AMSI
AUSTRALIAN MATHEMATICAL SCIENCES INSTITUTE

National Collaboration in the Mathematical Sciences

Application of approximate Bayesian computation to estimate parameters in models of infectious disease spread on a network

Brock Hermans

February 28, 2013

Contents

Postal Address: 111 Barry Street
c/- The University of Melbourne
Victoria 3010 Australia

Email: enquiries@
Phone: +61 3 834
Fax: +61 3 934
Web: www.amsi

1 Introduction

In many epidemiological studies one of the key questions we want to ask is, what is the infectivity rate of a particular disease? That is, at what rate do people become infected by a disease. Knowing such information can help us to implement strategies in order to prevent a disease from spreading through a population. This report will lay the foundations for a study into a disease spread on a network of nodes, or people. In the report we will look at approximate Bayesian computation (ABC); a method in which we can estimate the parameters of a model. In addition, we will look at some of the new research into ABC methods. We will then attempt to apply ABC to a particular SEIR model; a continuous time Markov chain model which uses three parameters to describe how a disease can move through a population. We will finish by examining some of the difficulties in applying our ABC method to the SEIR model.

2 Bayesian statistics

Bayesian statistics is a school of thought that interprets the broad idea of 'statistics' differently from the more well known Frequentist statistics (which involves unbiased estimators, confidence intervals and p-values). In its most general form, Bayesian statistics formalises the scientific principle of having a hypothesis, collecting some data, and then updating your hypothesis, based on the data you collected. It's origins lie in the famous Bayes' theorem, which states:

Theorem 1. *Consider events A and B.*

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)},$$

if $P(B) > 0$.

The theorem, named after the famous English mathematician Thomas Bayes (1701-1761), is fundamental to Bayesian statistics. Consider reformulating Bayes' theorem in the following way. Suppose we have some experiment, which we believe can be modelled by the set of parameters, θ. Let us also suppose that we observe some set of data, y. We now wish to update our belief about θ, which we can do using Bayes' theorem. Suppose that we have the probability model $f(y|\theta)$ and that our belief about the set of parameters can be given by $\pi(\theta)$, which we call the prior density. Using Bayes' theorem we have:

$$\pi(\theta|y) = \frac{\pi(\theta)f(y|\theta)}{\int \pi(\theta)f(y|\theta)d\theta}$$

[Christensen et al., 2011]

By noticing that $f(y|\theta)$ is the likelihood of the data y (likelihood is equivalent to the probability of observing the data given a set of parameters θ) and the integral on the denominator is simply a constant (normalising constant), then we interpret the above formula as

$$\text{posterior} \propto \text{prior} \times \text{likelihood}.$$

Here our posterior density, $\pi(\theta|y)$, is a summary of our beliefs about the parameter θ after having observed some data, y. This fundamental idea is rejected by some Frequentist statisticians, however we won't go into the Bayesian vs Frequentist debate. What we will say however is that Bayesian statisticians view parameters of a model as random variables. These random variables can be updated via Bayes' theorem, and the concept of probability is a measure of personal belief. On the other hand, Frequentist statisticians view parameters of a model as unknown constants that can be estimated using unbiased estimators and confidence intervals. Because of this, Frequentist statisticians need to recalculate their estimate every time they gain some new information, rather than simply updating their belief as a Bayesian statistician would.

Because of this difference of interpretation, we have the Bayesian version of confidence intervals (credible intervals) and p-values (Bayes' factors). We will not discuss how to calculate credible intervals and Bayes' factors, but we will say that they are used in inference and will be discussed later.

To get a better understanding of Bayesian statistics, we consider the following example. Say we have a coin and we want to calculate the probability of a head, p. We are told that after 20 tosses we have 2 heads. In this case our prior would be a standard uniform, as the only thing we know about p is that it is a probability and thus lies between 0 and 1 (a belief that is contained in a standard uniform density). Calculating our posterior density, we get a beta density with parameters $\alpha = 3$ and $\beta = 19$.

