Using Hawkes Processes to Model and Simulate the Spread of Disease

A. Bhattacharyya, M. Imran, A. Man, M. Zhou, Y. Wang

June 2023

Contents

1 Abstract **4**

2 Introduction **4**

3 Background **6**

3.1 Poisson Process 6

3.1.1 Homogeneous Poisson Process 7

3.1.2 Non-Homogeneous Poisson Process 7

3.2 Hawkes Process 9

3.2.1 Reproduction Number 11

4 Methodologies **12**

4.1 Maximum Likelihood Estimation 12

4.2 Simulation by Thinning Algorithm 13

4.3 Simulation by Branching Algorithm 15

4.4 Our Adjustments 16

4.4.1 Evaluation of the Intensity Function 16

4.4.2 Evaluation of the Integral Term 17

4.4.3 Evaluation of the Log Term 19

4.5 MLE and Simulation Test 19

5 Ebola Data **21**

5.1 Data by Country 22

6 Ebola Simulations **24**

6.1 Model Evaluation 26

6.2 Comparisons between countries 28

6.2.1 Initial stage and explosion 28

6.2.2 Peak stage and international engagement 30

6.2.3 Control and path to Ebola-free 31

7 Interpretation of Ebola Results **33**

8 Covid Data **34**

9 Covid Simulations **36**

9.1 Model Evaluation 37

10 Interpretation of Covid Results **38**

11 Conclusion **39**

12 Appendix **40**

1 Abstract

In this paper, we present a use of Hawkes processes to model infectious diseases. We introduce Hawkes processes as a subset of point processes and a special type of non-homogeneous Poisson process and explain how they can be used to model sequences of events that are dependent on their near history. In particular, we discuss the advantages of using them to understand mechanics of epidemics such as the proportion of spread coming from both internal and external sources, and the R number.

Using data from the Ebola outbreak in Guinea, Liberia, and Sierra Leone, and the Covid outbreak in the UK, we estimated parameters for models using Hawkes processes. With these parameters, we simulate cases using 1) a thinning approach and 2) a cluster-based approach. We apply our model to the data to predict cases over time and compare them to the actual data.

2 Introduction

Millions of people die of diseases every year. Understanding how diseases spread is a vital step in decreasing this number. If we understand the pattern or the behaviour of diseases, we may be able to predict how a disease will spread in future and take appropriate measures to limit the number of people affected. In order to achieve the results, there are many mathematical tools available, such as the SIR model, which are commonly used to model outbreaks. Network-based models often require very good data, in high quantities, and are often computationally very difficult.

We propose an alternative, semi-mechanistic method for disease modelling: Hawkes processes. Hawkes processes are a subset of non-homogeneous Poisson processes, which are a type of point process. Hawkes processes provide several advantages specific to disease modelling. Its ability to model self-exciting processes is representative of how diseases spread, as new cases increase the chance of the disease spreading. It also allows us to easily model the intensity of the disease at any given time, and predict how it will spread in future. Hawkes processes also allow us to easily identify high-risk time periods, as well as enabling us to separate the weightings of exogenous and endogenous sources of spread.

In this paper, we provide a brief introduction to what Hawkes processes are, as a subset of Poisson processes, in order to model discrete events over time. We explain 2 different ways of visualising them, with an event count function, and a branching method to help explain the underlying nature of the process. We then discuss the use of different kernels and the significance of the parameters. After this, we show how to find the parameters using Maximum Likelihood Estimation (MLE), and 2 methods of simulating future events, based on past data. One method uses a thinning algorithm, whereas the other is cluster-based, and we will see the advantages of using both. We also discuss a few changes we make to general methods in order to be disease specific, as well as improving computational efficiency.

Next we apply the aforementioned methodologies to data from the West African Ebola outbreak in Sierra Leone, Liberia and Guinea from the end of 2013 till September 2015. We give plots to help visualise the data and then analyse the spread in different stages of the outbreak using Hawkes processes. Subsequently, we analyse the accuracy of our models and the significance of the parameters they give us.

To finish we conduct a similar analysis on the Covid pandemic, with data from the UK from March to September 2020. We also give visualisations of this data and assess the spread of the disease in phases, and examine the accuracy of our model.

We use Python to implement all of our algorithms on our data, which is available at `https://github.com/FormulaRabbit81/Group_Disease_2.0`. It can be used easily to work on other data. We provide pseudocode for all of our algorithms as well for those unfamiliar with Python, who may wish to write their own implementations in another language.

3 Background

3.1 Poisson Process

Poisson processes are a type of point process. A point process is a random process on a defined mathematical space, in our case the non-negative real line. We will not discuss general point processes further as it is not relevant to us. When the space we are working on is non-negative time, we measure a sequence of stochastic event times t_i, where T_i denotes the realisation of the i^{th} event time. Another way to look at the process is as a counting process $N(t)$, where instead of noting the realisations of event times, we use a function taking integer values, that measure the number of events before the time t. These are equivalent as $N(t)$ is uniquely determined by a sequence of T_i. It can be expressed as the sum of indicator variables:

$$N(t) = \sum_{i \geq 1} \mathbb{1}(T_i \leq t)$$

Definition 1. *[1] A* ***Poisson process*** *with intensity $\lambda(t)$ is the counting process $N(t)$ as defined above, consisting of a sequence of i.i.d exponential random variables $(\tau_i)_{i\geq 1}$ with parameter $\lambda > 0$, and event times $T_n = \sum_{i=1}^{n} \tau_i$. Additionally, the following 3 conditions hold:*
1. N(0) $= 0$*;*
2. N(t) *has independent increments (due to independence of τ_i);*
3. The number of arrivals in any interval of length t > 0 *has Poisson (λt) distribution.*

Some common uses of Poisson processes include:

- The number of earthquakes in a specific area over time
- The number of customers arriving at a shop over a number of hours or days
- The number infected by a disease over time in a given area, which we will explore later

The Poisson process is useful as the model can be used to simulate a random number of points, once we have observed, or are given an intensity function. The τ_i are called inter-arrival times, and $\lambda(t)$ represents the rate at which events occur. For a more rigorous, yet easy-to-follow derivation of Poisson processes as a subset of point processes can be found in [2].

3.1.1 Homogeneous Poisson Process

The Poisson Process is homogeneous when the process has a constant intensity ie. $\lambda(t)$ is a constant λ. A key property of homogeneous Poisson processes is memorylessness, which means, that the distribution of future events is not conditional on past events.

Definition 2. *The* ***Memoryless Property*** *[1] of Poisson Process states that the future state of the system only depends on information from the current event, not events from the past.*

$$P\left(X_{m+1}=j \mid X_m=i, X_{m-1}=i_{m-1}, \cdots, X_0=i_0\right)=P\left(X_{m+1}=j \mid X_m=i\right) \tag{1}$$

3.1.2 Non-Homogeneous Poisson Process

A Poisson process is said to be non-homogeneous when the intensity function $\lambda(t)$ is non-constant. When we wish to model more complex situations, these processes are much more useful. Another way of thinking about the intensity function is as an un-normalised probability density function. Formally, we can relate the intensity function to the counting function of a process by thinking about the probability of an event in a small amount of time as follows [3]:

Definition 3. *[1] Let* $\mathrm{\lambda(t)} : [0,\infty) \mapsto [0,\infty)$ *be an integrable function. Let the process* $\{\mathrm{N(t), t} \geq 0\}$ *to have the property such that, the property is then called a* ***non-homogeneous*** *Poisson process with intensity* $\mathrm{\lambda(t)}$*:*

$$\begin{aligned}
&\mathrm{P(N(t+\delta_t) - N(t) = 0) = 1 - \lambda(t)\delta_t + o(\delta_t)} \\
&\mathrm{P(N(t+\delta_t) - N(t) = 1) = \lambda(t)\delta_t + o(\delta_t)} \\
&\mathrm{P(N(t+\delta_t) - N(t) \geq 2) = o(\delta_t)}
\end{aligned} \tag{2}$$

where the function $\mathrm{o}(\delta_t)$ *satisfies:*

$$\lim_{\delta_t \to 0} \frac{\mathrm{o}(\delta_t)}{\delta_t} = 0 \tag{3}$$

In other words, the probability of observing an event in a small time interval δ_t after τ is $\lambda(\tau)\delta_t$ as $\lim_{\delta_t \to 0}$, and the probability of observing more than 1 event at a time is negligible. Another way to understand $\lambda(t)$ is by considering the expectation of the counting function, $N(t)$. We note that the intensity function $\lambda(t)$ can be interpreted as the rate of change over time of the expectation of the counting function, $N(t)$.

We claim that

$$\int_0^t \lambda(s)\mathrm{ds} = \mathbb{E}[\mathrm{N(t)}]$$

The derivation can be found in **Appendix 1**.

In general, the non-homogeneous case of Poisson process will be a better model for more difficult simulations. This is due to the fact that a variable intensity function has more flexibility to fit different types of data compared to a constant intensity function, as in many cases, the likelihood of events happening will be different at different times.

3.2 Hawkes Process

In the previous section, we discussed Poisson Processes in which events occur independently and are not impacted by the past. This is however not suitable for some processes where the occurrence of events increases the likelihood of new events happening in the near future, so we need to use a special type of non-homogeneous process. These processes are called **self-exciting**. Hawkes processes are one of the most well-known self-exciting point process. Here we have a conditional intensity function $\lambda\left(t \mid \mathcal{H}_t\right)$, so the intensity at any given time is dependent on previously observed events, where we denote the event history as $\mathcal{H}_t$, which contains a list of observed event times T_i. The conditional intensity function $\lambda\left(t \mid \mathcal{H}_t\right)$ is defined by:

$$\lambda\left(t \mid \mathcal{H}_t\right) = \lim_{h \to 0} \frac{\mathbb{P}\left\{N_{t+h} - N_t = 1 \mid \mathcal{H}_t\right\}}{h}$$

We note that a non-homogeneous Poisson process is completely characterised by its conditional intensity function. From now on when we write $\lambda(t)$ or λ, this is shorthand for $\lambda\left(t \mid \mathcal{H}_t\right)$. This is now appropriate for studying the spread of infectious diseases, as we know that infections increase the likelihood of more infections in the area.

Definition 4. *Fix $\lambda_0(t), \phi(t) \geq 0$. A* ***Hawkes process*** *[4] is a point process defined by a conditional intensity function $\lambda\left(t \mid \mathcal{H}_t\right)$ which takes the form:*

$$\lambda\left(t \mid \mathcal{H}_t\right) = \lambda_0(t) + \sum_{i:T_i<t} \phi(t - T_i)$$

given a history of event times $\{T_1, T_2, ..., T_{N(t)}\}$.

The conditional intensity function is comprised of a background intensity λ_0, and a kernel function ϕ. The background can be interpreted as the probability of events being caused by an external source, and the kernel can be interpreted as the increase in probability an event at time T_i has on an event happening at time t. The kernel function is often taken to be monotonically decreasing so that events which occur more recently have a higher impact on the intensity function.

There are many possible choices for our kernel, however, we refine it to a few sensible choices. We mention 3 of the most popular kernel choices; the exponential and the power-law kernel are both monotonically decreasing kernels, whereas the Rayleigh is not. We express the kernels below, fixing $\alpha, \delta \geq 0$, $\eta > 0$ and $\alpha < \eta\delta^{\eta}$:

1. **Exponential** - An exponential decay function, which is monotonically decreasing. It is the most popular choice of kernel for Hawkes processes.

$$\phi(x) = \alpha e^{-\delta x} \tag{4}$$

2. **Power-Law** - Another popular kernel which is widely used, especially in monitoring earthquakes. [5].

