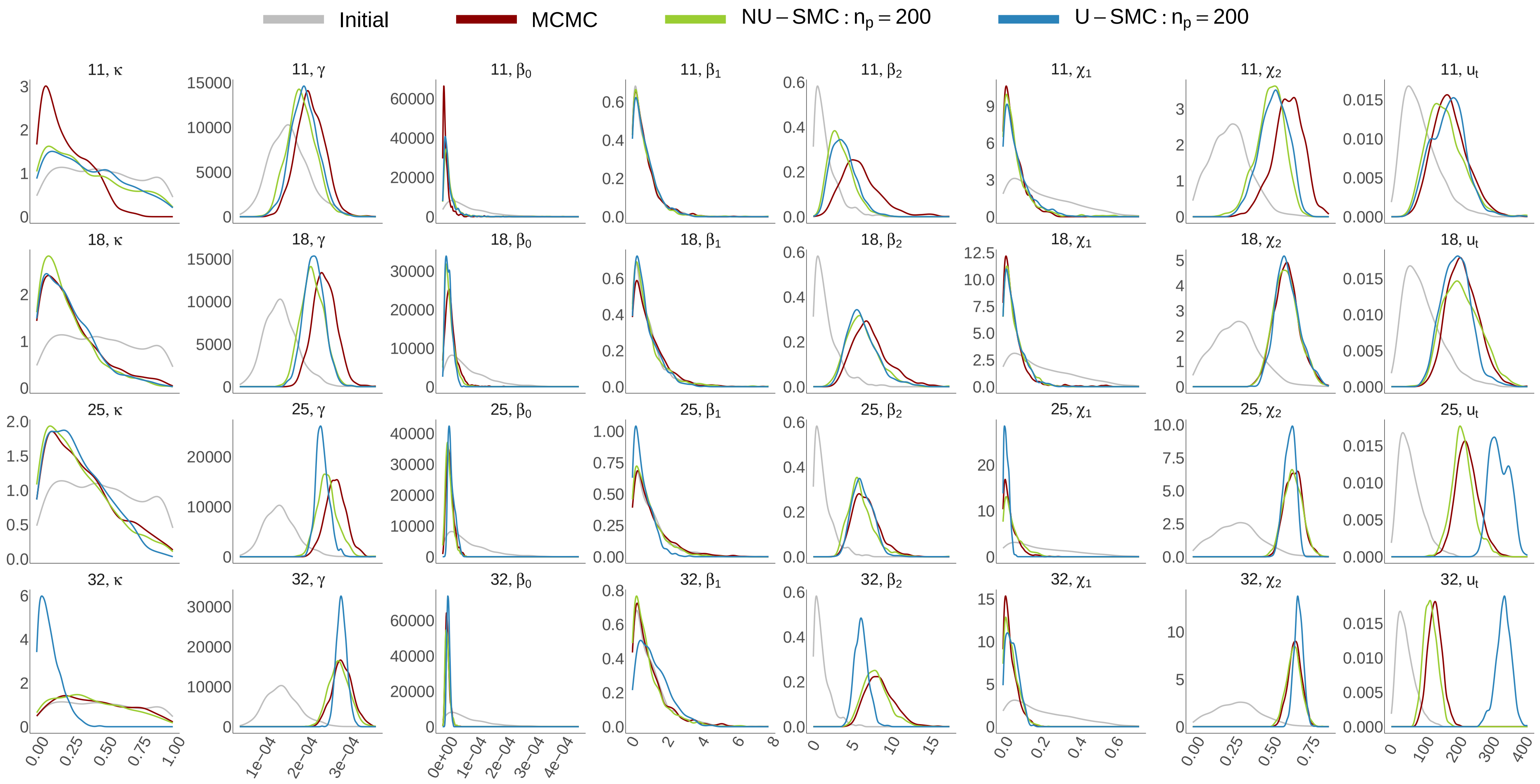


Figure 5.16: A comparison of the densities generated using the outputs from the MCMC and SMC methods applied to the FMD data set. The SMC method has been applied using a uniform and a non-uniform adjustment, both with $n_p = 200$.

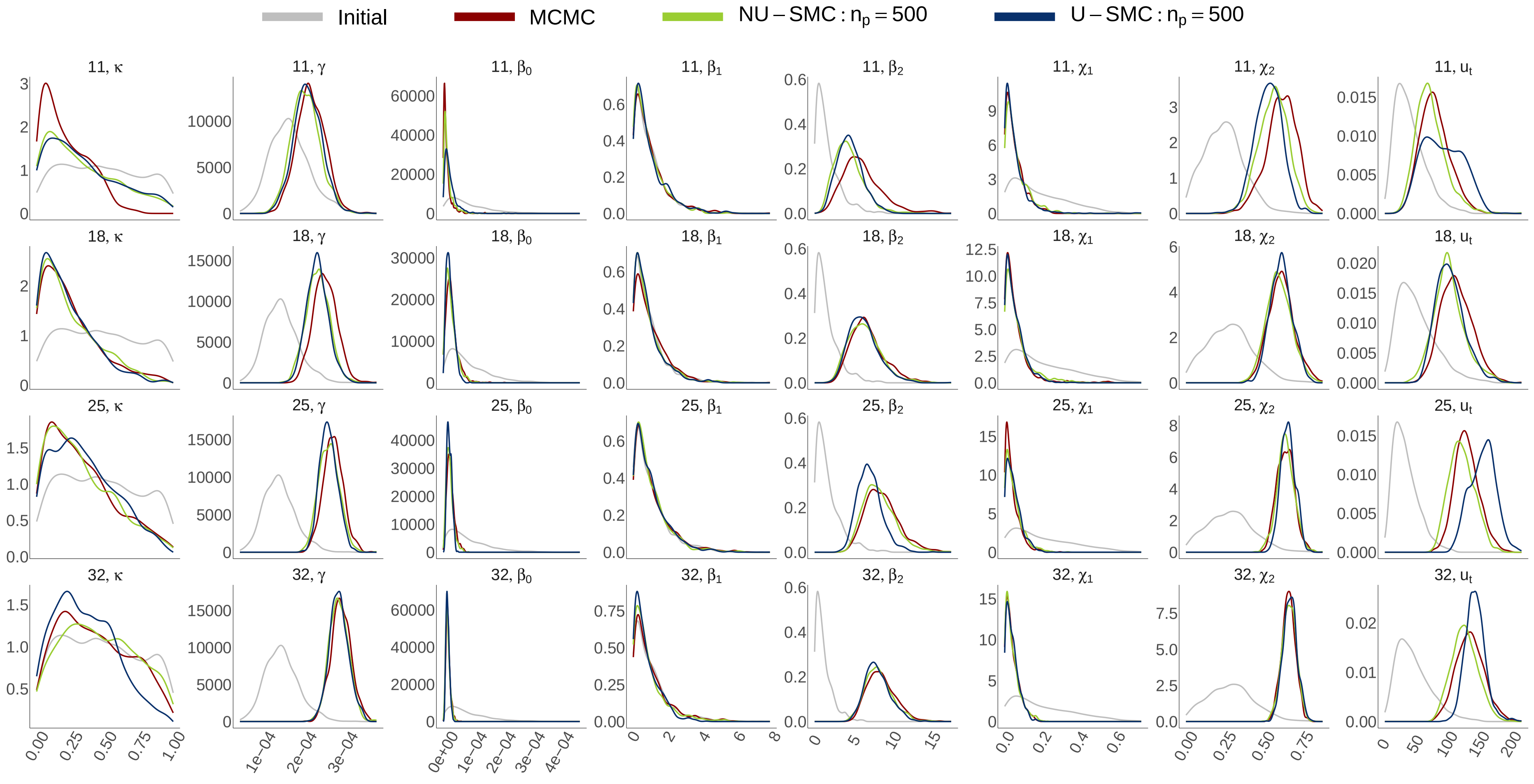


Figure 5.17: A comparison of the densities generated using the outputs from the MCMC and SMC methods applied to the FMD data set. The SMC method has been applied using a uniform and a non-uniform adjustment, both with $n_p = 500$.