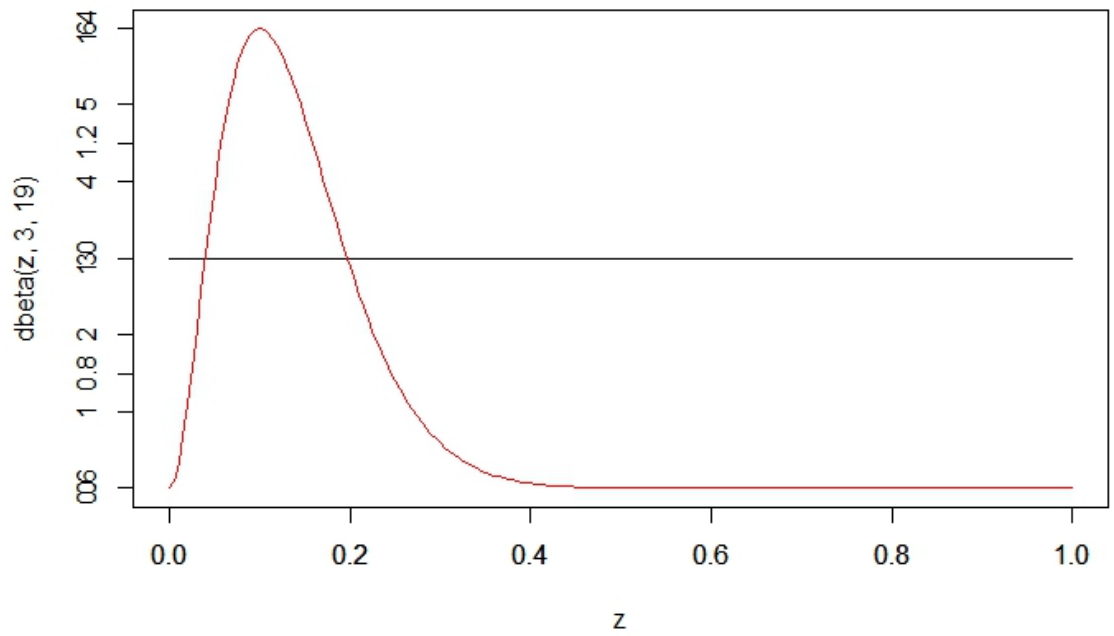

Figure 1: Plot of the prior (in black) and posterior (in red) for the coin example.

From the above plot we can see that observing the data, y, we get a better idea of the region in which the parameter (p) lies. Using the standard $\frac{\text{number of successes}}{\text{number of trials}}$ method we get an estimate for p as 0.1, and it should be noted that the posterior density has a peak around 0.1. We will refer to this coin example throughout the report as a frame of reference.

3 Markov Chain Monte Carlo methods

One of the problems with Bayesian statistics is that for most real-world problems the normalising constant is extremely difficult to compute. However, in the 1980s a new method for estimating the posterior density that bypasses the normalising constant was developed with the introduction of Markov Chain Monte Carlo methods. This method, known as the Metropolis-Hastings algorithm, allows us to sample from the posterior without having to calculating the normalising constant. The idea behind

Metropolis-Hastings is to construct a Markov chain whose stationary distribution is the posterior of interest. We can then take a sample from this chain (after it reaches this stationary distribution) and that sample should be the same as the sample we would get from the posterior.

However, an even more powerful method, approximate Bayesian computation, does not require the calculation of the likelihood, unlike the Metropolis-Hastings algorithm. We will briefly explain the Metropolis-hastings algorithm and then move onto the approximate Bayesian computation algorithm.

3.1 Metropolis-Hastings algorithm

The idea behind Metropolis-Hastings is to start with some initial guess for the parameters θ, generate a new value of θ from a proposal distribution (where ideally the proposal distribution 'mimics' the posterior) and then accept the generated value with a certain probability. The algorithm is defined below:

- Step 1: Initialise (guess) θ^1
- Step $k : [k = 2, ..., n]$
    1. Generate θ^* from the proposal density $h(\theta^*|\theta^{k-1})$
    2. Define $\alpha(\theta^*, \theta^{k-1}) = min\left\{1, \frac{\pi(\theta^*)f(y|\theta^*)h(\theta^{k-1}|\theta^*)}{\pi(\theta^{k-1})f(y|\theta^{k-1})h(\theta^*|\theta^{k-1})}\right\}$
    3. Simulate $U \sim U(0, 1)$
    4. If $U \leq \alpha$ then $\theta^k = \theta^*$

       Else $\theta^k = \theta^{k-1}$

We should end up with a series of dependent estimates for our parameters that are themselves samples from the posterior (assuming we reached the stationary distribution). Notice however that we still need to calculate the likelihood in the Metropolis-Hastings algorithm. We now want to consider an algorithm that doesn't require the likelihood.