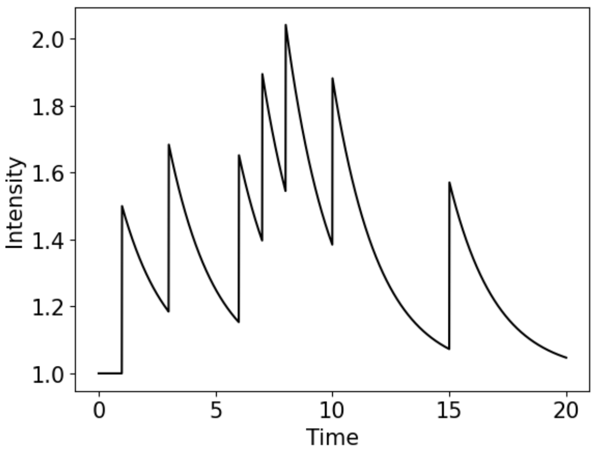
$$\phi(x) = \frac{\alpha}{(x+\delta)^{\eta+1}} \tag{5}$$

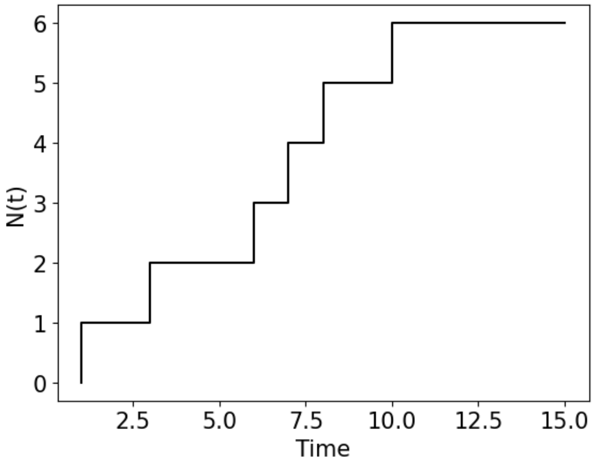
3. **Rayleigh** - A useful kernel to model infections such as malaria, due to the fact that this kernel is not monotonically decreasing and that people with malaria are not immediately at their most infectious. A kernel comprised of multiple Rayleigh kernels can be used for diseases which involve many different stages.

$$\phi(x) = \alpha x e^{-\frac{\delta x^2}{2}} \tag{6}$$

In both the exponential and power-law kernels, δ represents how fast the effect of an event decays over time, and α is a weighting for how much events increase the likelihood of future events.

Below we can see an example Hawkes process's intensity and count function with parameters $\mu = 1$, $\alpha = 0.5$, $\delta = 0.5$ to intuitively see how they are related.





3.2.1 Reproduction Number

The basic reproduction number, R_0 , is arguably the most important attribute of an outbreak epidemiologists look for to assess the severity of the spread of a disease. It represents the average number of secondary infections caused by an infected individual in the population, or roughly how many new people each infected person infects. In general, the outbreak is expected to continue if $R_0 > 1$ [6]. It can be used to calculate the number of people that need to be infected to free the population from a disease. We claim that we can calculate R_0 using the exponential or Rayleigh kernel by: [7]

$$R_0 = \frac{\alpha}{\delta}$$

Proof. Since the kernel function denotes the intensity of the internal spread of the disease, by integrating the kernel function from 0 to infinity we can find out the number of transmissions that an individual has from any time T_i in the entire future [8]. Indeed we have:

$$R_0 = \int_0^\infty \phi(t)dt$$

where $\phi(t)$ denotes the exponential kernel. We then obtain

$$\begin{aligned} R_0 &= \int_0^\infty \alpha e^{-\delta t} dt \\ &= \alpha[-\frac{0}{\delta} + \frac{e^{-\delta \cdot 0}}{\delta}] \\ &= \frac{\alpha}{\delta} \end{aligned} \tag{7}$$

□

Consequently, a result of $R_0 \geq 1$ implies that the spread is out of control since on average each infected individual transmits the disease to more than one other person before recovering or dying. This means the spread of the disease is effectively unbounded.

When we have the case $R_0 < 1$, it indicates that each diseased individual on average transmits the disease to less than one other person. In this scenario, the disease has a higher chance of being contained, and dying out in the population over time even naturally.

As a result, being able to monitor and understand the evolution of R_0 over time remains an important factor to inform public health policies when assessing the impact of a pandemic.

4 Methodologies

We would now like to fit the data with a Hawkes Process by combining a background intensity and a kernel function with parameters that could be found by the maximum likelihood estimation. From now on we choose the background intensity to be constant because of the fact that external spread is only caused by contact with mammals which are not influenced by either seasons or other factors. Moreover, we choose an exponential kernel to do our analysis as it is a very popular kernel in literature [3]. After we get our parameters we can simulate the Ebola virus by different algorithms such as Thinning and Branching.

4.1 Maximum Likelihood Estimation

In order to find the parameters of our intensity function that best fit the data given, we can use Maximum Likelihood Estimation. To do this we have to find the likelihood function for our intensity and then attempt to maximise it. Or equivalently, we will attempt to minimise the negative likelihood using the scipy package in Python.

If $[T_1, T_2, T_3, ..., T_N]$ are the event times of one simulation between [0,T] then our likelihood function is:

$$L(\theta) = \prod_{i=1}^{n} \lambda\left(T_i\right) e^{-\int_0^T \lambda(t) dt}$$

where θ are our parameters.

For the proof, which comes from [9]: **See Appendix 2**.

Since log(x) is a monotonically increasing function, maximising $L(\theta)$ is equivalent to maximising log $L(\theta)$:

$$l(\theta) = \log L(\theta) = -\int_0^T \lambda(t) dt + \sum_{i=1}^{N(T)} \log \lambda\left(T_i\right)$$

where $N(T)$ denotes the total number of events. So now we have to evaluate this function and then maximise it with respect to our parameters θ. In order to increase the efficiency of our code, calculating the integral term by hand is much faster than numerically. For an exponential kernel we get (See appendix 3 for the derivation):

$$\int_0^T \lambda(t) dt = \mu T + \sum_{t_i < T} \frac{\alpha}{\delta}\left(1 - e^{-\delta(T - T_i)}\right)$$

In addition, we can also simplify the summation term in the intensity function by using recursion.

$$\begin{aligned} \lambda(t_i) &= \mu + \sum_{t_j<t_i} \alpha e^{\delta(t_i-t_j)} \\ \lambda(t_{i+1}) &= \mu + \sum_{t_j<t_i} \alpha e^{\delta(t_{i+1}-t_i+t_i-t_j)} + \alpha e^{\delta(t_{i+1}-t_i)} \\ &= \mu + e^{\delta(t_{i+1}-t_i)} \sum_{t_j<t_i} \alpha e^{\delta(t_i-t_j)} + \alpha e^{\delta(t_{i+1}-t_i)} \qquad (8) \\ &= \mu + e^{\delta(t_{i+1}-t_i)} \Bigg(\sum_{t_j<t_i} \alpha e^{\delta(t_i-t_j)} + \alpha \Bigg) \\ &= \mu + e^{\delta(t_{i+1}-t_i)} \Big(\lambda(t_i) - \mu + \alpha \Big) \end{aligned}$$

Therefore, instead of working out the intensity function at each event time, which can be computationally costly, we can use our recursion formula. Once we have our simplified version of the equation, we use the `minimize` or the `differential evolution` function from `scipy.optimize` [10] to minimise the negative Likelihood.

4.2 Simulation by Thinning Algorithm

Simulating the intensity function can be quite complex because of its shape. The Thinning algorithm provides us with a method to get a good approximation of this while not being too algorithmically complex. Firstly, we reduce the problem to simulating a homogeneous Poisson Process with intensity $\lambda(t) \leq \lambda^*$ where λ^* is the maximum intensity in the time frame we are simulating in. In order to improve the efficiency of our code, we can consider a local maximum around the current time instead of over the whole time frame we want to simulate in. This maximum is actually equal to the intensity at the current time because the kernel we use here is an exponential kernel, which is monotonically decreasing. This means at time T, we set $\lambda^* = \lambda(T)$ until the next event occurs.

To simulate this homogeneous Poisson process, we use a method called rejection sampling. The inter-arrival times between events of a Poisson process with intensity λ follows an exponential distribution with PDF:

$$f(t) = \lambda e^{-\lambda t}$$

With CDF:

$$F(t) = 1 - e^{-\lambda t}$$

With the inverse:

$$F^{-1}(u) = \frac{-ln(u)}{\lambda}$$

In order to draw a sample from the distribution, we need the following lemma:

Lemma 1. *(**Probability Integral Transform**) Let $U \sim$ **Unif(0,1)** and $X = F^{-1}(U)$ where F is a strictly increasing CDF. Then X is a random variable with CDF F.*

Therefore, in order to get a sample of the inter-arrival time we get a random sample u between 0 and 1 and then the inter-arrival time, $\tau = \frac{-ln(u)}{\lambda}$. We update our current time T by adding τ. In order to make the simulation of this homogeneous intensity similar to the intensity function, we reject some of the samples. We do this by sampling a random number s between 0 and 1 and accept the sample if $s < \frac{\lambda(T)}{\lambda^*}$. When a sample is accepted, we need to add it to the event times and update the intensity function as there is one more event and add one more term to the self-exciting term of the intensity function. Repeating this until our current time T is greater than the time we wanted to simulate in. Here is the pseudocode for the Thinning Algorithm.

Algorithm 0: Simulation with Thinning Algorithm

```
Input: Intensity function: $\lambda$(t), Event history: $t_i$s, End time: $T_{max}$
Output: Simulated event times: $t_{sim}$s
Procedure Simulate Hawkes Processes
    Set $T$ = Time of final event in history and initialise an empty list of simulated event times;
    while $T \leq T_{max}$ do
        1.1 Set the upper bound of future Poisson intensity $\lambda^*$ to $\lambda(T)$(as $\lambda(T)$ is monotonically
          decreasing);
        1.2 Take a sample $u$ from a Unif$(0, 1)$ distribution and let the inter-arrival time $\tau = -\frac{\log u}{\lambda^*}$
          (by Probability Integral Transform);
        1.3 Update $T = T + \tau$;
        2 Decide to accept the event or not:
        Draw sample $s$ from a Unif$(0, 1)$ distribution;
        if $s \leq \frac{\lambda(T)}{\lambda^*}$ then
            Accept the current event: Add $T$ to the list of simulated times;
        else
            Reject the current event;
        end
    end
    return $t_{sim}$ list
```

4.3 Simulation by Branching Algorithm

Unlike the Thinning algorithm, the Branching algorithm does not use the intensity function directly. It is based on the assumption that except for the base cases, which are generated independently by exogenous factors, all cases are offspring of some other case (and maybe parents) through Homogeneous Poisson Process with constant intensity m, also known as the branching factor, which represents the expected number of direct offspring excited by a single case. Each base case combined with its offspring forms a cluster, and the clusters form the branching structure. By the definition, we can compute the branching factor by integrating the kernel function $\phi(t)$ from 0 to ∞, as ϕ represents the contribution from a single case. In our case with the exponential kernel, we have:

$$m = \int_0^\infty \phi(t)\mathrm{dt} = \int_0^\infty \alpha \cdot e^{-\delta \cdot t}\mathrm{dt} = \frac{\alpha}{\delta}$$

Notice that the branching factor is exactly the same as the reproduction number R_0, which makes sense. We know that if $m < 1$, the number of cases is bounded in each cluster, and eventually every branch will end. If $m \geq 1$, the number of cases in a cluster will tend to infinity.

However, since the Branching Algorithm only uses the reproduction number, some information may be lost which we encountered later in the test section, and we will discuss it there.

To perform the Branching algorithm, we still need to determine the base cases. In order to do so, we can use the Thinning algorithm with intensity function $\mu(t)$. Here, as $\mu(t)$ is constant in the model we are using, we could instead use $\mathbf{Exp}(\frac{1}{\mu})$ to simulate the inter-arrival times for the base cases. For a more general base function μ, we need to adjust the λ^* in the thinning algorithm to $\mu_{max} = \sup_{t<T}\mu(t)$ as the base function might not be monotonic.

For offspring of each base case, we first simulate the number of direct offspring, C, with **Poisson**(m). Then simulate C inter-arrival times with **Exp**(m). Do this for all base cases and combine the inter-arrival times, we get the second generation of cases and apply the same process on it. Repeat this process until all the inter-arrival times given are not in the interval we are simulating. Here is the pseudocode for the Branching Algorithm.