		$\boldsymbol{t = 11}$		$\boldsymbol{t = 18}$		$\boldsymbol{t = 25}$		$\boldsymbol{t = 32}$	
		Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.
κ	NU-SMC: $n_p = 200$	0.367	0.276	0.226	0.193	0.312	0.238	0.402	0.255
	NU-SMC: $n_p = 500$	0.335	0.260	0.253	0.214	0.330	0.245	0.464	0.267
	MCMC	0.214	0.164	0.257	0.216	0.337	0.245	0.433	0.262
$\gamma/10^{-4}$	NU-SMC: $n_p = 200$	1.78	0.291	2.30	0.282	2.71	0.258	2.89	0.243
	NU-SMC: $n_p = 500$	1.86	0.292	2.42	0.283	2.81	0.262	2.91	0.235
	MCMC	1.96	0.288	2.60	0.290	2.92	0.255	2.96	0.243
$\beta_0/10^{-5}$	NU-SMC: $n_p = 200$	2.77	3.65	2.48	1.67	2.41	1.25	1.84	0.808
	NU-SMC: $n_p = 500$	1.74	1.85	2.74	1.95	2.42	1.23	1.74	0.730
	MCMC	1.37	1.68	3.01	2.18	2.72	1.30	1.83	0.763
β_1	NU-SMC: $n_p = 200$	1.02	1.07	1.03	1.08	0.895	0.815	0.910	0.931
	NU-SMC: $n_p = 500$	0.947	0.919	1.01	1.03	0.991	0.960	0.889	0.938
	MCMC	1.03	1.03	1.13	1.09	1.05	1.02	0.991	0.991
β_2	NU-SMC: $n_p = 200$	2.54	1.11	4.29	1.42	5.30	1.41	8.01	1.65
	NU-SMC: $n_p = 500$	3.15	1.34	4.69	1.52	5.61	1.38	8.20	1.73
	MCMC	4.35	1.83	4.97	1.57	5.87	1.47	8.55	1.84
χ_1	NU-SMC: $n_p = 200$	0.0756	0.0942	0.0604	0.0548	0.0524	0.0479	0.0476	0.0382
	NU-SMC: $n_p = 500$	0.0712	0.0695	0.0680	0.0724	0.0497	0.0462	0.0425	0.0391
	MCMC	0.0610	0.0557	0.0595	0.0740	0.0406	0.0393	0.0418	0.0361
χ_2	NU-SMC: $n_p = 200$	0.600	0.108	0.630	0.0847	0.652	0.0603	0.658	0.0481
	NU-SMC: $n_p = 500$	0.675	0.112	0.618	0.0836	0.657	0.0578	0.670	0.0470
	MCMC	0.737	0.117	0.633	0.0849	0.657	0.0563	0.666	0.0441
u_t	NU-SMC: $n_p = 200$	86.0	28.1	104	25.9	169	25.6	122	20.5
	NU-SMC: $n_p = 500$	85.9	27.2	94.7	22.3	169	26.9	128	20.3
	MCMC	93.5	26.5	105	22.1	180	26.3	137	21.8

Table 5.4: The mean and standard deviation generated using the non-uniform SMC (NU-SMC), with $n_p = 200$ and $n_p = 500$, and MCMC methods. The results are displayed at times $t = 11\,,18\,,25\,,32$.

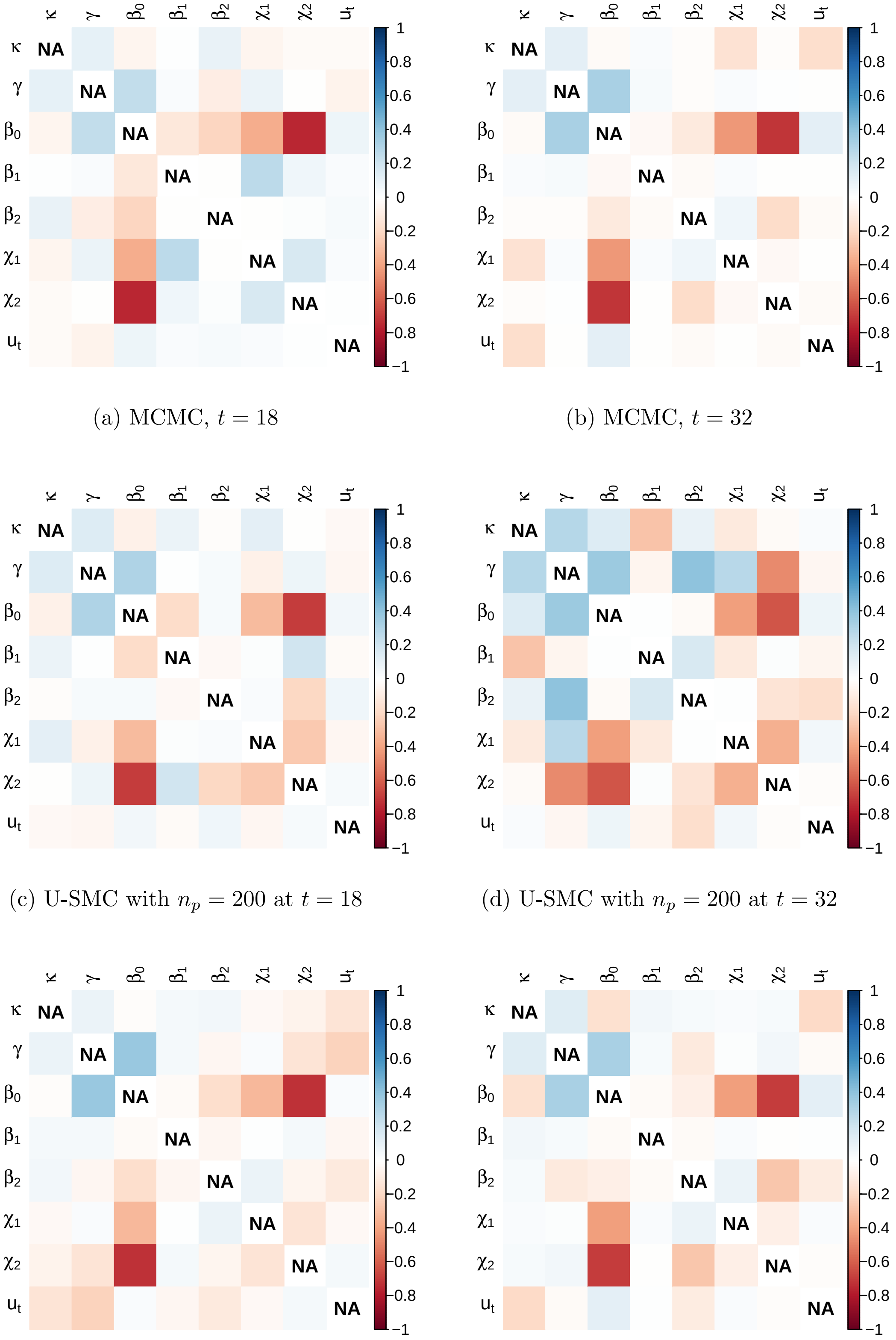
5.7.1 Parameter Relationships

Next we consider the relationship between the parameters. We see in Figure 5.18 that the NU-SMC correctly matches the output of the MCMC. Previously, when considering the uniform-SMC, we found that even when the overall distribution of the parameters was captured, the true relationship between the parameters was not outputted by the shorter movement step ($n_p = 200$). The alternative weighting scheme has rectified this, with the NU-SMC producing accurate estimation of the relationship between the transmission parameters, even for the smaller value of n_p.