3.2 Approximate Bayesian computation

The problem we mentioned earlier is that the likelihood function might be unknown (or at least computationally difficult to evaluate). Approximate Bayesian computation (ABC) is an algorithm that, like the Metropolis-Hastings algorithm, estimates the posterior density by generating values that are samples from the posterior. One of the key differences however is that unlike Metropolis-Hastings, ABC does not require the calculation of the likelihood, which means it can be applied to a wider range of real-world applications.

Like before, we assume to have a prior density and some data y. We then follow the basic steps:

1. Choose a set of summary statistics, S. Calculate $S(y)$.
2. For $k = 1, ..., n$
   (a) Simulate θ^* from the prior
   (b) Use θ^* to simulate some artificial data, x. Calculate $S(x)$.
   (c) If $|S(x) - S(y)| \leq \epsilon$ accept θ^*.
   (d) Else reject θ^* and repeat 2 (a) to 2 (d)

[Majoram et al., 2003]

The best summary statistic function, S, to use is a function that simplifies the data without loosing information. These kind of summary statistics are called sufficient statistics. Conceptually, the idea of sufficiency means that if one person has the data and another has the sufficient statistic, they would both reach the same conclusions when forming credible intervals, posterior density, etc. With regard to the coin example, the number of heads is a sufficient statistic.

Another thing to note is that we also haven't defined what ϵ is. Like summary statistics, we look at the value of ϵ on a case-by-case basis. What we should say however is that if ϵ is large then we will be accepting more θ^* values and if ϵ is small then the values of θ^* that we accept will be a more accurate estimate of θ. The choice of ϵ is therefore a trade off between accuracy and computability.

If we apply ABC to the coin example and then do a histogram of the values θ^* that we kept, we get the plot shown below.

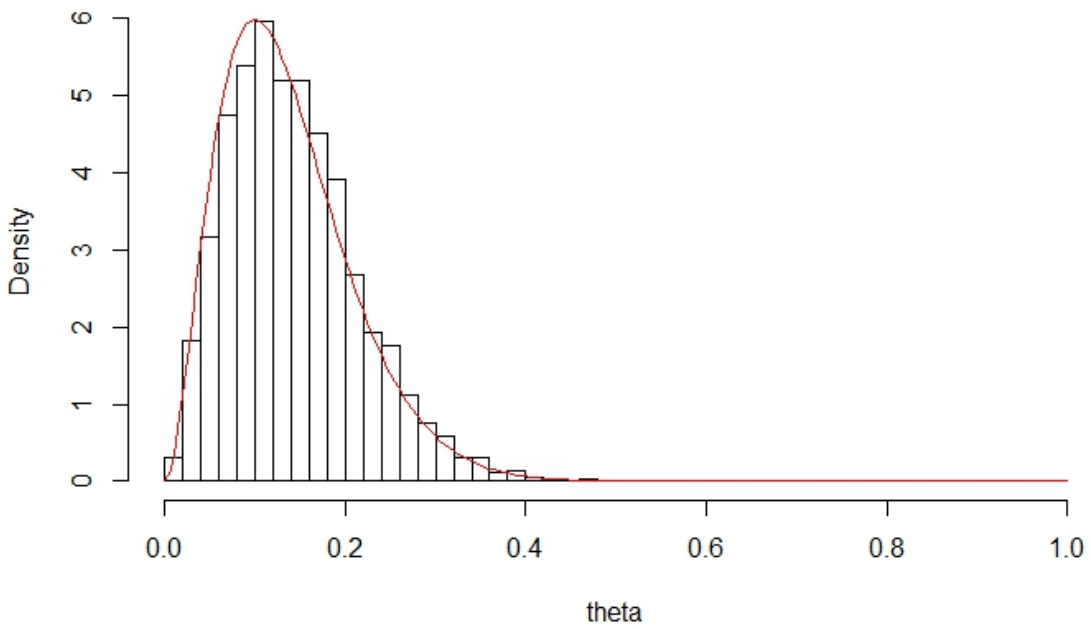

Figure 2: Histogram of theta values retained for the coin example. The red line is the true posterior we calculated earlier.

From the above plot we can see that the results we get from ABC are quite close to the true posterior.

