Dipl.-Ing. Aleksandar Karakaš, BSc BSc MSc

On the Relation between Stochastic Epidemic Models and Self-Exciting Point Processes

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Software Development and Business Management

submitted to

Graz University of Technology

Supervisor

Assoc.Prof. Dipl.-Ing. Dr.techn. Denis Helić

Institute of Interactive Systems and Data Science

Graz, February 2019

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master‘s thesis.

Date

Signature

Acknowledgments

I want to thank Prof. Helić for offering me the opportunity to write this master's thesis and for his guidance throughout this process. I am also thankful for my family's sustained support.

Abstract

Rizoiu et al. (2018b) present a link between the stochastic SIR model and a self exciting point process which they call the HawkesN process. The intensity of this variant of the Hawkes process features an exponential kernel. We have found a connection between the SEIR model and the HawkesN process. It leads to two new excitation functions and we show that one of them generalizes the exponential kernel. Furthermore, we discuss how parameters of an SIR or SEIR model can be estimated using HawkesN processes. This is useful when for example modeling information diffusion where we can observe only one type of events. In contrast to Rizoiu et al. (2018b) we present our results for several parameter combinations and also consider the case where all parameters are unknown.

Kurzfassung

Rizoiu et al. (2018b) präsentieren eine Verbindung zwischen dem stochastischen SIR-Modell und dem HawkesN-Prozess – einem Hawkes-Prozess, dessen Intensität durch einen zusätzlichen Faktor modifiziert wird. Die Intensität des von Rizoiu et al. (2018b) gefundenen Prozesses weist einen exponentiellen Kernel auf. Wir zeigen, dass auch eine Verbindung zwischen dem HawkesN-Prozess und dem SEIR-Modell existiert. Diese führt zu zwei neuen Kernels, wobei eine dieser beiden Funktionen eine Verallgemeinerung des exponentiellen Kernels darstellt. Weiters zeigen wir in der vorliegenden Arbeit, wie die Parameter des SIR- und SEIR-Modells mit Hilfe von Hawkes-Prozessen geschätzt werden können. Dies ist unter anderem bei der Modellierung von Informationsausbreitungsprozessen hilfreich, da hierbei nur eine Ereignisart beobachtbar ist. Im Gegensatz zu Rizoiu et al. (2018b) präsentieren wir unsere Resultate für verschiedene Parameterkombinationen und betrachten auch den Fall, in dem sämtliche Parameter als unbekannt angenommen werden.

Contents

List of Figures

List of Tables

List of Algorithms

1. Introduction

Rizoiu et al. (2018b) present a connection between the stochastic SIR model and a variation of the Hawkes process which they call the HawkesN process. This link between the two models requires the HawkesN process to have an exponential kernel as excitation function. This is why Rizoiu et al. (2018b) considered the kernel function as an area of future work. In this thesis we introduce two new kernel functions for the HawkesN process establishing the link to the SEIR model. Furthermore, we show that the exponential kernel is a special case of our more general excitation function.

Using simulated SIR data, Rizoiu et al. (2018b) show that parameter estimation works better with likelihood maximization of the SIR model than with the corresponding approach related to the Hawkes process. However, in the generation of the simulations they use only one set of parameters. We investigate how the choice of different parameters affects the estimation results. Furthermore, they assume that all but one of the parameters are fixed when fitting the parameter they are most interested in. Since knowing all but one parameter is not very realistic, we analyze the consequences of estimating more than one parameter.

This thesis is divided into the following parts. In chapter 2 we deal with some basics in probability theory. We define some distributions needed in the rest of the thesis and we dive into the theory of stochastic processes with an emphasis on point processes. In chapter 3 we introduce the SIR as well as the SEIR model and discuss a deterministic and a stochastic variant of each model. Furthermore, we present how the final size of an epidemic can be modeled. Chapter 4 is then devoted to Hawkes processes. In this chapter we also introduce the HawkesN process as defined in Rizoiu et al. (2018b). Following these authors we show that there is a close relation between the SIR model and the HawkesN process. This chapter contains our new kernel

function as well and we present the link between SEIR models and HawkesN processes. Chapter 5 contains our conclusions.