5.7.2 Computation Time

Finally, in Chapter 4 we provided an in-depth discussion of the speed increase offered by the SMC. We noted that, as the movement step is dominant, we can compare the number of iterations of the MCMC in the movement step of the SMC, to the number of iterations in the burn-in of the full MCMC. As such, as long as $n_p \times (n/X)$, where X is the number of parallel jobs we can run, is shorter than the length of the MCMC burn-in we will produce the estimates in a shorter amount of time. This is true again here. For example, if we can run 25 parallel jobs then the $t = 32$ iteration of the SMC, with $n_p = 500$, requires at most 7 hours to complete, the MCMC, however, was computed over the course of multiple days, making it unsuitable for real-time analysis. Even if we could construct an MCMC requiring a shorter burn-in period it will still take considerably longer to implement than the SMC method.

Perhaps of greatest importance though, is that we can see in Table 5.5 that we can easily produce results that can be generated within a day and thus we can quickly incorporate the new data. This is key as it means that the SMC is suitable for the real-time analysis of infectious disease outbreaks. We note that, as previously, we assume that all of the longest running particles are contained in the same job. This is an overestimate and if, as in practice will occur, the particles are allocated at random then the expected computation time is much faster, e.g. for $X = 5$ and $n_p = 500$ the expected computation time is actually around 5 hours quicker.



(a) MCMC, $t = 18$

(b) MCMC, $t = 32$

(c) U-SMC with $n_p = 200$ at $t = 18$

(d) U-SMC with $n_p = 200$ at $t = 32$

(e) NU-SMC with $n_p = 200$ at $t = 18$

(f) NU-SMC with $n_p = 200$ at $t = 32$

Figure 5.18: The correlation between the particles generated at times $t = 18$ and $t = 32$, using MCMC, U-SMC and NU-SMC algorithms, where for the latter two methods $n_p = 200$.

	$X = 5$	$X = 10$	$X = 25$	$X = 50$	$X = 100$	$X = 1000$
NU-SMC: $n_p = 200$	12.9	6.7	2.8	1.4	0.7	0.1
NU-SMC: $n_p = 500$	28.8	15.2	6.4	3.4	1.7	0.2

Table 5.5: The (maximum) expected time it will take, in hours, to complete the movement step of the NU-SMC at time $t = 32$, where we split the calculations into X parallel jobs and have $n = 1000$ particles. The burn-in of the comparative MCMC was computed over the course of several days.

5.7.3 Conclusions

Overall the alternative weighting scheme has proven useful, especially in the case of the FMD data set. The non-uniform weighting scheme is likely to be most appropriate when we have longer infectious periods or outbreaks which are very intense i.e. many new cases each day. The ease of introducing this extension (see, Section 3.5) illustrates the flexibility and potential of the SMC algorithm we have constructed. Additionally it has demonstrated the robustness of the algorithm to the use of approximate weights, which we have found to be sufficient in weighting the particles so that they represent samples from the evolving posterior distributions.

Another interesting extension to the SMC algorithm would be the incorporation of the infection process, as well as the removal process, when adjusting the particles. Currently we have only focussed on the removal process as this is significantly simpler than the infection process.

5.8 Discussion

In this chapter we have applied the SMC methods of Chapter 3 to the 2001 UK Foot-and-Mouth outbreak data set, focusing on Cumbria. This data acted as a true test of if the SMC method developed could be applied to data that is less ‘well behaved’. This added difficulty arose in many ways, such as:

- The requirement to incorporate many new observations at each time step.
  - As we incorporate the new data the shape and location of the posterior distribution can change significantly, over a short number of time-steps, which the SMC needed to capture.

- The outbreak occurring on a much larger population, with many occult individuals at each time step.
  - We might expect the SMC algorithm to have greatest difficulty when there are a large number of unknown infectious individuals.

Overall, despite the increase in difficulty, the (uniform) SMC method produced estimates comparable to those generated using the MCMC algorithm. A longer movement step was required to ensure that the two methods concur, however, this coincides with the requirement of a longer burn-in for the MCMC. Additionally, we found that the accuracy of the SMC algorithm when applied to the FMD data set can be improved by using the alternative adjustment scheme. This alternative weighting enabled the shorter SMC to match the MCMC for each parameter and the number of occults, u_t.

Our analysis of the 2001 UK FMD outbreak has contributed to the current understanding of this epidemic. Specifically, we have successfully incorporated information relating to both the notification and removal times of the farms. Although the notification parameter is often difficult to determine its incorporation is important as we produce a model closer to the true behaviour of the outbreak. The incorporation of the notification stage is additionally crucial in determining the importance of the prompt culling of the infected farms.

Overall, despite the difficulties which arise when analysing this data set, the results have shown that the SMC can be utilised in the analysis of ongoing epidemics. At each time step the SMC successfully incorporates multiple new observations and produces samples from the evolving set of posterior distributions. These samples then inform us about the current characteristics of the outbreak, in real-time.

5.8.1 Further Analysis of the FMD Data Set

Our analysis has presented some areas that would be interesting to further consider when analysing this data set.

5.8.1.1 Premise Type

To match with previous work we have only considered the data on those farms removed as infectious premises. However, during this outbreak there were farms removed as a preventative measure—before they were declared infectious. An interesting extension would be to incorporate this additional information and see if it changes our inference. This will increase the accuracy of the model, and thus we will gain an even greater understanding of the propagation of FMD throughout the UK, during the 2001 epidemic.

5.8.1.2 Contact Network

Another interesting extension would be to incorporate the contact network of the farms within the population. There are clear links between certain farms, for example if they have shared external contacts. Therefore, to better understand the spread of FMD, and future outbreaks, incorporating known information about the contact rates between farms would be an interesting addition. This idea was considered for a simulated Avian Influenza outbreak within the UK by Jewell et al. (2009), where the authors incorporated knowledge of the commercial contacts of each premise.

5.8.1.3 Model Selection

Within this example we have used an exponential distance kernel, however, as we previously noted, there has been significant work into finding an appropriate kernel for the Foot-and-Mouth data set. Therefore, it would be interesting to incorporate the ideas of model selection (see, Neal and Roberts (2004), O'Neill and Marks (2005), Clancy and O'Neill (2007), Knock and O'Neill (2014)) and see if the SMC method can be used to determine the kernel best suited to this outbreak. We discuss this idea further in the next chapter.

Chapter 6

Conclusions

6.1 Summary

Within this thesis we have focused on the construction of methods capable of conducting Bayesian inference on a (progressing) infectious disease outbreak.

6.1.1 The MCMC Algorithm

The first algorithm we considered was an adaptive MCMC, which used data augmentation to produce samples from the target posterior distribution. This MCMC was novel in its proposal steps, which allowed the space in which the occult individuals lie to be thoroughly explored. Additionally, the adaptive scheme implemented ensured that we obtained a reasonable acceptance rate, for each example we considered.

The MCMC we have developed is highly flexible and can be applied even in problems with substantial missing data, as is often the case with epidemic data. However, its fundamental weakness is that it needs to be restarted whenever we observe new data. This flaw lead to our discussion of an alternative method which can sequentially incorporate the data as it is received, without the need to restart. Specifically, we were motivated by the desire for a method which could be used in the real-time analysis of an ongoing epidemic. With this in mind we next aimed to adapt sequential Monte Carlo methods, to be used within the context of infectious disease modelling.