Algorithm 1: Simulation with Branching Algorithm

```
Input: Base function: $\mu$, Kernel function: $\phi$, End time: $T_{max}$
Output: Simulated event times: $t_{sim}$
Procedure Simulate Cluster Structure
    1. Sampling background cases;
    Set $T$, the current time to 0;
    Initialize $G$ as the simulated background cases from $\mu$;
    Add G to $t_{sim}$;
    Let $m = \int_0^\infty \phi(t)$ dt;
    2. Sampling inter-arrival times;
    while $G \neq \emptyset$ do
        Let $t_{ia} = \emptyset$ (ia stands for inter-arrival);
        for $t \in G$ do
            Simulate the number of offspring, $C$, for t with Poisson(m);
            Simulate the inter-arrival times, $t'_1, t'_2, ..., t'_C$, with Exp(m) for C times ;
            Let $t_{ia} = t_{ia} \cup \{t + t'_1, t + t'_2, ..., t + t'_C\}$
        end
        Let $G = \{t \in t_{ia} | \ t \leq T_{max}\}$;
        Let $t_{sim} = t_{sim} \cup G$
    end
    Sort $t_{sim}$;
    return $t_{sim}$
```

4.4 Our Adjustments

One of the primary challenges we have encountered was the computational cost of evaluating the log-likelihood, which can be separated into two components, the integral component and the log component. As:

$$l(\theta) = -\int_0^T \lambda(t) dt + \sum_{i=1}^{N(T)} \log \lambda (T_i)$$

We will discuss the methods we used and the complexity of them in this section.

4.4.1 Evaluation of the Intensity Function

In order to improve the efficiency of the integration, we initially focused on accelerating the computation of the intensity function, which is also frequently invoked in the thinning algorithm. Given that kernel functions are commonly monotonically decreasing, events occurring further in the past have less impact on the self-exciting term, particularly when employing an exponential kernel, which decays with exponential speed. This observation is particularly appropriate in the context of diseases, as individuals with the disease either recover or die after a certain period, meaning they are no longer a vector of transmission.

Our initial design of the intensity function was to include a certain number of events as history. This approach, however, was unsatisfactory as it did not allow for the bounding or estimation of the error. Primarily, this is because the intensity is dependent on the time of the event rather than the preceding number of cases. As a result, this approach could be excessive during periods of low event frequency and too moderate during periods of high event frequency. To rectify this, we revised our $\lambda(t)$ to consider only event times within the $(t-30, t)$ interval, thereby limiting the self-exciting effect to cases from the preceding 30 days. Conveniently, this adjustment also significantly reduces the computational cost of calculating our intensity function

This modification technically means that our point process can no longer be called a Hawkes process as our conditional intensity function is not dependent on the whole history $\mathcal{H}_t$ but a reduced history $\mathcal{H}_t^* = \{T_{N(t-30)+1}, ..., T_{N(t)}\}$.

This approach fits the properties of the diseases we are considering very well. For Ebola, as noted by the WHO, an individual with Ebola can only transmit the disease after the symptoms show up, which is a period from 2 to 21 days after exposure to the virus. [11] Beyond this point patients are either dead or recovered. For Covid, as noted by the WHO, those with severe Covid-19 may remain infectious beyond 10 days and may need to extend isolation for up to 20 days. [12] Consequently, adopting a 30-day window seems to provide an adequate approximation and makes sense for both diseases.

Our plot of the kernel function, with parameters determined via maximum likelihood estimation, reveals that the function decreases significantly prior to the 30-day mark, ensuring the accuracy of the intensity function. Numerical error estimation confirmed this as well. Given $\alpha = \delta = 0.8$ and 33000 cases, the error can be loosely bounded by ϵ_{max}, which is given by:

$$\epsilon_{max} = 33000 \cdot \alpha \cdot e^{-\delta \cdot 30} \approx 1.0 \times 10^{-6}$$

This is considered negligible and affirms the validity of our approach.

4.4.2 Evaluation of the Integral Term

As we mentioned in Chapter 4.1, this particular integration with the exponential kernel can be evaluated precisely by solving the integral algebraically with a complexity of $\mathcal{O}(n)$, which is much faster than evaluating the integral numerically. Before that, we also tried a few approaches to speed up the evaluation of the integral with a more general setting where we don't have the algebraic solution to the integral, but this is very tricky computationally.

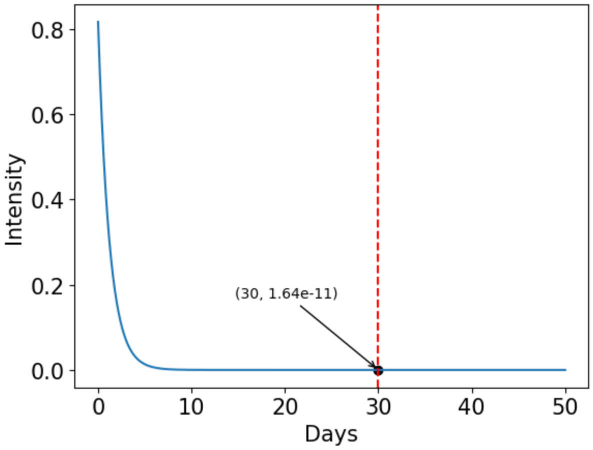


Figure 2: Exponential kernel with $\alpha = 0.817$, $\delta = 0.821$

By Definition 4 (or a plot of the intensity function), it is clear that the intensity function is not continuous, but made of discontinuities and spikes. Hence, calling the `scipy.integrate.quad` package directly on $\lambda(t)$ between 0 and T led to significant errors with a warning message: `IntegrationWarning: The maximum number of subdivisions (50) has been achieved`, which means the integral doesn't converge, and precision is not guaranteed.

The primary method we applied to resolve this problem was to split the interval by event time, evaluate the integral between event times separately, and sum. This way, we avoid discontinuities at each event and get a precise result with significantly less error. However, looping over each interval was extremely slow. Even when only considering the first 300 days of the Guinea data (1000-2000 cases), computing the log-likelihood took over an hour and a half. Aside from that, we still encountered the `IntegrationWarning` sometimes but could not find an explanation. For smaller sets of event times, it would not appear but would after a seemingly arbitrary number of days. This makes this method unstable.

Then we tried to split the interval into chunks each containing some event times, and passing the discontinuities to the 'points' parameter of `scipy.integrate.quad` function. After experiments, we decided to include maximum 50 events in each chunk. Again, the `IntegrationWarning` would appear after a certain number of days, and we could not find the reason, but it returned the correct output without any warning when tested with the intensity function set as the floor function (i.e. kernel function set to constant 1, event times at every integer). This was significantly faster, with each computation of the log-likelihood taking about 45 seconds, and we were able to find parameters to fit over 3000 events in 30 minutes. Even though it is a lot faster, it still has a complexity of $\mathcal{O}(n^2)$. With the algebraic formula, it could be improved to $\mathcal{O}(n)$.

4.4.3 Evaluation of the Log Term

If we evaluate the log term of the log-likelihood directly, the complexity is $\mathcal{O}(n^2)$ since it includes the following double summation:

$$\sum_{i=1}^{N_T} log\lambda(T_i) = \sum_{i=1}^{N_T}(log(\mu + \sum_{T_j<T_i} \alpha e^{-\delta(T_i-T_j)})) \tag{9}$$

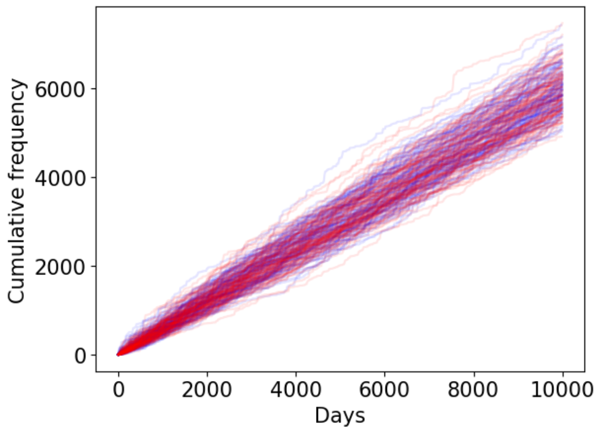
But with the recursive formula provided in chapter 4.1, the complexity could be reduced to $\mathcal{O}(n)$ as we don't need to evaluate the summation in the intensity function from the beginning every time.

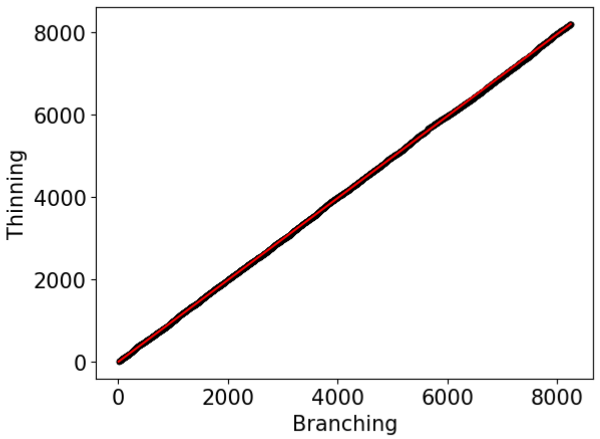
4.5 MLE and Simulation Test

We would like to test our code in order to see how effective and consistent it is before we apply it to our actual data. One method to achieve this is to start with fixed parameters:

$$\mu_{data} = 0.1 \ , \ \alpha_{data} = 0.5 \ , \ \delta_{data} = 0.6$$

Now we can simulate using these parameters and our algorithms above. We perform both the thinning and branching methods 100 times, the red lines represent simulation by thinning and the blue lines represent simulation by Branching.





On the left is our plot of the simulations, we can see the red lines and blue lines mixed together which means they are very similar and have similar ranges too. In order to deduce how different the 2 methods are we performed a QQ plot which is on the right hand side. We get a very straight line which means the distributions are nearly exactly equal.

Despite both methods giving nearly identical results, the branching algorithm is much quicker. 100 simulations using the branching algorithm take 5.6 seconds whereas 100 simulations using thinning algorithm take 93.7 seconds.

Now we will use our MLE algorithm on our simulations and see if we can recover our parameters.
For branching we get:

$$\mu_{branch} = 0.105 \ , \ \alpha_{branch} = 1.016 \ , \ \delta_{branch} = 1.213$$

For thinning we get:

$$\mu_{thin} = 0.097 \ , \ \alpha_{thin} = 0.499 \ , \ \delta_{thin} = 0.601$$

Therefore, we can see the parameters recovered from the thinning algorithm are very close to our actual parameters. The constant parameter for branching is also very accurate, however, our α and δ are out by a factor of 2. But this does not matter as the branching algorithm only considers the R values. In this case, it is $\frac{\alpha}{\delta}$. The R values are:

$$R_{data} = 0.833 \ , \ R_{thin} = 0.829 \ , \ R_{branch} = 0.837$$

Therefore, both methods are quite good for simulating the spread of a disease using Hawkes Processes even though they give different α and δ, which suggests the importance of the R value in the behaviour of the model. Thus, due to the much faster speed of the branching algorithm, we use that for our simulations in the rest of this paper.

5 Ebola Data

Ebola is the term used to describe a group of deadly viruses with the genus Ebolavirus [11] which can have extremely severe symptoms, with death rates varying from 25%-90%, and an average mortality of 50%. The virus was first discovered in 1976 in central Africa, during 2 simultaneous outbreaks in Nzara in what was then Sudan, and villages in the Democratic Republic of Congo near the Ebola River, which is why the virus was given the name Ebola.

Since its discovery, Ebola outbreaks had been rare, small and localized in remote rural areas in Central Africa prior to the 2014-2015 epidemic. The outbreak in 2014 was unprecedented in its scope and duration, as it occurred in urban centres for the first time. The West African countries Guinea, Liberia and Sierra Leone, were uniquely vulnerable to an epidemic due to a number of factors.