3.3 Kernel Density Estimation

At the end of the ABC algorithm and Metropolis-Hastings we are left with a series of estimates for θ. The natural question to ask then is how can we estimate the posterior from these values? The solution lies in Kernel Density Estimation. [Silverman, 1986]

To estimate the posterior, we place all of the accepted θ^* values on an x-axis and over each of them we draw a density. We can use any density (a square for example would do) so long as the density integrates to 1. We then sum the parts in which the densities overlap and that should give us our estimate of the posterior. One thing we

have to specify is the band width (or variance) of the density. For the coin example we use a Gaussian, or normal, density and found that a bandwidth of 0.02 worked best (although there are functions in R Studio [R Core Team, 2012] that will calculate this band width automatically, such as the 'density' [R Core Team, 2012] function).

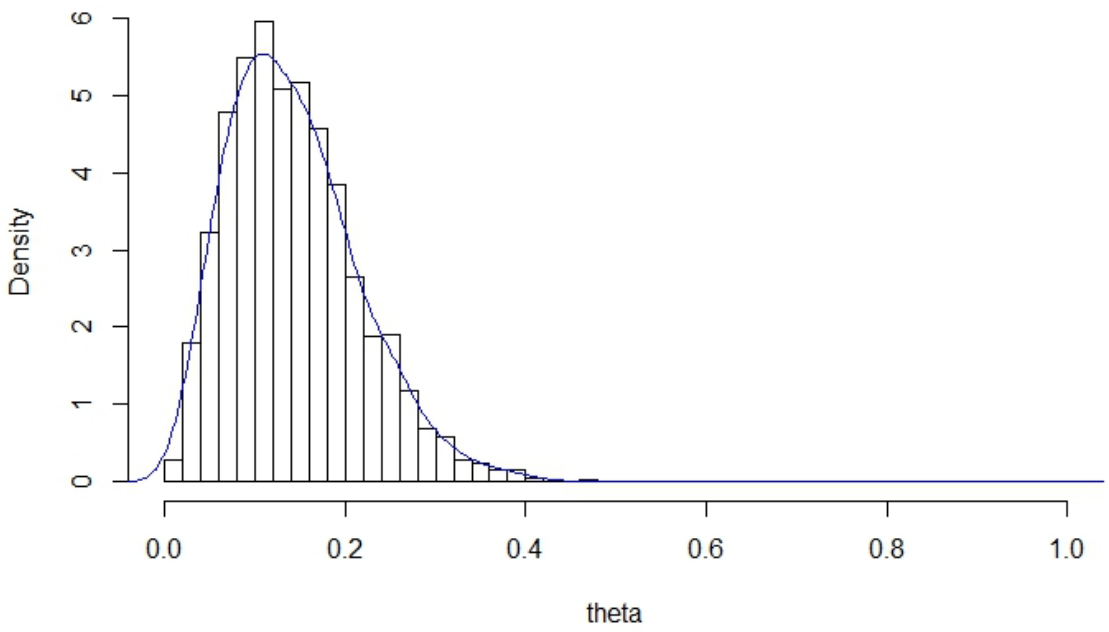

Figure 3: This figure plots a histogram of the theta values we get from applying ABC to the coin example as well as plotting the kernel density estimation with a bandwidth of 0.02.