6.1.2 The SMC Algorithm

The construction of a suitable sequential method was, unfortunately, not straightforward. The abundance of missing data can result in our previous analysis not always being consistent with the newly observed data. This was the main obstacle in applying SMC methods in conjunction with outbreak data. However, we overcame this incompatibility by incorporating an additional, adjustment, step. This step ensured that the newly observed data and the samples previously generated were fully consistent. The resulting SMC algorithm was as flexible as its MCMC counterpart; additionally, it was constructed in such a way that the usual problems associated with SMC methods, such as particle degeneracy, were not an issue.

When developing the MCMC and SMC algorithms we made sure to keep the form of the transmission probability and the infectious period distribution as general as possible. As a result we can easily incorporate any covariates we are interested in modelling. For example, if desired, we could easily consider the effect of an individuals age on the probability of transmission. Overall both the MCMC and SMC algorithms we have developed can be applied to a wide range of outbreaks, with few rigid assumptions required.

6.1.3 Testing the Methods

The SMC algorithm we have produced required multiple approximations to ensure that the new data and the previous days analysis were compatible. It is important to understand the impact of such approximations, therefore we rigorously tested the SMC on a range of simulated outbreaks. These examples illustrated the effectiveness of the method we had constructed, as well as aiding our understanding of each of the steps of the SMC. We additionally could compare the MCMC and the SMC algorithms, where we found that they matched each other in output, however, the SMC was much quicker due to its parallel nature.

The study of simulated data was important for understanding the strengths of the SMC, however, a true test of the algorithm would be an outbreak that was rapid, with many new cases each day, occurring on a large, heterogeneous population. As such, following the simulated study, we applied the SMC and MCMC algorithms to the 2001 UK Foot-and-Mouth disease outbreak. This was an important test of the capabilities of the SMC methods developed. We found that overall the SMC generated samples which matched those produced by the MCMC. This was particularly impressive as the posterior distribution showed dramatic changes in location and shape, with the incorporation of the new data. However, we found that the SMC required a much longer movement step in order to match the MCMC when predicting the number of occults. This was accompanied, however, by the necessity for a much longer burn-in in the MCMC, which still resulted in the SMC being computationally faster.

Although the SMC was successful in estimating the transmission parameters, the difficulty it faced in estimating the number of occult individuals motivated the use of an alternative method of adjusting the particles, which utilised information about how long individuals had been inferred to be infectious for. This extension proved to be highly successful, with the MCMC and SMC algorithms producing similar results, only with the SMC requiring considerably less time to do so.

6.1.4 Final Conclusions

Overall we have constructed a new method of analysing infectious disease data, which can be applied to an epidemic which is still in progress. The SMC algorithm developed can repeatedly incorporate newly observed data throughout the course of an epidemic, whilst maintaining its accuracy across multiple iterations. Additionally the SMC can be trivially parallelized and thus offers a significant reduction in computation time when compared to alternative methods. Therefore, as well as being novel, it stands to be competitive with the current gold-standard of MCMC algorithms for conducting online inference of infectious disease outbreaks.

6.2 Future Work

Due to its flexibility, the SMC algorithm we have developed is well suited for the introduction of additional steps or extensions. We have already considered a couple of extensions to the SMC algorithm within this thesis, in this section we propose several more—as well as points of interest for work going forward.

6.2.1 The Movement Step

Within the SMC algorithm we include a movement step, the length of which has thus far been chosen such that it is sufficient to ensure we obtain samples from the desired distribution. In the future, however, it would be good to have a greater understanding of this step of the algorithm. Of use would be a metric which informs upon the length of the movement step. For example, we have noted that the SMC has difficulty when there is a large amount of new data, therefore perhaps the length of the movement step should be a function of the number of new observations.

6.2.2 Utilising Research on Sequential Monte Carlo Algorithms

Linked to the improvement of the movement step is the incorporation of the vast work that has been performed in the development of SMC methods.

Previously in Chapter 1 we briefly highlighted some of the proposed extensions to SMC methods. For example, using alternative methods of resampling which can reduce the randomness from the resampling step (Li et al. (2015)), or reducing the particle degeneracy by using the idea of a threshold, such that we only resample in some iterations of the SMC (Liu and Chen (1995), Doucet and Johansen (2011)). These extensions would be simple to incorporate and may offer slight improvements to the methods developed. We do not expect these to have a significant impact, however any reduction in particle degeneracy will aid in improving the SMC algorithm developed.

Next we note that throughout, when performing the movement step, we update the infection times relating to all those individuals who are removed at the time of analysis. However, far into the future we do not expect the new data to change our inference from early on in the outbreak. As such we could use the ideas of *fixed-lag approximation* (see, for example, Doucet and Johansen (2011)) and thus if we began the SMC at time T and we are currently at time $t > T$ then we do not update the infection times relating to those who were removed before time $t - \Delta$, where Δ is to be selected and we assume this is at a time such that $t - \Delta > T$. This will aid in reducing the computation time for the SMC algorithm.

There exists many more extensions to the field of SMC algorithms and thus this would be an interesting area to further consider.

6.2.3 Model Selection

As we have stated throughout, to model an epidemic we must make various assumptions about the outbreak. These can come in many forms, for example deciding which heterogeneities to include, or making assumptions about how the virus develops. This was clearly illustrated when we discussed the FMD data set, for which there has been considerable work performed in choosing an appropriate transmission kernel—which captures the spatial aspect of the spread of the disease. One method of making these decisions is to use expert knowledge and previous work to make an informed choice. Alternatively, by using model selection we can instead let the data itself choose which assumptions to make. For this reason model selection is a well-used method within infectious disease modelling (see, for example, Neal and Roberts (2004), O'Neill and Marks (2005) and Clancy and O'Neill (2007)).

The SMC algorithm developed is well suited for use in conjunction with model selection techniques. Typically these involve using RJ-MCMC methods to allow for jumps between models of varying dimension. Here we describe an alternative approach which could be used with the SMC algorithm developed in Chapter 3. Suppose that we are interested in m models: $\mathcal{M}_1, \ldots \mathcal{M}_m$. We could initialise the SMC by generating n particles from the posterior distribution at time t, under model $\mathcal{M}_i$:

$$\left\{\left(\boldsymbol{\theta}^{i,(j)}, \boldsymbol{y}_{\tau:t}^{i,(j)}, \mathcal{M}_i\right) : j = 1, \ldots, n \quad \text{and} \quad i = 1, \ldots, m\right\}. \tag{6.2.1}$$

Thus we would have a total of $m \times n$ particles under m different models. We could then combine the particles and run the SMC algorithm using all $m \times n$ samples.

During the SMC, we would expect the models least suited to have a lower chance of being resampled. We can then recover the posterior probabilities of the m models and observe which is (are) the preferred model(s) for the data. The advantage to this method of model selection is that we do not need to construct dimension jumping steps as each particle keeps the same model throughout, with changes to the proportion of each model occurring only through the resampling step.

There are some technicalities with this method, such as how to adaptively tune, however, we expect this is simple enough to specify. For example, each model could have its own adaptively tuned parameters and covariance matrix. For example,

$$\hat{\Sigma}^i = \frac{1}{n^i - 1} \sum_{j=1}^{n^i} \left(\boldsymbol{\theta}^{i,(j)} - \bar{\boldsymbol{\theta}}^i\right)\left(\boldsymbol{\theta}^{i,(j)} - \bar{\boldsymbol{\theta}}^i\right)^{\mathrm{T}} \quad \text{for } i = 1, \ldots, m, \tag{6.2.2}$$

where n^i is the number of samples under model i and $\bar{\boldsymbol{\theta}}^i$ is the average parameter values for those samples from model i. These will become progressively worse if there are fewer particles under that model, however, this is to be expected if the model is a poor descriptor of the data.