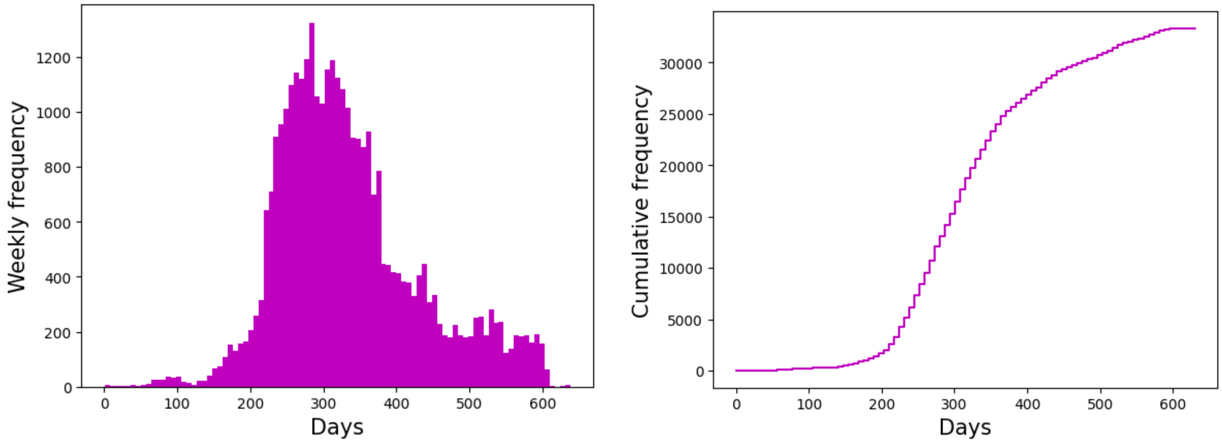
Countries in equatorial Africa have experienced Ebola outbreaks for nearly four decades. Though they also have weak health systems, they knew this disease well. Critically, geography aided containment in those outbreaks. [13]

Ebola enters the human population primarily through contact with an infected mammal, usually bats, but potentially any non-human mammal. After the initial spread from mammals to humans, the virus can then be spread from person to person through contact with an infected's bodily fluids, or through organs including the eyes, nose, and mouth.

Using our model we can calculate important statistics such as the R value, R_0, which tells us whether the disease was out of control or not. Moreover, we can compare the contribution of the background and internal infection sources per country to investigate whether they should have closed their borders.

The data we have comes from three different countries: Guinea, Sierra Leone, and Liberia, with a combination of over 33000 events. The samples were taken from the outbreak in 2014 and 2015. These countries were particularly vulnerable to Ebola because of weak surveillance systems and low standards of public healthcare which resulted in the disease spreading to the cities in July 2014 causing a sharp spike in infections. [14]

Using the data we can initially plot the cases per week and the total number of cases as shown in the figures below (each step is a week):



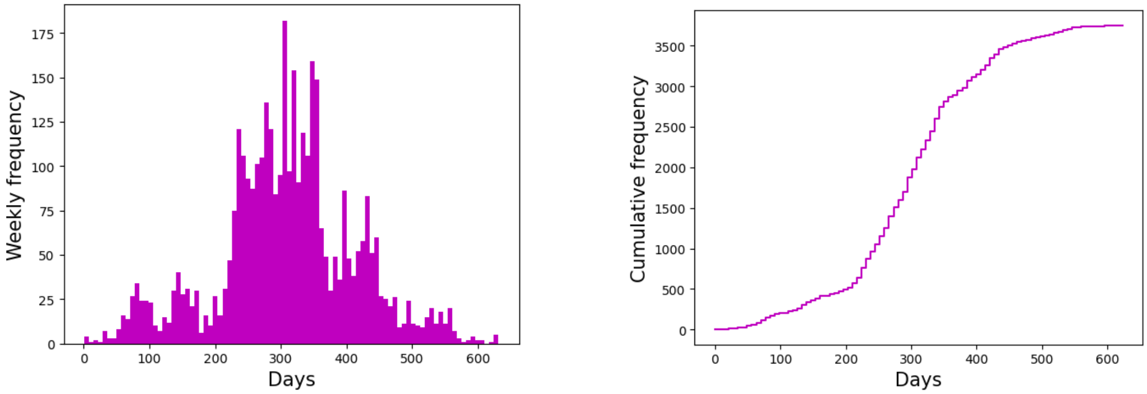
(a) Weekly cases plot

(b) Total cases plot

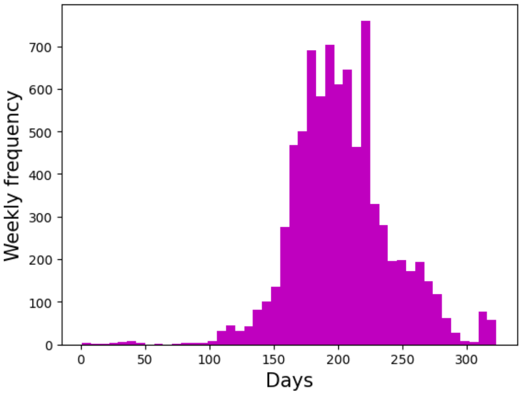
5.1 Data by Country

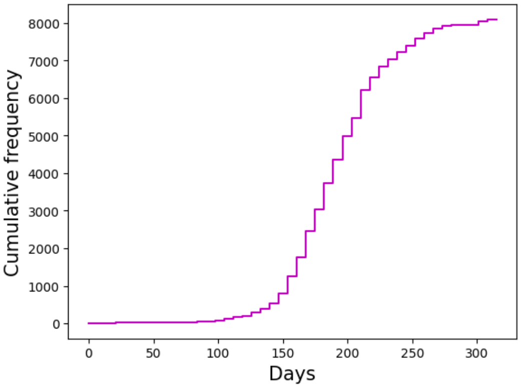
We can also separate the data into countries to highlight any differences. More information regarding our approach can be found via our github page: `https://github.com/FormulaRabbit81/Group_Disease_2.0`.

Guinea:

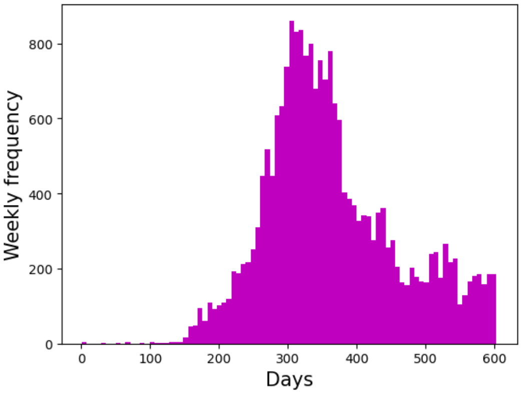


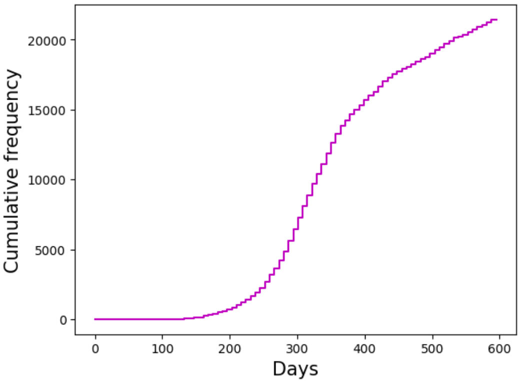
Liberia:





Sierra Leone





The data looks very similar for every country, however Liberia was declared Ebola-free in May 2015 whereas Sierra Leone wasn't till March 2016 and Guinea till June 2016. [14] It is also worth noting that most of the cases occurred in Sierra Leone, and very few were in Guinea, even though it has the largest population among the three countries.

6 Ebola Simulations

Now with the parameters obtained from the MLE algorithm discussed in the methodology section, we can simulate the spread after a certain time and see if it matches our data well. We chose to start our simulations from roughly every quarter of the outbreak to try and understand how the spread of Ebola changes over time.

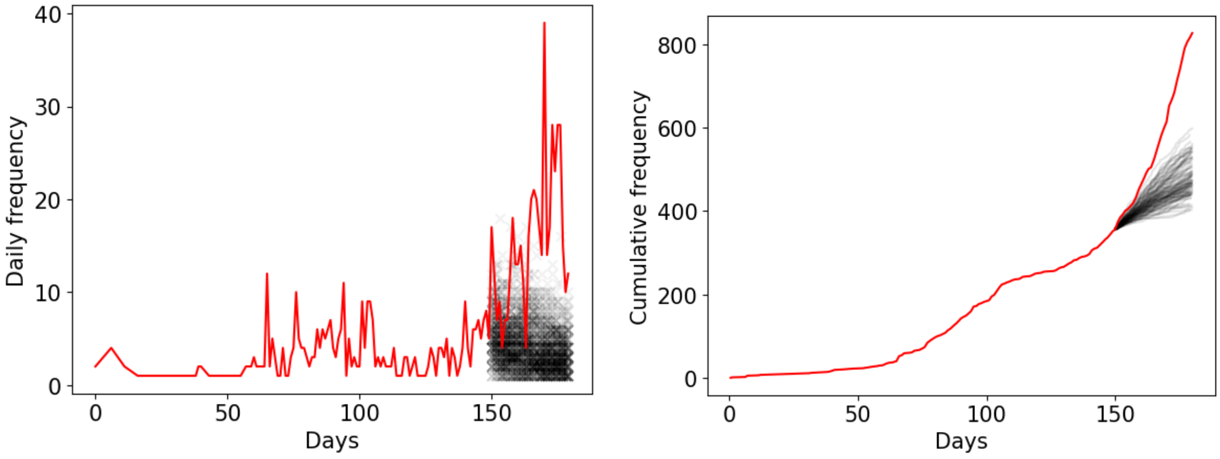
Here we choose to combine the countries when predicting. This region is highly interconnected with relatively easy movement between villages and cities, including the capitals, and which has substantial cross-border traffic in an area where borders barely exist and where people identify more with the region rather than particular nation states.[15] Moreover, the traditional custom of returning to a native village to die and be buried near their ancestors magnified population movement across borders further.

The red lines represent the actual data while the black lines and crosses are 100 simulations using the Branching algorithm. We conduct predictive testing on an 80:20 ratio of data used to train our model to data used to test it. All the parameters are rounded to 3 decimal places.

150 Days:

Performing MLE on the first 150 days of data gives us the parameters:

$$\hat{\mu} = 0.347 \ , \ \hat{\alpha} = 0.258 \ , \ \hat{\delta} = 0.302$$



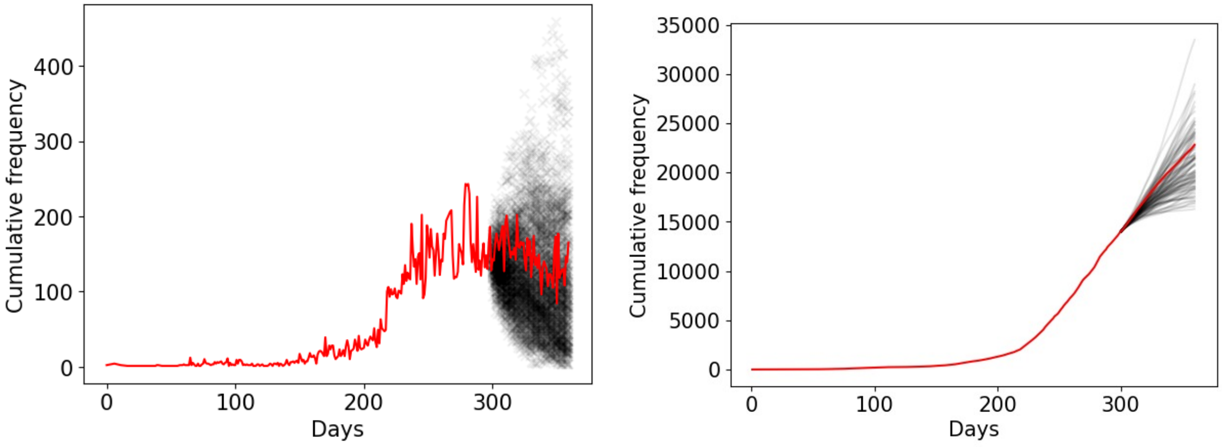
(a) Daily cases prediction

(b) Total cases prediction

300 Days:

Performing MLE on the first 300 days of data gives us the parameters:

$$\hat{\mu} = 0.366\ ,\ \hat{\alpha} = 1.004,\ \hat{\delta} = 1.012$$



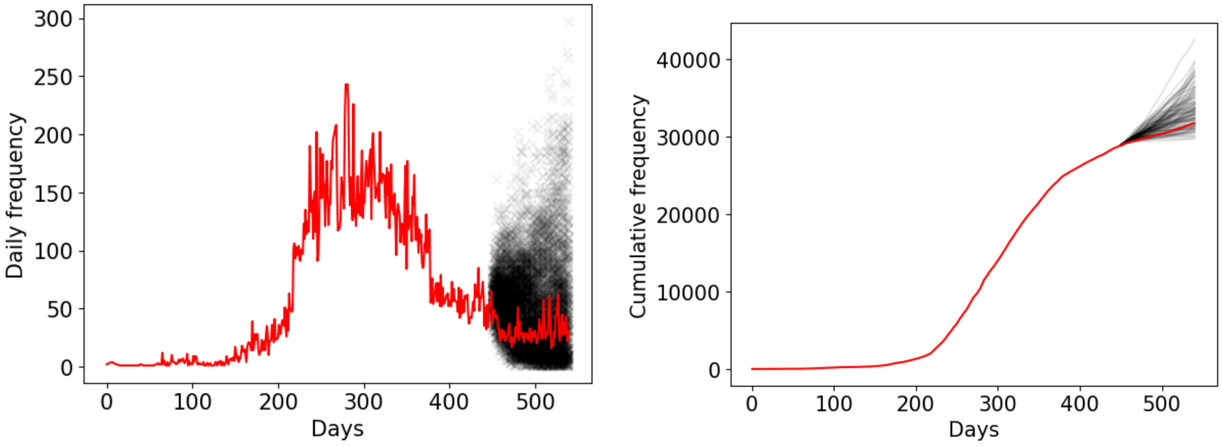
(a) Daily cases prediction

(b) Total cases prediction

450 Days:

Performing MLE on the first 450 days of data gives us the parameters:

$$\hat{\mu} = 0.351\ ,\ \hat{\alpha} = 0.880\ ,\ \hat{\delta} = 0.884$$



(a) Daily cases prediction

(b) Total cases prediction

6.1 Model Evaluation

In order to check the validity of our predictions, we chose to use box-and-whisker plots. It is a graphical representation of the distribution of a data set. With it, we can see the central tendencies of our simulations and how well the simulations fit the actual data. 3 different box plots are created at the stage of 150 days, 300 days and 450 days. The red dots represent the real data, the boxes represent the inter-quartile range and the remaining black dots are the outliers. We choose to aggregate the data on a weekly basis to make the plots clearer while avoiding the effect of day to day perturbations in behaviour.

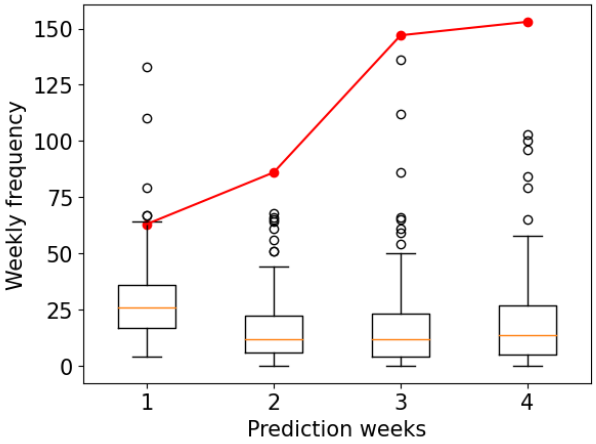
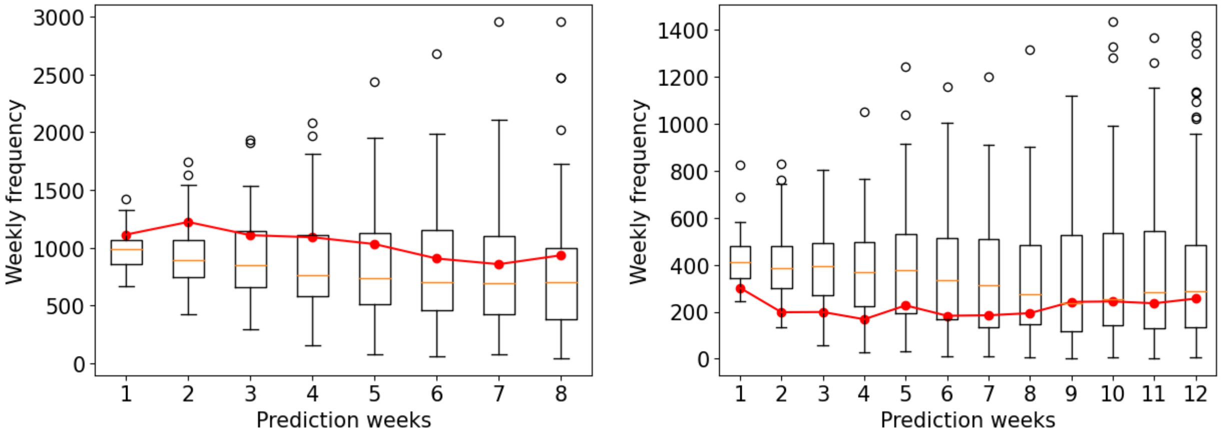


Figure 11: Box Plot at 150 days



(a) Box Plot at 300 days

(b) Box Plot at 450 days

150 Days: From the plot, it is shown that the actual data lies in the range of outliers, which implies that the simulations failed to predict the future based on the data until 150 days. One reason for it could be the fact that the 150 days plots include the least number of cases, which gives the highest variance for the MLE, due to the dependence of the Fischer Information on the amount of data points. We should also observe that relatively speaking, the difference in cases in this period is not much compared to the whole data. The inaccuracy is more likely, however, to be caused by some factor our model has not considered, which comes into effect around the 150 days mark. It might be caused by a small outbreak in one of the countries which we are going to discuss later.

300 Days: From the box plot for 300 days, it is clear that most of the actual data lies along the top of each box except for the prediction of the first 2 weeks. Hence, there is evidence to suggest that our model is suitable and produces accurate results for 300 days, even if it tends to be a slight overestimate of the actual data, and we will discuss it later.

450 Days: From the box plot of 450 days, more than half of the actual data lies within the bottom of the boxes, and half of the actual data lies just below the boxes for the first few prediction weeks. Overall, the result of the box plot provides a positive response and we conclude that the model remains suitable for 450 days simulation.

Overall, we used a Hawkes process to generate predictions of future cases starting from a specific point. We conclude that even though the simulation for 150 days seems to be inaccurate, we managed to produce sensible simulations for both 300 days and 450 days simulation. We are therefore able to claim that the real data mostly agrees with the predicted data and the prediction model holds.

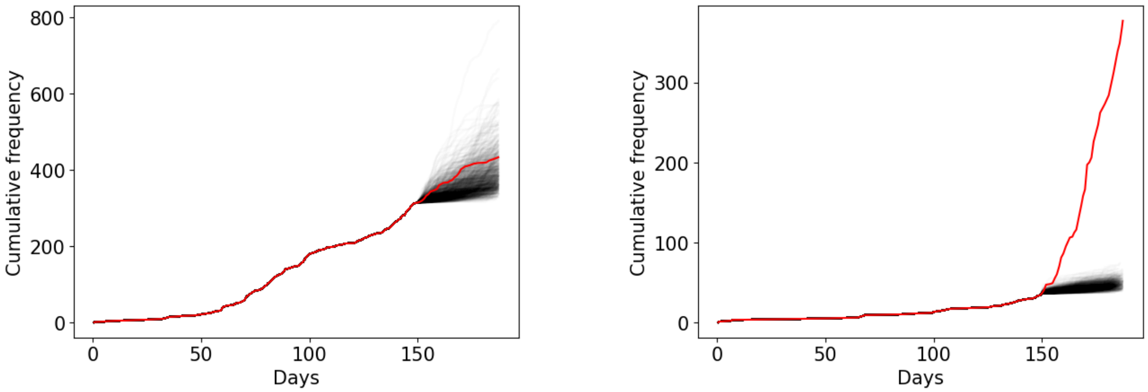
6.2 Comparisons between countries

Liberia, Guinea, and Sierra Leone were all severely affected by the Ebola virus outbreak. We have looked at the aggregated data, but we now would like to focus on the individual countries and compare them. As with the aggregated data, we will do simulations at different stages of the epidemic. We aim to use the real-life context to explain the actual data and evaluate our simulations to understand when they are most effective and if they are limited, what the model lacks.

Firstly, we notice that Sierra Leone had by far the most cases, accounting for almost two-thirds of the total. Checking the daily frequency plots shows us that the daily frequency peaked at about 150 in Liberia and Sierra Leone, versus 40 in Guinea. We would then expect that α would be smaller in Guinea, while it would be similar in Sierra Leone and Liberia.

One factor that could have contributed to lower case numbers in Guinea was their early detection and response to the outbreak. They reported the first cases in December 2013, while Sierra Leone and Liberia initially struggled to identify and respond promptly. We will first look at the early stages of the outbreak.

6.2.1 Initial stage and explosion



(a) $\hat{\mu} = 0.335$, $\hat{\alpha} = 0.241$, $\hat{\delta} = 0.287$

(b) $\hat{\mu} = 0.227$, $\hat{\alpha} = 0.001$, $\hat{\delta} = 0.006$

Figure 13: 1000 simulations - Guinea on the left and Liberia on the right.

We start the analysis of Liberia after the first case in the country, which is roughly 2 months later than the other 2 countries. It is clear that the optimised parameters obtained from Liberia data from the first 150 days meant that the simulations would not capture the explosion in cases at around 150 days. Since α is effectively zero, there is no self exciting contribution to the intensity and so all the cases are driven by importations from neighbouring countries.

We offer a few possible explanations for the large discrepancy between the simulated and actual cases. In contrast to Guinea, poor governance and slower response in Sierra Leone and Liberia meant that effective testing may not have been in place yet, and the actual numbers of Ebola cases was higher than was reported during the first 150 days.

Testing was initially slow in all three countries because there were no existing facilities to adequately diagnose EVD[16]. The popular sentiment was that a 'mystery' disease characterized by fever, severe diarrhea, and vomiting was rampant. It was variously misdiagnosed to be malaria, typhoid fever, Lassa fever, cholera and so on, as many of the symptoms are shared. Poor testing has a positive feedback loop: fewer reported cases lead people to underestimate the severity of the outbreak, leading to conservative or insufficient intervention, leading to more undetected cases. Any public health messaging, policy changes and interventions have time lags, whilst the virus continues to spread.

A major factor in the evolution of the outbreak was health infrastructure and resources. Relative to Liberia and Sierra Leone, Guinea's public health system was more developed, able to prevent the virus from exploding in major urban areas with the early availability of Ebola treatment units (ETUs) and by limiting within-hospital transmission through isolation of cases, at the beginning of the epidemic. [17] Sierra Leone and Liberia both faced devastating civil wars that lasted until 2002 and 2003 respectively, leaving basic health infrastructure severely damaged or destroyed.