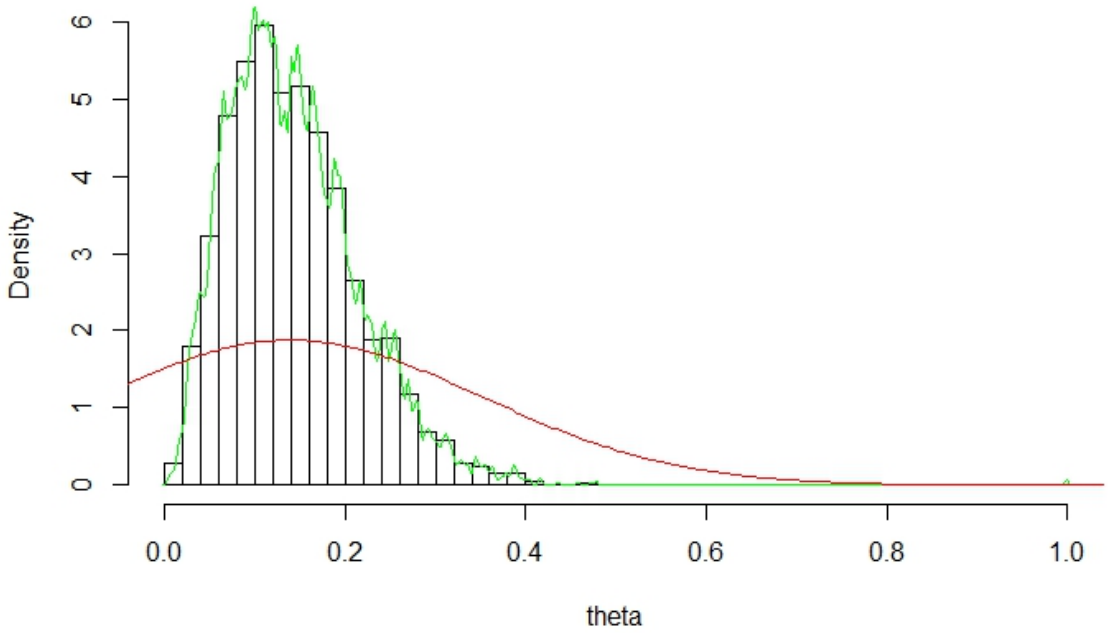

Figure 4: This figure plots a histogram of the theta values we get from applying ABC to the coin example. The red line is the result of kernel density estimation with a bandwidth of 0.002 and the green line has a bandwidth of 0.2.

Figure 3 highlights the accuracy of kernel density estimation with an appropriate bandwidth. Figure 4 however shows us the error in choosing a bandwidth too small or too large. The green line has a bandwidth of 0.002 and is too rough, whereas the red line has a bandwidth of 0.2 and is not an accurate representation of the posterior.

4 Semi-automatic ABC

A new paper by Paul Feanhead and Dennis Prangle ('Constructing summary statistics for approximate Bayesian computation' [Fearnhead and Prangle, 2012]) published last year considers some new methods in ABC. We will focus on their so-called Semi-automatic ABC, which is a method for constructing appropriate summary statistics.

In the paper, they argue that "we wish to choose summary statistics that are equal to posterior means" [Fearnhead and Prangle, 2012], and we can estimate these summary statistics via the following algorithm:

1. Use a pilot run of ABC to determine a region of non-negligible posterior mass
2. Simulate sets of parameter values and data
3. Use these simulated sets of parameter values and data to estimate the summary statistics
4. Run ABC with this choice of summary statistics.

[Fearnhead and Prangle, 2012]

This method works on the logic that summary statistics are functions of the data, and so we are using simulated data to write the summary statistics in a particular form and then use the actual data to estimate the summary statistics.

Step 1 is not necessary if we have a relatively informative prior (a prior that tells us something about the parameters). The reason for Step 1 is simply to reduce the area that we are looking in. For the coin example we could skip this step, as the area we are looking in is relatively restricted (restricted to between 0 and 1). Step 2 is identical to the steps we take in normal ABC, except in Semi-automatic ABC we make no decision about rejecting or accepting the simulated values. Step 3 is the key step in Semi-automatic ABC. The authors suggest that linear regression is the best method. To apply step 3 to the coin example we first show how we represent our data generated in step 2:

p	T_1	T_2	...	T_{20}
.	.	.	...	.
.	.	.	...	.

The above table is a representation of step 2, where the first column contains all the simulated values for p. Column 2, T_1 is representing the 'first toss', where a 1 represents a head and a 0 represents a tail. We continue this for 20 tosses. We then write

$$p = \beta_0 + \beta_1 T_1 + ... + \beta_{20} T_{20}$$

and estimate the β values. Then we substitute $T_1, ..., T_{20}$ for the values we get in the observed data, y and get an estimate for p, which we then use in our normal ABC method.

One of the problems we will see later is that this method requires a very specific and well-defined definition of our 'data'. Now that we have multiple ABC methods, we consider a model we wish to apply them to.

5 SEIR Model

There are a number of different ways in which we can model how an epidemic moves through a population. We will consider the continuous-time SEIR Model. This model splits a population of people into one of four categories; susceptible, exposed, infective, and recovered.