6.2.4 SEIR Model

Another possible extension is to apply the methods developed to an SEIR compartmental framework. Recall that this has an additional 'exposure' state, during which an individual has been infected but is yet able to infect others. This extension would require the incorporation of additional unknown data in the form of exposure times, for example $\boldsymbol{e}_{\tau:t}$ where now τ would denote the time of the first exposure.

We could model the exposure time by either setting it as a fixed length of time before infection, for example there are x days between exposure and becoming infectious. This is often used within SEIR models, for example Keeling et al. (2001) and Deardon et al. (2010) assumed a fixed period between exposure and infection when modelling the FMD outbreak. Alternatively we could allocate the time between exposure and infectiousness to have some distribution, a function of underlying and possibly unknown parameters.

The SMC algorithm could easily be used to model such an outbreak, with little changes required to be applied. SEIR models are frequently discussed within the epidemic literature thus this extension would prove simple and useful.

6.2.5 Adjustment Step Based on Individual Covariates

In Chapter 3 we discussed an alternative adjustment scheme, which improved the accuracy of the SMC algorithm when applied to the FMD data set in Chapter 5. This extension adjusted particles according to the length of the current occults infectious periods. However, another option would be to perform the adjustment by using knowledge of the characteristics of the individuals themselves.

We could aim to match individuals based on key variables and swap those most similar. For example, suppose we have a sample at time T,

$$\boldsymbol{i}_{\tau:T}^{I,(j)} = \Big\{\{v,w,x,y,z\} : \mathcal{I}_T^{(j)} = \{A,B,C,D,E\}\Big\}$$

and at time $T+1$ we have new a removal, $\mathcal{V}_{T+1} = \{F\}$, such that we need to select one of $\{A,B,C,D,E\}$ to swap with individual F. We could achieve this by weighting the particles according to some 'closeness' to individual F. For example, if we have a highly spatial epidemic then it may be desirable to adjust the particles in such a way that the spatial locations of the infectious individuals does not change significantly. Therefore we could select who to swap with F with weight proportional to a function of their distance e.g. individual A would be adjusted with probability proportional to $w^A \propto d(A,F)^{-1}$ where $d(A,F)$ is the Euclidean distance between individuals A and F.

This extension would be useful for highly heterogeneous populations, where the adjustment step becomes problematic if it results in highly unlikely particles.

6.2.6 Time-Varying Parameters

Throughout we have assumed that the parameters underpinning the epidemic are static, if this is a reasonable assumption then we should witness the SMC converging to a point estimate, with the variance decreasing as we perform multiple iterations. However, if we witness the mean changing through time then this might suggest that the parameters themselves are changing as the outbreak progresses. This is an interesting advantage of using the SMC algorithm, we can directly witness changes in behaviour. Therefore, analysis of simulated data generated with parameters which vary through time would be an interesting study on the behaviour of the SMC; for example, how quickly can it detect a change in behaviour?

Another, simple, extension is the incorporation of parameters which change dependent on the behaviour and characteristics of the individual. So far, in our simulations, we have used the transmission probability

$$P(i \text{ infects } j) = (1-p)\exp\{-\gamma d(i,j)\}. \tag{6.2.3}$$

We expect that an individual may not have constant infectiousness throughout the course of their infection. Therefore, we could instead model parameter p as dependent on how many days an individual has been infectious for. This could represent various mechanisms, such as a host becoming less infectious as the disease progresses in their body, or the reduced chance of spreading the disease to others once symptoms begin to show.

In many ways we have already considered this by including a notification period, this can been seen as a change point at which the infectiousness of an individual takes a different form. We are therefore proposing to extend this idea by allowing the transmission probability to instead be a function of how long that individual has been infectious for. For example

$$P(i \text{ infects } j\,;\, t) = (1 - p_t(i)) \exp\{-\gamma d(i,j)\}, \tag{6.2.4}$$

where $p_t(i)$ depends on how many days individual i has been infectious for at time t.

There exist many more extensions we could consider for the constructed SMC algorithm: its strength lies in its ability to be adapted for different epidemics, with a range of characteristics. It is this flexibility, as well as the viability of on-line inference, that make the SMC method competitive with the current gold standard of MCMC algorithms for applications within stochastic epidemic modelling.

Appendix A

Appendix of Additional Calculations

A.1 Alternative Likelihood Calculation

Calculation of the likelihood can be slow: this is especially true in large populations where the number of infectious individual is small when compared to the total population. In this section we describe a method of reducing the computation time of this calculation.

We denote by Q_t the contribution to the likelihood at time t, such that

$$Q_t = \underbrace{\prod_{\ell \in \mathcal{S}_{t+1}} \left(\prod_{k \in \mathcal{I}_t} p_t(\ell, k) \right)}_{\text{Main Calculation}} \underbrace{\prod_{\ell \in \mathcal{S}_t \setminus \mathcal{S}_{t+1}} \left(1 - \prod_{k \in \mathcal{I}_t} p_t(\ell, k) \right)}_{A_t}. \tag{A.1.1}$$

The majority of the time spent in computing (A.1.1) is in calculating the probability that each susceptible individual avoids infection. Therefore if we can avoid repeating this calculation we may reduce the overall computation time.

We wish to determine the relationship between Q_t and Q_{t+1}. We begin re-writing

(A.1.1) as

$$
\begin{aligned}
Q_t \;=& \prod_{\ell\in\mathcal{S}_{t+1}\cap\mathcal{S}_{t+2}} \left(\prod_{k\in\mathcal{I}_t\cap\mathcal{I}_{t+1}} p_t(\ell,k) \right) \\
&\times \underbrace{\prod_{\ell\in\mathcal{S}_{t+1}\cap\mathcal{S}_{t+2}} \left(\prod_{k\in\mathcal{I}_t\setminus\mathcal{I}_{t+1}} p_t(\ell,k) \right)}_{B_t} \times \underbrace{\prod_{\ell\in\mathcal{S}_{t+1}\setminus\mathcal{S}_{t+2}} \left(\prod_{k\in\mathcal{I}_t} p_t(\ell,k) \right)}_{C_t} \times A_t \\
=& A_t \,\times\, B_t \,\times C_t \,\times \prod_{\ell\in\mathcal{S}_{t+1}\cap\mathcal{S}_{t+2}} \left(\prod_{k\in\mathcal{I}_t\cap\mathcal{I}_{t+1}} p_t(\ell,k) \right).
\end{aligned}
$$

Similarly we can rewrite Q_{t+1}, noting that $\mathcal{S}_{t+2}\cap\mathcal{S}_{t+1}=\mathcal{S}_{t+2}$ and $\mathcal{S}_{t+2}\setminus\mathcal{S}_{t+1}=\emptyset$,

$$
\begin{aligned}
Q_{t+1} \;=& \prod_{\ell\in\mathcal{S}_{t+1}\cap\mathcal{S}_{t+2}} \left(\prod_{k\in\mathcal{I}_t\cap\mathcal{I}_{t+1}} p_{t+1}(\ell,k) \right) \underbrace{\prod_{\ell\in\mathcal{S}_{t+1}\cap\mathcal{S}_{t+2}} \left(\prod_{k\in\mathcal{I}_{t+1}\setminus\mathcal{I}_t} p_{t+1}(\ell,k) \right)}_{E_{t+1}} \times A_{t+1} \\
=& \; A_{t+1} \,\times\, E_{t+1} \,\times \prod_{\ell\in\mathcal{S}_{t+1}\cap\mathcal{S}_{t+2}} \left(\prod_{k\in\mathcal{I}_t\cap\mathcal{I}_{t+1}} p_{t+1}(\ell,k) \right).
\end{aligned}
$$

We can therefore relate Q_t and Q_{t+1} in the following way,

$$
Q_{t+1} \;=\; Q_t \,\times\, \left(\frac{A_{t+1} \,\times\, E_{t+1}}{A_t \,\times\, B_t \,\times\, C_t} \right). \tag{A.1.2}
$$

Note: the different time subscripts, p_t and p_{t+1}, do not affect the cancellation as they simply denote that the probability of avoiding infection will be dependent on which individuals are infectious at each time step. Using this relationship we can calculate part of the likelihood sequentially. This may not always offer a speed increase: however, it will be useful when individuals have long infectious periods or we have a large susceptible population.