Additionally, it is likely that the exponential growth in cases at the 150 day mark (around July 2014) coincides with the virus spreading to the cities and the capital. At this stage, they had become too numerous to be traced. The prior three months of linear growth in cases may have given a false sense of security and the West African countries were poorly prepared for the unfamiliar disease at every level.[13]

6.2.2 Peak stage and international engagement

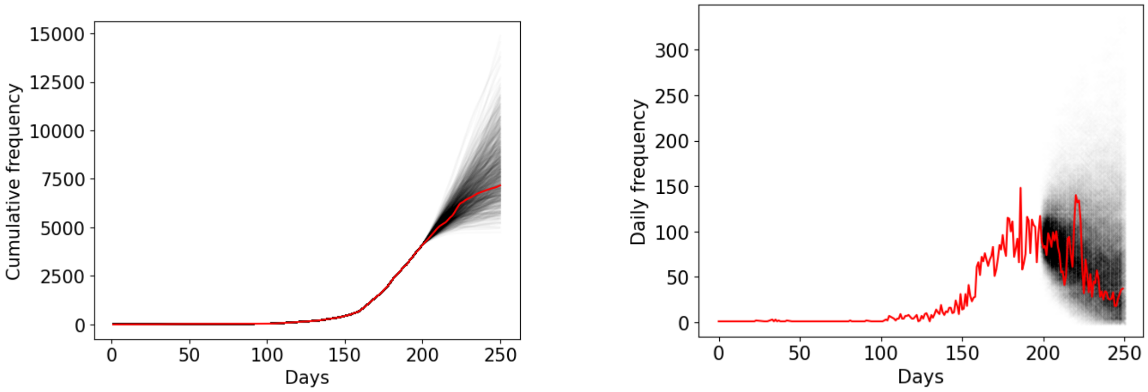


Figure 14: 1000 simulations - Liberia. $\hat{\mu} = 0.209$, $\hat{\alpha} = 0.682$, $\hat{\delta} = 0.689$

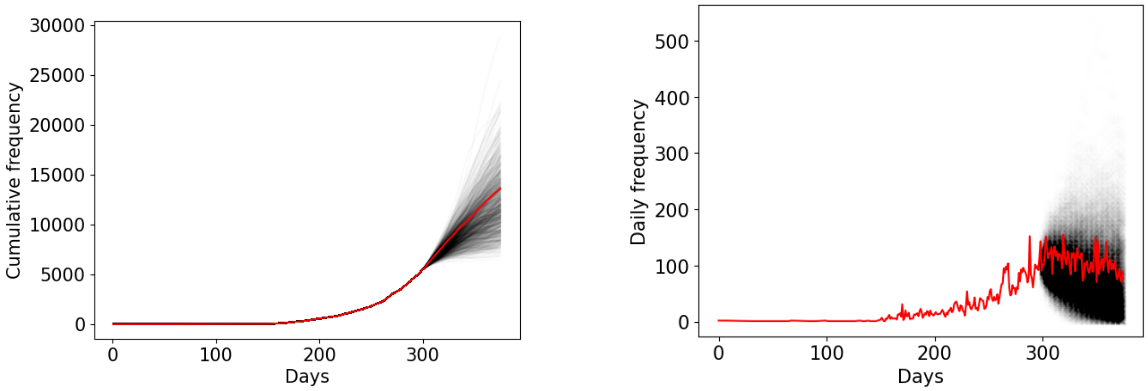


Figure 15: 1000 simulations - Sierra Leone. $\hat{\mu} = 0.121$, $\hat{\alpha} = 0.843$, $\hat{\delta} = 0.849$

We can see that the peak stage of the epidemic is where our Hawkes process model is most suited. The self-exciting contribution to the intensity function is appropriate as the occurrence of historical cases increases the rate of new cases.

This self-exciting process in the theory represents real factors that exacerbated the spread of Ebola.

- Intrahospital transmission: As patients are admitted to hospital, the risk of health care workers and others in hospital being infected increases.

- Cultural beliefs: Ancestral funeral and burial rites involved direct contact with the deceased, whom remain infectious after death. In August 2014, Guinea's Ministry of Health reported 60% of cases could be linked to these practices, while in November 2014, WHO staff in Sierra Leone estimated an even higher 80%.[13]

- Traditional medicine: Many surges in cases were traced to contact with a traditional healer or herbalist or attendance at their funerals.

By definition, our model does not include any negative terms. This means that our simulations will not naturally slow down. We can see in the Liberia cumulative plot that the cases start to level off, while many of the simulations continue to grow. However, the international response was a factor in managing the epidemic.

In August 2014, the WHO declared the outbreak an international public health emergency and the UN Security Council held its first ever emergency meeting on a public health crisis in September. [18] This led to a scaling up of the international response.

6.2.3 Control and path to Ebola-free

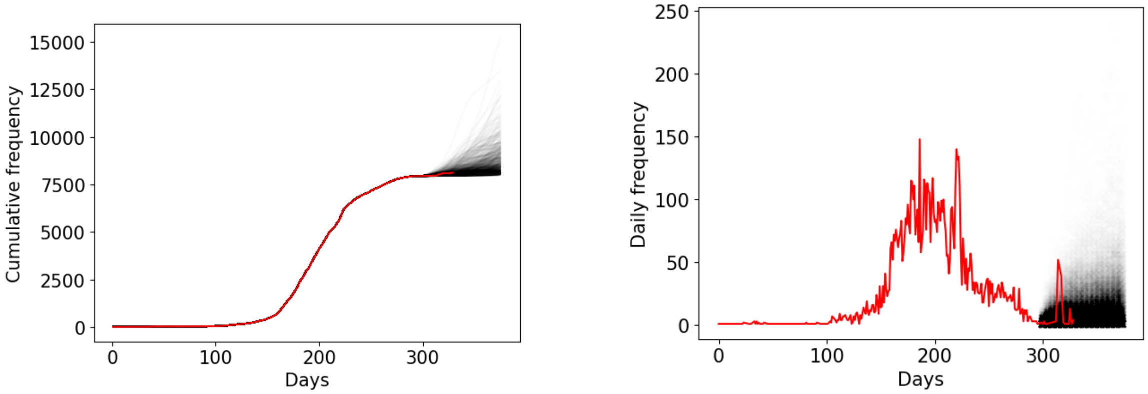


Figure 16: 1000 simulations - Liberia. $\hat{\mu} = 0.225$, $\hat{\alpha} = 0.956$, $\hat{\delta} = 0.964$

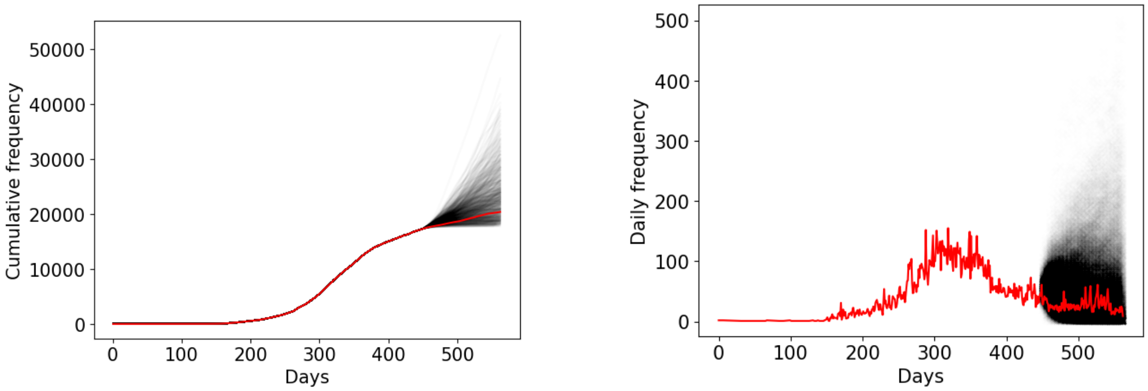


Figure 17: 1000 simulations - Sierra Leone. $\hat{\mu} = 0.116$, $\hat{\alpha} = 0.726$, $\hat{\delta} = 0.728$

We found that the data from Liberia ended almost a year earlier than the other two countries. While in 2014 Liberia was the hardest hit, it was formally declared to be Ebola-free on 9th May 2015, significantly earlier than March 2016 and June 2016 in Sierra Leone and Guinea respectively.

Looking at our Liberia simulations from 300 days, we can see the desired behaviour, a levelling off in cumulative cases, and a daily frequency decaying to zero. The growth in cases stems from the importations $\hat{\mu}$, but with effective contact tracing and isolation, these cases will not have any descendants.

Our simulations generally overestimate the cases at this point of the outbreak and this is more clear with the Sierra Leone data, as we obtain the parameters fitted to 450 days. Our model does not account for the lessons learnt by communities and government or the aid sent by the international community but in reality, the post-peak period signalled effective interruption of transmission. Liberia's control of Ebola was helped by a number of factors: 1) government leadership and sense of urgency, 2) coordinated international assistance, 3) sound technical work, 4) flexibility guided by epidemiologic data, 5) transparency and communication, and 6) efforts by communities themselves. [16]

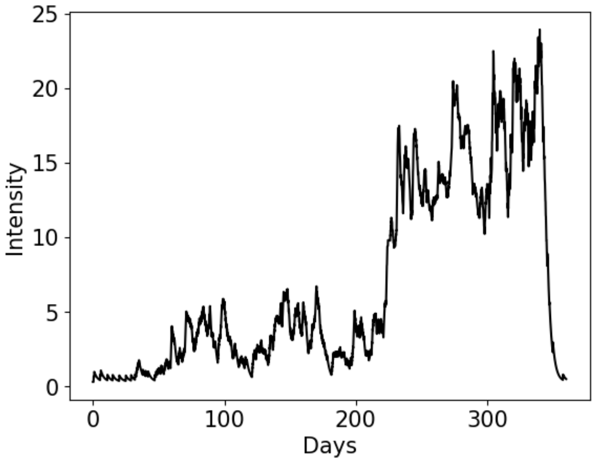
7 Interpretation of Ebola Results

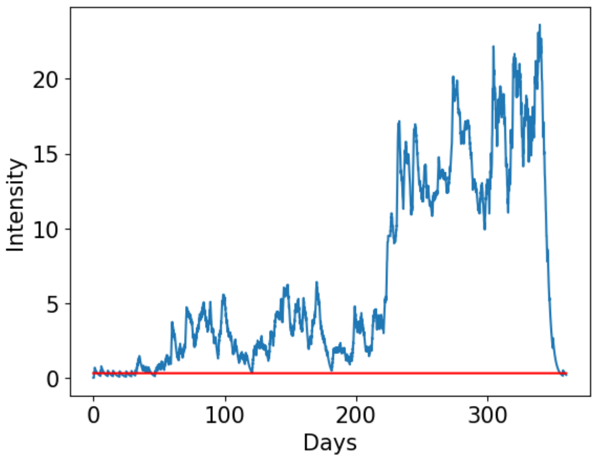
What do the parameters we used for our simulation mean? Using α and δ we can compute the R number:

$$R_{150} = 0.854 \ , \ R_{300} = 0.992 \ , \ R_{450} = 0.995$$

We can see that the reproduction numbers for all three stages are all less than 1 which means that the spread of Ebola was controllable. However, as time went on, the value for R increased and was getting closer to 1 and becoming uncontrollable.

We can also check the contribution of the background or kernel to our intensity function. For the plots below, the parameters from 300 days in Guinea is chosen:





On the left is a plot of our intensity function using the event times from one simulation. On the right is the intensity function split up into kernel contribution (blue) and background contribution(red). Here we can see the background is only meaningful around the first 50 days and after that it is minuscule. This is a very useful result as it tells us that closing borders or preventing new immigrant cases will have a small impact and other measures such as preventing contact between people would be better.

Due to the fact that the trend of Ebola changes significantly from nearly 200 days, we have also tried to evaluate the MLE parameters with the data after 200 days from the beginning. By doing so, the parameters we obtained gives a $R = 0.9999$ and $\mu = 0.005$. It is suggesting a Hawkes Process without a base. Even though the simulation based on these parameters is worse, but the idea of changing parameters provides a better understanding of the analysis of Covid.

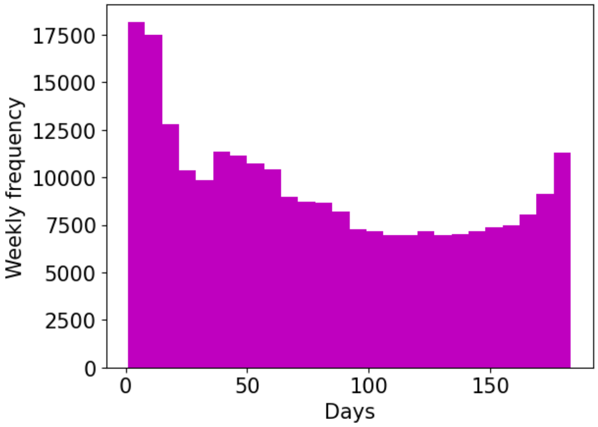
8 Covid Data

Covid-19, which is short for the Coronavirus Disease Epidemic 2019, is a highly dangerous illness that was initially caused by the coronavirus SARS-CoV-2. The first case was found in China in December 2019 and the epidemic still remained a global challenge until today from all aspects. The physical and mental health issues that individuals encounter, and the social disruptions including lockdowns of people, businesses, and schools, were all significant impacts caused by Covid-19. Since the first case that happened in December 2019, there are now more than 760 million confirmed cases and around 7 million deaths.[19]

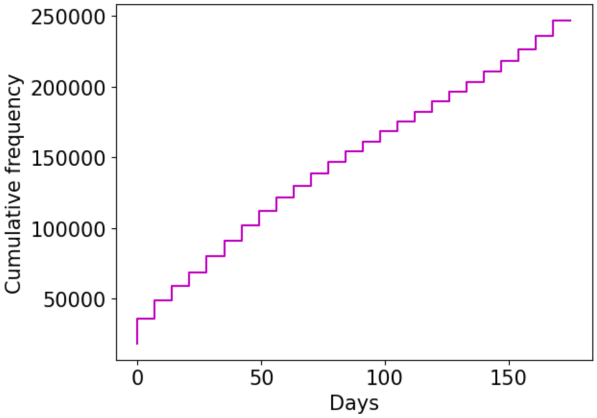
The spread of Covid-19 is likely to be primarily transmitted to humans through non-human mammals such as bats. The virus can then be spread from person to person in several different ways. According to [20], the virus is mainly transmitted through close contact between people such as coughing, sneezing, and simply by breathing, and can usually be spread with a conversational distance. In addition, the virus is also easily transmitted in crowded and non-ventilated indoor places.