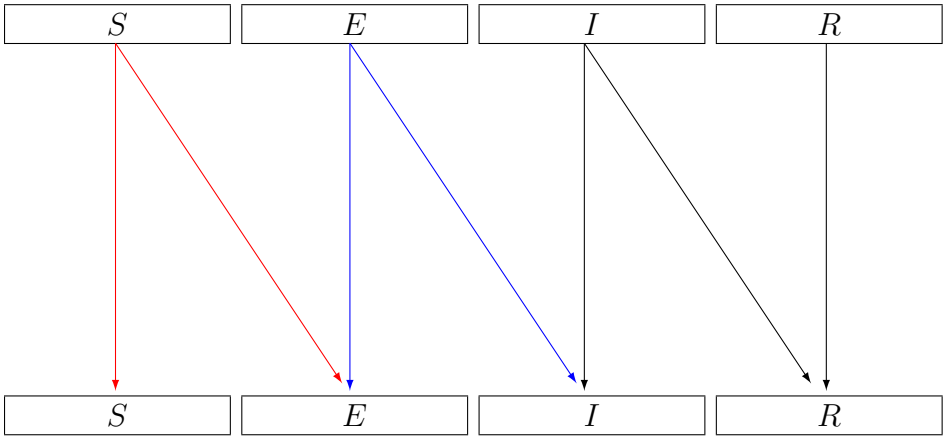

Figure 5: A depiction of how subjects can move through the system

This model is based on the idea that certain infections have 4 stages; first you don't have the disease, then after adequate contact with an infectious person you have been exposed to it (so you will develop the disease, but are not yet contagious), then you are infectious, and then you recover. Figure 5 presents how an individual moves between these 4 categories or groups. We will consider the epidemic to have ended or died-out when there are no individuals in either the exposed or infective group.

The goal of this project is to be able to take some real data and use our ABC algorithm to estimate the rates at which people move from one group to another. Before we can even start to think about applying the ABC algorithm to our SEIR model we first have

to define our parameters of interest.

We mentioned above that this model was a continuous time Markov chain. It is in continuous time because at any given time we consider the state of the system (s, e, i) to be the number of people in the susceptible, exposed and infective group. At any given time, a change to this system may occur, thus it is a continuous time model. Note that we don't record the number of people in the recovered group; this is because that group is relatively uninteresting, and so we consider an individual to be outside of our study once they reach the recovered group. This system is a Markov chain, because we limit the changes to the system (s, e, i) to one of the following:

1. $(s, e, i) \to (s-1, e+1, i)$ [a susceptible becomes exposed] @ rate $\frac{\beta si}{N}$.
   (NB: $N = s + e + i$ at the start of the epidemic)
2. $(s, e, i) \to (s, e-1, i+1)$ [an exposed becomes infectious] @ rate σe.
3. $(s, e, i) \to (s, e, i-1)$ [an infectious recovers] @ rate γi.

So at a random time, any one of these changes can occur, according to the parameters β, σ and γ. The first change has rate $\frac{\beta si}{N}$ because there are two things governing the rate at which a susceptible moves to the exposed group. Firstly, the rate at which a susceptible moves out of the group is proportional to $\frac{s}{N}$. However we need the i in the formula as the more infectives there are then the higher the chance is that a susceptible will become exposed to the disease. Changes 2 and 3 have rate σe and γi for more or less the same reason; because the exposed have to move to the infective group and the infectives have to move to the recovered group before our study is over (i.e. their transition to the next group is a matter of time only). Then the rate at which they move out is simply proportional to how many people are in that group at a given time.

The aim of this project was to apply ABC to the SEIR model in order to estimate these values of β, σ and γ, given a set of data, y. However, in order to apply ABC to the SEIR model we need a way of taking some simulated values for $\theta = (\beta, \sigma, \gamma)$ and generating some data; this requires a clear definition of what we consider data to be in regards to an epidemic.

5.1 Data and simulation for the SEIR Model

For the purpose of this project we consider data to be a matrix in the following form:

s	e	i	0
s_1	e_1	i_1	t_1
.	.	.	.
.	.	.	.
.	.	.	.
s_n	0	0	t_n

The first line of the matrix hold the number of individuals in each group at the start of the epidemic. Then, the rows record each change and the time at which it occurred. We continue this until we get no one in either the exposed or infective group, and finally record the number of susceptibles and the time at which the epidemic ended.