A.2 Status Changes

In this section we formally describe how we adjust the particles, to that ensure they are consistent with the newly observed data, specifically focusing on how we change the status of individuals. For clarity we drop the superscript denoting which particle we are adjusting.

Consider a particle with occult information, $\boldsymbol{y}_{\tau:T} = \{\boldsymbol{i}^I_{\tau:T}, \boldsymbol{i}^R_{\tau:T}\}$, that is not consistent with the newly observed removals, $\boldsymbol{r}_{T+1}$. Let $v_{T+1} = |\boldsymbol{r}_{T+1}|$ such that we have observed v_{T+1} new removals at time $T+1$. As stated we wish to change this particle as little as possible, therefore we keep the number of occult individuals, u_T, the same and do not change the infection times themselves. Rather, we look at swapping the individuals attached to these infection times so that the particle and the new observations are consistent. This will effectively only change the status of a small portion of the individuals and not the parameter values themselves.

If $v_{T+1} > u_T$ then, without changing the particle significantly, we cannot correct it to be compatible with all of the data. Therefore, we will give it a weighting of zero. If $v_{T+1} \leq u_T$ then we can adjust the particle. Supposing that the latter is true, the idea is to change the labellings of those individuals newly observed to be removed to have been occult individuals at time T. As such their status at time T will change from S to I. To counteract this we will also select the same number of occult individuals at time T to swap their status from I to S, so that the value of u_T remains the same.

Formally let $\mathcal{V}_{T+1}$ denote the set of indexes of those individuals that are newly removed at time $T+1$ such that $\mathcal{V}_{T+1} = \mathcal{R}_{T+1} \backslash \mathcal{R}_T$. Recall that we denote the indexes of the occult individuals and the susceptible individuals at time T by $\mathcal{I}_T$ and $\mathcal{S}_T$ respectively, where both of these depend on which particle we are adjusting. If $\mathcal{V}_{T+1} \subseteq \mathcal{I}_T$ then this particle and the new data are consistent and therefore it does not need to be altered. However, if $\mathcal{V}_{T+1} \nsubseteq \mathcal{I}_T$ then the particle and the new data are not compatible with one another. Supposing that the latter is true let G be a random sample of individuals from the set $\mathcal{I}_T \backslash \mathcal{V}_{T+1}$ of size $|G| = |\mathcal{I}_T| - |\mathcal{V}_{T+1}|$. We then define the corrected set of occult individuals at time T as $\tilde{\mathcal{I}}_T = \mathcal{V}_{T+1} \cup G$. This is such that $|\tilde{\mathcal{I}}_T| = |\mathcal{I}_T|$ and $\mathcal{V}_{T+1} \subseteq \tilde{\mathcal{I}}_T$. This leaves those in the set $H = G^c \cap (\mathcal{I}_T \backslash \mathcal{V}_{T+1})$ as those whose status will change to susceptible. We set $\tilde{\mathcal{S}}_T = (\mathcal{S}_T \backslash \mathcal{V}_{T+1}) \cup H$, which we can again show to be of a consistent size. Therefore, $\tilde{\mathcal{S}}_T$ and $\tilde{\mathcal{I}}_T$ are now fully consistent with the newly observed data, whilst retaining much of the same information.

In the amendment process we are only changing the indexes in each of the three states, as such the infection times do not change, they are just attached to different individuals.

If we let $D = \mathcal{V}_{T+1} \backslash \mathcal{I}_T$ then the set H contains the indexes of those individuals whose status will change from I to S and the set D contains those individuals moving from S to I. Thus, the infection times of those in H become the infection times for those in D.

Bibliography

Aalen, O., Borgan, O., and Gjessing, H. (2008). *Survival and event history analysis: A process point of view.* Springer-Verlag, New York.

Abbey, H. (1952). An examination of the Reed-Frost theory of epidemics. *Human Biology*, 24(3):201–233.

Alexandersen, S., Zhang, Z., Donaldson, A. I., and Garland, A. J. M. (2003). The pathogenesis and diagnosis of foot-and-mouth disease. *Journal of Comparative Pathology*, 129(1):1–36.

Andersson, H. (1998). Limit theorems for a random graph epidemic model. *Annals of Applied Probability*, 8(4):1331–1349.

Andersson, H. and Britton, T. (2000). *Stochastic epidemic models and their statistical analysis*, volume 151 of *Lecture Notes in Statistics.* Springer-Verlag, New York.

Bailey, N. T. J. (1950). A simple stochastic epidemic. *Biometrika*, 37(3–4):193–202.

Bailey, N. T. J. (1967). The simulation of stochastic epidemics in two dimensions. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 4, pages 237–257. University of California Press Berkeley and Los Angeles.

Bailey, N. T. J. (1975). *The mathematical theory of infectious diseases and its applications.* Griffin, 2nd edition.

Bailey, N. T. J. and Thomas, A. S. (1971). The estimation of parameters from population data on the general stochastic epidemic. *Theoretical Population Biology*, 2(3):253–270.

Bailey, W. N. (1964). *Generalized hypergeometric series.* Cambridge tracts in mathematics and mathematical physics. Stechert-Hafner Service Agency.

Ball, F. (1983a). The threshold behaviour of epidemic models. *Journal of Applied Probability*, 20(2):227–241.

Ball, F. (1983b). A threshold theorem for the Reed-Frost chain-binomial epidemic. *Journal of Applied Probability*, 20(1):153–157.

Ball, F. (1985). Deterministic and stochastic epidemics with several kinds of susceptibles. *Advances in Applied Probability*, 17(1):1–22.

Ball, F. (1986). A unified approach to the distribution of total size and total area under the trajectory of infectives in epidemic models. *Advances in Applied Probability*, 18(2):289–310.

Ball, F., Mollison, D., and Scalia-Tomba, G. (1997). Epidemics with two levels of mixing. *The Annals of Applied Probability*, 7(1):46–89.

Ball, F. and Neal, P. (2008). Network epidemic models with two levels of mixing. *Mathematical Biosciences*, 212(1):69–87.

Ball, F. G., Knock, E. S., and O'Neill, P. D. (2011). Threshold behaviour of emerging epidemics featuring contact tracing. *Advances in Applied Probability*, 43(4):1048–1065.

Ball, F. G., Knock, E. S., and O'Neill, P. D. (2015). Stochastic epidemic models featuring contact tracing with delays. *Mathematical Biosciences*, 266:23–35.

Bartlett, M. S. (1949). Some evolutionary stochastic processes. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 11(2):211–229.

Becker, N. (1979). An estimation procedure for household disease data. *Biometrika*, 66(2):271–277.

Becker, N. G. (1989). *Analysis of infectious disease data*. CRC Monographs on Statistics & Applied Probability. Chapman and Hall.

Becker, N. G. and Britton, T. (1999). Statistical studies of infectious disease incidence. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(2):287–307.