Given the destruction of Covid-19 to the community and the uncertainty feature of transmissions, it is important to gather data and characteristics from confirmed cases to understand and hence monitor the spread. By using similar methods for simulating and predicting Ebola, we may be able to provide statistics that help one understand Covid-19 more thoroughly.

The data we have, (publicly available at [21]) work with consists of daily cases from different parts of the UK from March 18^{th} to September 2020 20^{th}. It has a combination of over 250k events and the period went through phases including outbreaks and lockdowns.



(a) Weekly cases



(b) Total cases

Above we can see the weekly and total cases of Covid in around a 200-day span. At the start, we can see a decrease in weekly cases. This is explained by the introduction of a lockdown on March 16^{th} [22] which reduced close contact between people and hence reduced the spread. However, after 100 days the trend flips and the cases increase which is due to lockdown restrictions being eased on July 4^{th} [22].

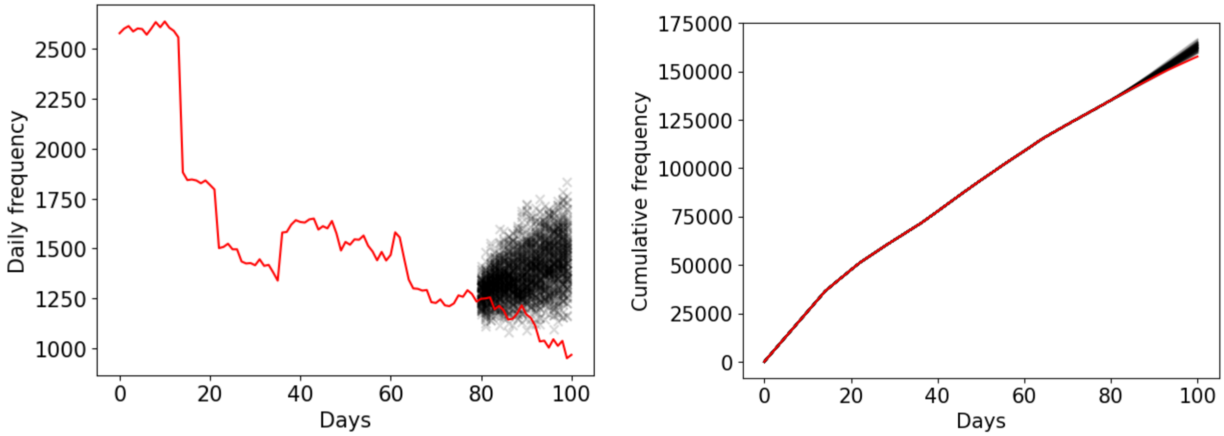
In order to model and simulate this, we will split the data at 100 days and try to model the spread with 2 separate Hawkes processes.

9 Covid Simulations

0 – 100 Days:

The parameters after doing Maximum Likelihood Estimation on the first 80 days are:

$$\hat{\mu} = 63.008 \ , \ \hat{\alpha} = 5.983, \ \hat{\delta} = 6.219$$



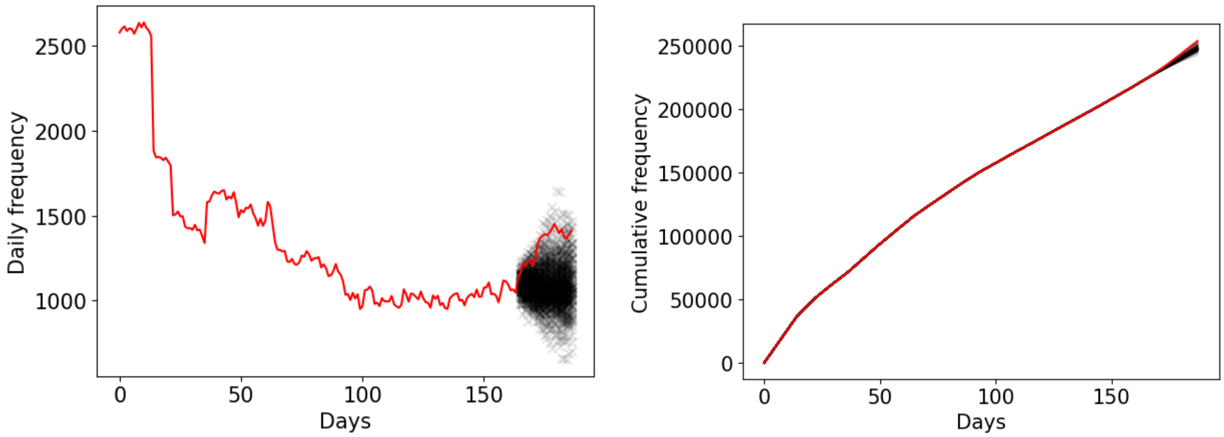
(a) Daily cases prediction

(b) Total cases prediction

100 – 185 Days:

The parameters after doing Maximum Likelihood Estimation on the data between 100 and 165 days are:

$$\hat{\mu} = 0.009 \ , \ \hat{\alpha} = 4.559 \ , \ \hat{\delta} = 4.561$$



(a) Daily cases prediction

(b) Total cases prediction

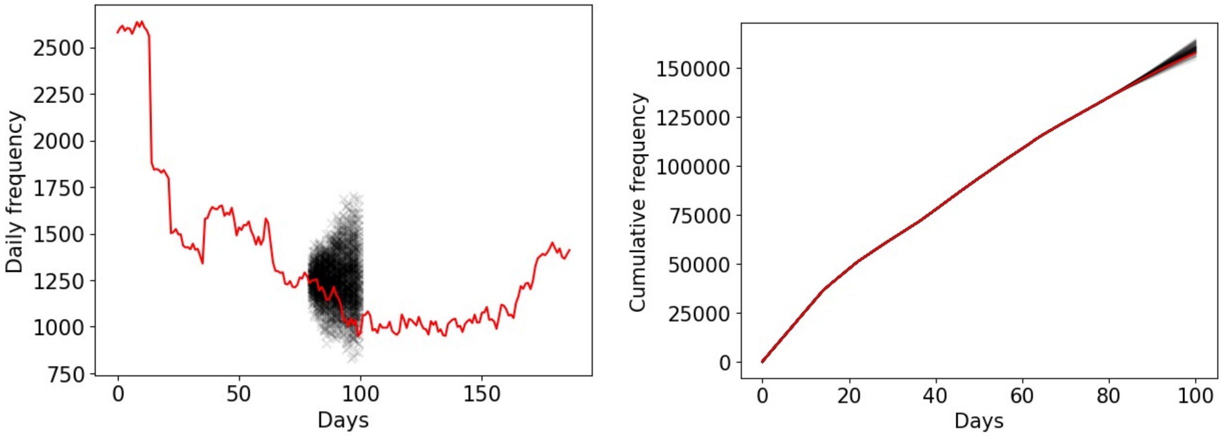
9.1 Model Evaluation

We can see our simulation for the first half of the data is not very accurate which could be due to the high number of cases for the first 10 days. Hawkes processes are not good at modelling data we do not have access to the whole history starting from the first case. We will try to repeat this process but using the data for 20 to 80 days and then do a prediction.

20 – 100 Days:

The parameters obtained by Maximum Likelihood Estimation on the data between 20 and 80 days are:

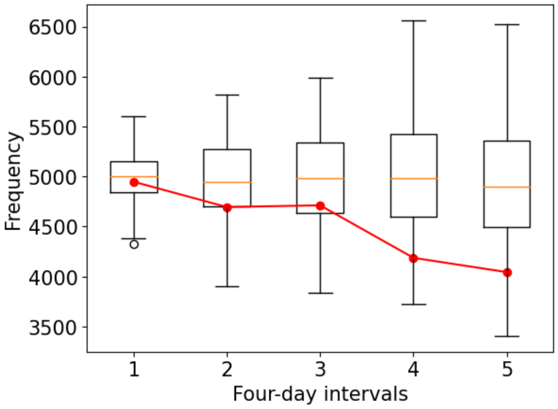
$$\hat{\mu} = 0.052 \ , \ \hat{\alpha} = 6.311 \ , \ \hat{\delta} = 6.311$$



(a) Daily cases prediction

(b) Total cases prediction

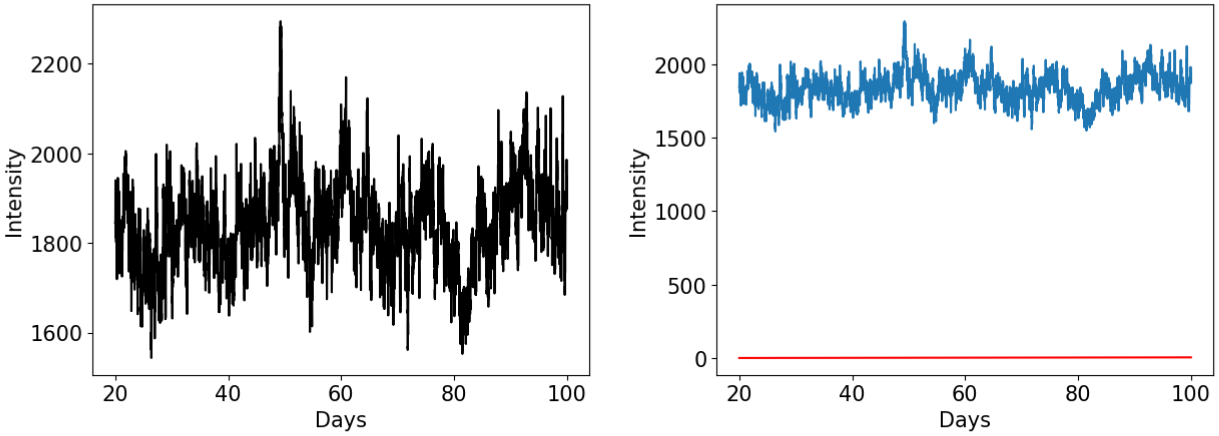
After simulating a further 20 days, we can see our simulations are now spread around the data more than before (the data is covered by the simulations) which is a good sign that a Hawkes process can be used to fit Covid. This is also shown by the boxplot below.



10 Interpretation of Covid Results

Firstly, the most notable parameter is the constant term which represents the background rate or immigrant of infection. For 0 to 100 days we can see that there is a background rate as high as 63. However, after removing the first 20 days, the μ value drops down to 0.0052. It is worth noting that the data starts from 18/03/2020, while the first lockdown in the UK starts from 23/03/2020. [22] This suggests that the first few days have a different background than the later days. Mixing them together would lead to large errors since they should not be described with the same model. We can see that the prediction with the parameters evaluated by the data from 20 to 80 days is much better. Also, fitting the data from 100 to 185 days gives a background rate of 0.009. These show the effectiveness of a lockdown.

With incomplete data at the beginning, we cannot fit or analyze the Hawkes Processes before the lockdown. However, the blue line on the graph below shows us that the bulk of infections is still caused by contact between UK residents. This explains why the UK government decided to close the borders and introduce a lockdown. As a result, in the second phase, we can see how the background intensity has diminished which means the main spread of covid is caused by contact within the UK population. This is also solved by a lockdown but as restrictions were eased we can see the intensity rises again.