Now we come to how we can simulate data in the above form, given a set of simulated parameters for $\theta = (\beta, \sigma, \gamma)$. To do this, we consider the following:

1. $T_1 \sim exp(\frac{\beta si}{N}) \rightarrow$ random time until next event 1
2. $T_2 \sim exp(\sigma e) \rightarrow$ random time until next event 2
3. $T_3 \sim exp(\gamma i) \rightarrow$ random time until next event 3

Using simulation, we now have a framework with which to generate our artificial data. One possible way to do this is to simulate from T_1, T_2 and T_3. This would give us the next time at which each of the events occur. We could then find the minimum time, work out which event occurred and then update our system. It might seem obvious to then go to the next minimum and update our system again. However, due to the exponential property we would have to re-simulate from T_1, T_2 and T_3 and again find the minimum. However, if we are looking at 20 possible changes over the course of the epidemic, and doing three simulations per change, then we have a total of 60 simulations. This is very computationally inefficient and so we would like a simpler method for calculating which changes occur, and at what time.

5.2 Gillespie algorithm

The Gillespie algorithm is a method that we can use to simulate the time until the next change and then work out which change occurs. It works by using only one simulation to find the time until the next change and then calculates which change occurs.

1. $\min\{T_1, T_2, T_3\} \sim exp(\frac{\beta si}{N} + \sigma e + \gamma i)$
2. Choose the next event to occur
   Event 1 occurs with probability $\frac{\frac{\beta si}{N}}{\frac{\beta si}{N}+\sigma e+\gamma i}$
   Event 2 occurs with probability $\frac{\sigma e}{\frac{\beta si}{N}+\sigma e+\gamma i}$
   Event 3 occurs with probability $\frac{\gamma i}{\frac{\beta si}{N}+\sigma e+\gamma i}$
3. Update (s, e, i) accordingly. Repeat until $e = 0$ and $i = 0$.

[Gillespie, 1977]

The Gillespie algorithm now gives us a framework with which to efficiently generate data, given a simulated value for $\theta = (\beta, \sigma, \gamma)$. Now we discuss some of the difficulties with applying the ABC algorithm to the SEIR model.

5.3 Problems with applying ABC to SEIR model

The only thing we don't have in place yet is an appropriate set of summary statistics. One might think that the best summary statistics to use would be things like, the number of susceptibles at the end of the epidemic, the number of changes that occurred, the length of the epidemic, the peak number of infectives and exposed and the times at which these peaks occurred. Implementing ABC with this choice of summary statistics gives extremely volatile results and as such, we look to other methods for choosing summary statistics.

5.3.1 Applying Semi-automatic ABC to SEIR model

Before, we discussed Semi-automatic ABC however it is not appropriate for the SEIR model. The reason is because we need a clear definition of data in Step 3 of the Semi-automatic algorithm. As it stands, our data in matrix form is not suitable for linear

regression.

One possible solution to define the data as follows:

γ	s	e	i	0	s_1	e_1	i_1	t_1	s_2	e_2	i_2	t_2	...	s_{10}	e_{10}	i_{10}	t_{10}
.	.	.	.	.	.	.	.	.	.	.	.	.	...	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	...	.	.	.	.

So here, we limit the number of transitions to 10. When we run the algorithm and get less than 10 transitions we simply replace the values for s, e and i for N/A. Now we can do the linear regression similar to how we did it in the coin example. The problem we found with this method is that there was too much confounding between the predictor variables, and so we concluded that we cannot apply Semi-automatic ABC to the SEIR model either.

6 Conclusion

Our aim in this project was to lay the foundations for a study into a disease spread on a network of nodes, or people. Within that, we have considered some basic Bayesian statistics and looked at the ABC algorithm which allows us to estimate parameters of a model. We focused on some of the new ABC methods developed by Paul Fearnhead and Denis Prangle, specifically Semi-automatic ABC. We looked at the SEIR model which is the model we used to describe how a disease can move through a population. We finally looked at some of the difficulties in applying Semi-automatic ABC to the SEIR model. Future work would focus on further research into choosing appropriate summary statistics for the SEIR mode and then applying our method to epidemic data to estimate transition rates.

References

Ronald Christensen, Wesley Johnson, Adam Branscum, and Timothy Hanson. *Bayesian Ideas and Data Analysis*. Taylor and Francis Group, New York, 1st edition, 2011.

Paul Fearnhead and Dennis Prangle. Constructing summary statistics for approximate bayesian computation. *Journal of the Royal Statistical Society*, 74(3):419–474, 2012.

Daniel Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemisty*, 81(25):2340–2361, 1977.

Paul Majoram, John Molitor, Vincent Plagnol, and Simon Tavare. Markov chain monte carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States of America*, 100, 2003.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2012. URL `http://www.R-project.org/`. ISBN 3-900051-07-0.

Bernard Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London, 1st edition, 1986.