Bellman, R. (1961). *Adaptive control processes: A guided tour.* Princeton University Press.

Berzuini, C., Best, N. G., Gilks, W. R., and Larizza, C. (1997). Dynamic conditional independence models and Markov chain Monte Carlo methods. *Journal of the American Statistical Association*, 92(440):1403–1412.

Birrell, P. J., De Angelis, D., Wernisch, L., Tom, B. D., Roberts, G. O., and Pebody, R. G. (2016). Efficient real-time monitoring of an emerging influenza epidemic: how feasible? *arXiv preprint arXiv:1608.05292*.

Britton, T. (2010). Stochastic epidemic models: A survey. *Mathematical Biosciences*, 225(1):24–35.

Britton, T. and Becker, N. G. (2000). Estimating the immunity coverage required to prevent epidemics in a community of households. *Biostatistics*, 1(4):389–402.

Britton, T., Kypraios, T., and O'Neill, P. D. (2011). Inference for epidemics with three levels of mixing: Methodology and application to a measles outbreak. *Scandinavian Journal of Statistics*, 38(3):578–599.

Britton, T. and O'Neill, P. D. (2002). Bayesian inference for stochastic epidemics in populations with random social structure. *Scandinavian Journal of Statistics*, 29(3):375–390.

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011). *Handbook of Markov chain Monte Carlo*. CRC press.

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

Clancy, D. and O'Neill, P. D. (2007). Exact Bayesian inference and model selection for stochastic models of epidemics among a community of households. *Scandinavian Journal of Statistics*, 34(2):259–274.

Daley, D. J. and Gani, J. (2001). *Epidemic modelling: An introduction*, volume 15 of *Cambridge studies in mathematical biology*. Cambridge University Press.

Danon, L., Ford, A. P., House, T., Jewell, C. P., Keeling, M. J., Roberts, G. O., Ross, J. V., and Vernon, M. C. (2011). Networks and the epidemiology of infectious disease. *Interdisciplinary Perspectives on Infectious Diseases*, 2011.

Deardon, R., Brooks, S. P., Grenfell, B. T., Keeling, M. J., Tildesley, M. J., Savill, N. J., Shaw, D. J., and Woolhouse, M. E. J. (2010). Inference for individual-level models of infectious diseases in large populations. *Statistica Sinica*, 20(1):239–261.

del Rey, A. M. (2015). Mathematical modeling of the propagation of malware: a review. *Security and Communication Networks*, 8(15):2561–2579.

Demiris, N. and O'Neill, P. D. (2005). Bayesian inference for stochastic multitype epidemics in structured populations via random graphs. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(5):731–745.

Demiris, N. and O'Neill, P. D. (2006). Computation of final outcome probabilities for the generalised stochastic epidemic. *Statistics and Computing*, 16(3):309–317.

Devroye, L. (1986). *Non-uniform random variate generation*. Springer-Verlag New York.

Diekmann, O., Heesterbeek, H., and Britton, T. (2012). *Mathematical tools for understanding infectious disease dynamics*. Princeton University Press.

Diekmann, O. and Heesterbeek, J. A. P. (2000). *Mathematical epidemiology of infectious diseases: Model building, analysis and interpretation*. John Wiley.

Diggle, P. J. (2006). Spatio-temporal point processes, partial likelihood, foot and mouth disease. *Statistical Methods in Medical Research*, 15(4):325–336.

Doucet, A., de Freitas, N., and Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. Information Science and Statistics. Springer-Verlag New York.

Doucet, A., Godsill, S., and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208.

Doucet, A. and Johansen, A. M. (2011). A tutorial on particle filtering and smoothing: Fifteen years later. In *The Oxford Handbook of Nonlinear Filtering*, page 656–704. Oxford University Press.

Ferguson, N. M., Donnelly, C. A., and Anderson, R. M. (2001a). The foot-and-mouth epidemic in Great Britain: Pattern of spread and impact of interventions. *Science*, 292(5519):1155–1160.

Ferguson, N. M., Donnelly, C. A., and Anderson, R. M. (2001b). Transmission intensity and impact of control policies on the foot and mouth epidemic in Great Britain. *Nature*, 413(6855):542–548.

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association*, 85(410):398–409.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741.

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 57(6):1317–1339.

Gibson, G. J. (1997). Markov chain Monte Carlo methods for fitting spatiotemporal stochastic models in plant epidemiology. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 46(2):215–233.

Gibson, G. J. and Renshaw, E. (1998). Estimating parameters in stochastic compartmental models using Markov chain methods. *IMA Journal of Mathematics Applied in Medicine & Biology*, 15(1):19–40.

Gibson, G. J., Streftaris, G., and Thong, D. (2018). Comparison and assessment of epidemic models. *Statistical Science*, 33(1):19–33.

Gilks, W. R. and Berzuini, C. (2001). Following a moving target—Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):127–146.

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

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

Groendyke, C., Welch, D., and Hunter, D. R. (2011). Bayesian inference for contact networks given epidemic data. *Scandinavian Journal of Statistics*, 38(3):600–616.

Haario, H., Saksman, E., and Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242.

Halton, J. H. (1970). A retrospective and prospective survey of the Monte Carlo method. *SIAM Review*, 12(1):1–63.

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

Heesterbeek, J. A. P. (2002). A brief history of R_0 and a recipe for its calculation. *Acta Biotheoretica*, 50(3):189–204.

Isham, V. (1991). Assessing the variability of stochastic epidemics. *Mathematical Biosciences*, 107(2):209–224.

Isham, V. (2005). Stochastic models for epidemics. In *Celebrating statistics : papers in honour of Sir David Cox on his 80th birthday*, chapter 1, pages 27–54.

Jewell, C. P., Kypraios, T., Neal, P., and Roberts, G. O. (2009). Bayesian analysis for emerging infectious diseases. *Bayesian Analysis*, 4(3):465–496.

Jombart, T., Cori, A., Didelot, X., Cauchemez, S., Fraser, C., and Ferguson, N. (2014). Bayesian reconstruction of disease outbreaks by combining epidemiologic and genomic data. *PLoS Computational Biology*, 10.

Keeling, M. J. (2005). Models of foot-and-mouth disease. *Proceedings of the Royal Society B: Biological Sciences*, 272(1569):1195–1202.

Keeling, M. J., Woolhouse, M. E. J., Shaw, D. J., Matthews, L., Chase-Topping, M., Haydon, D. T., Cornell, S. J., Kappey, J., Wilesmith, J., and Grenfell, B. T. (2001). Dynamics of the 2001 UK foot and mouth epidemic: Stochastic dispersal in a heterogeneous landscape. *Science*, 294(5543):813–817.

Kermack, W. O. and McKendrick, A. G. (1927). A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 115(772):700–721.

King, A. A., Nguyen, D., Ionides, E. L., et al. (2016). Statistical inference for partially observed Markov processes via the R package pomp. *Journal of Statistical Software*, 69(i12).

Knock, E. S. and O'Neill, P. D. (2014). Bayesian model choice for epidemic models with two levels of mixing. *Biostatistics*, 15(1):46–59.

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

Kypraios, T. (2007). *Efficient Bayesian inference for partially observed stochastic epidemics and a new class of semi-parametric time series models.* PhD thesis, Lancaster University.

Kypraios, T., Neal, P., and Prangle, D. (2017). A tutorial introduction to Bayesian inference for stochastic epidemic models using Approximate Bayesian Computation. *Mathematical Biosciences*, 287:42–53.

Kypraios, T. and O'Neill, P. D. (2018). Bayesian nonparametrics for stochastic epidemic models. *Statistical Science*, 33(1):44–56.