(a) Intensity function

(b) Blue: Kernel intensity, Red: Background intensity

Moreover, as lockdown reopens, we can see the reproduction number increases from 0.962 to 0.9996 which suggests the spread of Covid is very close to spiralling out of control, which implies that easing the restrictions is causing a negative impact. This in fact ended up spreading at an uncontrollable rate, which caused another lockdown in November 2020. [22]

11 Conclusion

We believe that statistical modelling is an effective method to understand and visualize the spread of diseases. To summarise the aims of this paper, we have looked at the Ebola and Covid-19 data and we would like to investigate the effectiveness of Hawkes process for simulating infectious diseases. The benefit of modelling such diseases is that we could extend our models using simulations at a specific point of the data and hence compare the simulations with the actual data. A popular way for such simulation is by simulating the final 20% of the data. This method provides reliability of our simulation and we have demonstrated the method in the paper.

Throughout the paper, we provided a brief summary and introduction of Poisson process and Hawkes process, along with their characteristics and their use in the world of statistical modelling. We then discussed the choices of different kernels for Hawkes process and other indicators that we may find from the data, such as the R value. For data simulations, we have introduced two techniques of simulation, the Thinning and Branching method. We have also demonstrated the differences and advantages of using either method.

In order to understand the data, we initially created the Maximum Likelihood Estimation algorithm to find the desired parameters for Hawkes process. We tested the algorithm by recreating simulations using fixed parameters and we have shown that our algorithm aligns with the initial parameters. We then found the parameters using MLE from the data and we used Thinning and Branching method to produce 80/20 simulations. After the simulations, using the described techniques, we then compare the final 20% of the real data to the predictions that we obtained by box plots, which proves the reliability of the simulation.

We finally deduce that the use of Hawkes process is partially well suited for modelling diseases in cases with a stable external environment within a short period. We recognise from the covid data that Hawkes processes might not be a good fit when taking into account external factors, such as lockdowns enforced by the government and sudden disease outbreaks when the disease reaches a city. These events would either cause a spike or turnaround in cases that our simulations would not be able to predict. Also, the parameters it gives do not vary too much for Ebola and covid, even though the two diseases are different in terms of lethality and infectiousness. Otherwise, when we have enough data from the past we can conclude that Hawkes process will be a good fit within local regions where much more uniform results are simulated. To improve the model, we could try out more kernels or combine it woth other models like SIR. Overall, Hawkes Process is a good method for modelling diseases.

12 Appendix

1:

Proof. Consider the expectation of $\mathrm{N(t} + \delta_t) - \mathrm{N(t)}$, by definition,

$$\mathbb{E}[\mathrm{N(t} + \delta_t) - \mathrm{N(t)}] = \mathrm{P}(\mathrm{N(t} + \delta_t) - \mathrm{N(t)} = 1) \cdot 1 + \sum_{n \geq 2} \mathrm{P}(\mathrm{N(t} + \delta_t) - \mathrm{N(t)} = n) \cdot n$$

by definition 3, we obtain:

$$\mathbb{E}[\mathrm{N(t} + \delta_t) - \mathrm{N(t)}] = \lambda(\mathrm{t})\delta_t + \sum_{n \geq 1} \mathrm{o}(\delta_t) \cdot n$$

By linearity of expectation and dividing both sides by δ_t, we derive:

$$\frac{\mathbb{E}[\mathrm{N(t} + \delta_t)] - \mathbb{E}[\mathrm{N(t)}]}{\delta_t} = \lambda(t) + \sum_{n \geq 1} n \cdot \frac{\mathrm{o}(\delta_t)}{\delta_t}$$

let $\delta_t \to 0$, we get:

$$\lim_{\delta_t \to 0} \frac{\mathbb{E}[\mathrm{N(t} + \delta_t)] - \mathbb{E}[\mathrm{N(t)}]}{\delta_t} = \lambda(t) + \sum_{n \geq 1} n \cdot \lim_{\delta_t \to 0} \frac{\mathrm{o}(\delta_t)}{\delta_t}$$

by definition 3 and differentiation from first principles, we get:

$$\frac{\mathrm{d}}{\mathrm{dt}} \mathbb{E}[\mathrm{N(t)}] = \lambda(t) + \sum_{n \geq 1} n \cdot 0 = \lambda(t)$$

finally we obtain the result by the fundamental theorem of calculus:

$$\int_0^t \lambda(s) \mathrm{ds} = \mathbb{E}[\mathrm{N(t)}]$$

□

2:

Proof. Let H_t denote the history of times of all previous events $T_1, T_2, ..., T_n$ such that $T_n + 1 \geq T_n$, the lists of events are up to and does not include the time t. We can then define the conditional probability function $f^*(t) = f\left(t \mid H_t\right)$, which calculates the probability in the time of next event, T_{n+1}, and is dependent on all of the previous events $T_1, T_2, ..., T_n$. By the behaviour and characteristics of Hawkes process, we know that $f(T_i) = f\left(T_i \mid T_1, T_2, \ldots, T_{i-1}\right)$ for each i and hence we can find out $f(H_t)$ to be the product of the all probability function up to T_n given all events before T_n. We therefore obtain that the likelihood function

$L(\theta)$ to be the product of the all conditional probability function up to T_n:

$$\prod_{i=1}^{n} f^*\left(T_i\right)=L(\theta)$$

According to [23], we find out that the conditional intensity function $\lambda^*(t)$ can be linked to the likelihood function by the equation

$$\lambda^*(t)=\lambda\left(t \mid H_t\right)=\frac{f\left(t \mid H_t\right)}{1-F^*\left(t \mid H_t\right)}=\frac{f^*(t)}{1-F^*(t)}$$

where $F\left(t \mid H_t\right)=F^*(t)$ denotes the cumulative distribution function of the conditional probability function. We then would like to find $f^*(t)$ and $F^*(t)$ in terms of λ. By differentiation, it is clear to note that $f^*(t)=\frac{\partial}{\partial t} F^*(t)$, and hence we find that $\lambda^*(t)$ could be expressed as a partial derivative of $1-F^*(t)$.

$$\lambda^*(t)=\frac{\frac{\partial}{\partial t} F^*(t)}{1-F^*(t)}=\frac{\partial}{\partial t}\left(1-F^*(t)\right)$$

We then aim to get rid of the partial derivative by integrating both sides by t from T_n to t (note that T_n is up to but not including t), we can find out that

$$\begin{aligned} \int_{T_n}^t \lambda^*(u) d s & =-\left[\log \left(1-F^*(t)\right)-\log \left(1-F^*\left(T_n\right)\right)\right] \\ & =-\left[\log \left(1-F^*(t)\right)-\log (1-0)\right] \\ & =-\log \left(1-F^*(t)\right) \end{aligned} \tag{10}$$

Rearranging the equation to find out $F^*(t)$, which can then be used to find $f^*(t)$. Finally, according to [3], we get

$$f^*(t)=\lambda(t) \exp ^{\int_{T_n}^t \lambda(s) d s}$$

Finally, we sub $f^*(t)$ back into the likelihood equation to obtain the final result.

□

3:

$$\begin{aligned}
\int_0^T \lambda(t)dt &= \int_0^T \mu + \sum_{t_i<t} \alpha e^{-\delta(t-t_i)} \mathrm{dt} \\
&= \int_0^T \mu \mathrm{dt} + \int_{t_1}^T \alpha e^{-\delta(t-t_i)} \mathrm{dt} + ... + \int_{t_{N(T)}}^T \alpha e^{-\delta(t-t_{N(T)})} \mathrm{dt} \\
&= [\mu t]_0^T + \sum_{t_i<T} \int_{t_i}^T \alpha e^{-\delta(t-t_i)} \mathrm{dt} \\
&= \mu T + \sum_{t_i<T} \alpha e^{-\delta t_i} \left[\frac{e^{-\delta t}}{-\delta} \right]_{t_i}^T \\
&= \mu T + \sum_{t_i<T} \frac{\alpha}{\delta} e^{\delta t_i} \left(e^{-\delta t_i} - e^{-\delta T} \right) \\
&= \mu T + \sum_{t_i<T} \frac{\alpha}{\delta} \left(1 - e^{-\delta(T-T_i)} \right)
\end{aligned} \tag{11}$$

References

[1] Pishro-Nik H. Introduction to probability, statistics, and random processes. Kappa Research LLC; 2014. Available from: https://www.probabilitycourse.com/.

[2] Kingman JCF. Poisson processes (Oxford Studies in probability; 3). Clarendon Press; 1993.

[3] Marian-Andrei Rizoiu SMLX Young Lee. A Tutorial on Hawkes Processes for Events in Social Media. 2017:191-218.

[4] Reinhart A. A review of self-exciting spatio-temporal point processes and their applications. Cornell University; 2018. Available from: https://arxiv.org/abs/1708.02647.

[5] Ozaki T. Maximum likelihood estimation of Hawkes' self-exciting point processes. Annals of the Institute of Statistical Mathematics. 1979;31(1):145–155.

[6] Delamater PL, Street EJ, Leslie TF, Yang YT, Jacobsen KH. Complexity of the Basic Reproduction Number (R0). Emerging Infectious Diseases. 2019 Jan;25(1):1–4.

[7] H Juliette T Unwin SFMARSLJCDJWSMSB Isobel Routledge. Using Hawkes Processes to model imported and local malaria cases in near-elimination settings. PLOS computational biology. 2021;11.

[8] Garetto M, Leonardi E, Torrisi GL. A Time-modulated Hawkes process to model the spread of covid-19 and the impact of countermeasures. Annual Reviews in Control. 2021;51:551–563.

[9] Daley DJ, Vere-Jones D. An Introduction to the Theory of Point Processes 2003. 2nd. Springer-Verlag;.

[10] Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods. 2020;17:261-72.

[11] Ebola Disease. Centers for Disease Control and Prevention; 2023. Last accessed 18 June 2023. Available from: https://www.cdc.gov/vhf/ebola/index.html.

[12] Ending isolation and precautions for people with covid-19: Interim guidance. Centers for Disease Control and Prevention; 2022. Last accessed 18 June 2023. Available from: https://www.cdc.gov/coronavirus/2019-ncov/hcp/duration-isolation.html.

[13] Factors that contributed to undetected spread of the Ebola virus and impeded rapid containment. World Health Organization; 2015. Last accessed 18 June 2023. Available from: https://www.who.int/news-room/spotlight/one-year-into-the-ebola-epidemic/factors-that-contributed-to-undetected-spread-of-the-ebola-virus-and-impeded-rapid-containment.

[14] 2014-2016 Ebola Outbreak in West Africa. Centers for Disease Control and Prevention; 2019. Last accessed 18 June 2023. Available from: https://www.cdc.gov/vhf/ebola/history/2014-2016-outbreak/index.html.

[15] Tackling the epidemic across country borders. Medecins Sans Frontieres; 2015. Last accessed 18 June 2023. Available from: https://www.msf.org/ebola-tackling-epidemic-across-country-borders.

[16] Nyenswah TG, Kateh F, Bawo L, Massaquoi M, Gbanyan M, Fallah M, et al. Ebola and its control in Liberia, 2014–2015. Emerging Infectious Diseases. 2016;22(2):169–177.

[17] Ajelli MSFLea M. Spatiotemporal dynamics of the Ebola epidemic in Guinea and implications for vaccination and disease elimination: a computational modeling analysis. BMC Medicine. 2016;14(130).

[18] Kaner SS J. Understanding Ebola: the 2014 epidemic. 2016;12(1).

[19] WHO Coronavirus (COVID-19) Dashboard. World Health Organization;. Last accessed 18 June 2023. Available from: https://covid19.who.int/.

[20] World Health Organization; 2021. Available from: https://www.who.int/news-room/questions-and-answers/item/coronavirus-disease-covid-19-how-is-it-transmitted.

[21] Disease outbreaks database. Github user: finlaycampbell; 2020. Last accessed 18 June 2023. Available from: https://github.com/reconverse/outbreaks/tree/master/data.

[22] Timeline of UK government coronavirus lockdowns and restrictions. INSTITUTE FOR GOVERNMENT; 2022. Available from: https://www.instituteforgovernment.org.uk/sites/default/files/2022-12/timeline-coronavirus-lockdown-december-2021.pdf.

[23] Rasmussen JG. Temporal point processes and the conditional intensity function; 2018. Available from: https://arxiv.org/abs/1806.00221.