Lee, C. and Neal, P. J. (2018). Optimal scaling of the independence sampler: Theory and practice. *Bernoulli*, 24(3):1636–1652.

Li, T., Bolic, M., and Djuric, P. M. (2015). Resampling methods for particle filtering: Classification, implementation, and strategies. *IEEE Signal Processing Magazine*, 32(3):70–86.

Liu, J. S. (1994). The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, 89(427):958–966.

Liu, J. S. (2008). *Monte Carlo strategies in scientific computing.* Springer Science & Business Media.

Liu, J. S. and Chen, R. (1995). Blind deconvolution via sequential imputations. *Journal of the American Statistical Association*, 90(430):567–576.

Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044.

Longini Jr, I. M. (1980). A chain binomial model of endemicity. *Mathematical Biosciences*, 50(1-2):85–93.

Malice, M.-P. and Lefevre, C. (1985). On linear stochastic compartmental models in discrete time. *Bulletin of Mathematical Biology*, 47(2):287–293.

Marion, G., Gibson, G., and Renshaw, E. (2003). Estimating likelihoods for spatio-temporal models using importance sampling. *Statistics and Computing*, 13(2):111–119.

Martinussen, T. and Scheike, T. H. (2007). *Dynamic regression models for survival data*. Springer Science & Business Media.

McKinley, T., Cook, A. R., and Deardon, R. (2009). Inference in epidemic models without likelihoods. *The International Journal of Biostatistics*, 5(1).

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

Neal, P. (2012). Efficient likelihood-free Bayesian Computation for household epidemics. *Statistics and Computing*, 22(6):1239–1256.

Neal, P. (2016). A household SIR epidemic model incorporating time of day effects. *Journal of Applied Probability*, 53(2):489–501.

Neal, P. and Roberts, G. (2005). A case study in non-centering for data augmentation: Stochastic epidemics. *Statistics and Computing*, 15(4):315–327.

Neal, P. and Xiang, F. (2017). Collapsing of non-centred parameterized MCMC algorithms with applications to epidemic models. *Scandinavian Journal of Statistics*, 44(1):81–96.

Neal, P. J. and Roberts, G. O. (2004). Statistical inference and model selection for the 1861 Hagelloch measles epidemic. *Biostatistics*, 5(2):249–261.

OIE (2013). Foot and mouth disease: Aetiology, epidemiology, diagnosis, prevention, and control reference. `http://www.oie.int/fileadmin/Home/eng/Animal_Health_in_the_World/docs/pdf/Disease_cards/FOOT_AND_MOUTH_DISEASE.pdf`.

O'Neill, P. D. (2009). Bayesian inference for stochastic multitype epidemics in structured populations using sample data. *Biostatistics*, 10(4):779–791.

O'Neill, P. D. (2010). Introduction and snapshot review: Relating infectious disease transmission models to data. *Statistics in Medicine*, 29(20):2069–2077.

O'Neill, P. D., Balding, D. J., Becker, N. G., Eerola, M., and Mollison, D. (2000). Analyses of infectious disease data from household outbreaks by Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 49(4):517–542.

O'Neill, P. D. and Becker, N. G. (2001). Inference for an epidemic when susceptibility varies. *Biostatistics*, 2(1):99–108.

O'Neill, P. D. and Marks, P. J. (2005). Bayesian model choice and infection route modelling in an outbreak of Norovirus. *Statistics in Medicine*, 24(13):2011–2024.

O'Neill, P. D. and Roberts, G. O. (1999). Bayesian inference for partially observed stochastic epidemics. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 162(1):121–129.

O'Neill, P. D. and Wen, C. H. (2012). Modelling and inference for epidemic models featuring non-linear infection pressure. *Mathematical Biosciences*, 238(1):38–48.

Oxford Dictionaries (2018). Oxford dictionary of English.

Papaspiliopoulos, O. (2003). *Non-centered parameterisations for data augmentation and hierarchical models*. PhD thesis, Lancaster University.

Papaspiliopoulos, O., Roberts, G. O., and Sköld, M. (2007). A general framework for the parametrization of hierarchical models. *Statistical Science*, pages 59–73.

Rampey Jr, A. H., Longini Jr, I. M., Haber, M., and Monto, A. S. (1992). A discrete-time model for the statistical analysis of infectious disease incidence data. *Biometrics*, 48(1):117–128.

Robert, C. and Casella, G. (2005). *Monte Carlo statistical methods*. Springer Texts in Statistics. Springer New York.

Robert, C. and Casella, G. (2011). A short history of Markov chain Monte Carlo: Subjective recollections from incomplete data. *Statistical Science*, 26(1):102–115.

Roberts, G. O., Gelman, A., and Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1):110–120.

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

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

Roberts, G. O. and Smith, A. F. M. (1994). Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms. *Stochastic Processes and their Applications*, 49(2):207–216.

Roberts, M., Andreasen, V., Lloyd, A., and Pellis, L. (2015). Nine challenges for deterministic epidemic models. *Epidemics*, 10:49–53.

Savill, N. J., Shaw, D. J., Deardon, R., Tildesley, M. J., Keeling, M. J., Woolhouse, M. E. J., Brooks, S. P., and Grenfell, B. T. (2006). Topographic determinants of foot and mouth disease transmission in the UK 2001 epidemic. *BMC Veterinary Research*, 2(3).

Sherlock, C. (2006). *Methodology for inference on the Markov modulated Poisson process and theory for optimal scaling of the random walk Metropolis*. PhD thesis, Lancaster University.

Smith, A. F. M. and Gelfand, A. E. (1992). Bayesian statistics without tears: A sampling–resampling perspective. *The American Statistician*, 46(2):84–88.

Stockdale, J. E., Kypraios, T., and O'Neill, P. D. (2017). Modelling and Bayesian analysis of the Abakaliki smallpox data. *Epidemics*, 19:13–23.

Streftaris, G. and Gibson, G. J. (2004). Bayesian inference for stochastic epidemics in closed populations. *Statistical Modelling*, 4(1):63–75.

Tanner, M. A. and Wong, W. H. (1987). The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82(398):528–540.

Tierney, L. (1994). Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22(4):1701–1728.

UK National Audit Office (2002). The 2001 outbreak of foot and mouth disease (executive summary). Report by the Comptroller and auditor general. `https://www.nao.org.uk/report/the-2001-outbreak-of-foot-and-mouth-disease/`.

von Neumann, J. (1951). Various techniques used in connection with random digits. *Journal of Research of the National Bureau of Standards*, 12:36–38.

Wang, X., Li, T., Sun, S., and Corchado, J. M. (2017). A survey of recent advances in particle filters and remaining challenges for multitarget tracking. *Sensors*, 17(12):2707.

Weiss, G. H. and Dishon, M. (1971). On the asymptotic behavior of the stochastic and deterministic models of an epidemic. *Mathematical Biosciences*, 11(3-4):261–265.

Whittle, P. (1955). The outcome of a stochastic epidemic—a note on Bailey's paper. *Biometrika*, 42(1-2):116–122.

WHO (2016). Ebola situation reports. `http://www.who.int/csr/disease/ebola/situation-reports/archive/en/`.

WHO (2018). Ebola virus disease fact sheet. `http://www.who.int/news-room/fact-sheets/detail/ebola-virus-disease`.

Xiang, F. and Neal, P. J. (2014). Efficient MCMC for temporal epidemics via parameter reduction. *Computational Statistics and Data Analysis*, 80:240–250.

Xu, X., Kypraios, T., and O'Neill, P. D. (2016). Bayesian non-parametric inference for stochastic epidemic models using Gaussian Processes. *Biostatistics*, 17(4):619–633.