In appendices B and C we demonstrate the differences between the R program by Rizoiu et al. (2018a) and our software for working with epidemic models as well as HawkesN processes. Since this part of the thesis contains relatively many code snippets we provide instructions on how to read these parts in appendix A.

2. Probability Theory and Stochastic Calculus

2.1. Introduction

In this chapter we first introduce distributions we use in later chapters. After that we provide an introduction into the theory of stochastic processes. Throughout the chapter we use the following notation.

We use Ω to denote a space of elementary events and we use $\mathcal{A}$ to denote a σ-algebra on Ω. A probability measure on $\mathcal{A}$ is denoted by $\mathbb{P}$. Thus, $(\Omega, \mathcal{A})$ is a measurable space and $(\Omega, \mathcal{A}, \mathbb{P})$ is a probability space.

2.2. Useful Distributions

In this section we introduce useful distributions. All definitions and statements can also be found in Klenke (2013). The first distribution we introduce is the exponential distribution.

Definition 2.1 (Exponential distribution)**.** Let $\lambda > 0$ and let X be a nonnegative random variable such that

$$\mathbb{P}[X \leq x] = \mathbb{P}\left[X \in [0, x]\right] = \int_0^x \lambda \, \mathrm{e}^{-\lambda t} \, \mathrm{d}t \quad \text{for } x \geq 0.$$

Then we call X *exponentially distributed* with parameter λ. We denote the distribution of X by $\exp_\lambda$.

We can calculate the integral in definition 2.1 and get

$$\mathbb{P}[X \leq x] = \int_0^x \lambda \, \mathrm{e}^{-\lambda t} \, \mathrm{d}t = \lambda \frac{\mathrm{e}^{-\lambda t}}{-\lambda} \Big|_0^x = 1 - \mathrm{e}^{-\lambda x}.$$

This implies

$$\overline{F}_X(x) := \mathbb{P}[X > x] = \mathrm{e}^{-\lambda x}. \tag{2.1}$$

For a random variable X we call $\overline{F}_X$ the *tail* of the distribution of X.

For independent exponentially distributed random variables X_i for $i = 1, 2, \ldots, n$ we might be interested in the distribution of the random variable $\min\{X_1, X_2, \ldots, X_n\}$. It turns out that this variable is again exponentially distributed as shown in the following theorem.

Theorem 2.2. *Let $X_i \overset{ind}{\sim} \exp_{\lambda_i}$ for $i = 1, \ldots, n$ be independent exponentially distributed random variables with parameters $\lambda_1, \ldots, \lambda_n$. Then the random variable $\min\{X_1, \ldots, X_n\}$ is also exponentially distributed with parameter $\lambda = \sum_{i=1}^n \lambda_i$.*

Proof. Using the definition of the minimum and the independence of the random variables we get

$$\mathbb{P}(\min\{X_1, \ldots, X_n\} > x) = \mathbb{P}(X_1 > x, \ldots, X_n > x) = \prod_{i=1}^n \mathbb{P}(X_i > x).$$

With equation (2.1) we get

$$\prod_{i=1}^n \mathbb{P}(X_i > x) = \prod_{i=1}^n \mathrm{e}^{-\lambda_i x}.$$

Using the properties of the exponential function we obtain

$$\prod_{i=1}^n \mathrm{e}^{-\lambda_i x} = \mathrm{e}^{-\sum_{i=1}^n \lambda_i x} = \mathrm{e}^{-\lambda x}.$$

Thus we get

$$\mathbb{P}(\min\{X_1, \ldots, X_n\} \leq x) = 1 - \mathrm{e}^{-\lambda x} \tag{2.2}$$

which proves that $\min\{X_1, \ldots, X_n\}$ is exponentially distributed with parameter λ. □

Now, we continue with the next distribution, the Poisson distribution.

Definition 2.3 (Poisson distribution)**.** Let $X \in \mathbb{N}_{\geq 0}$ be a random variable with $\mathbb{P}(X = n) = \frac{\lambda^n}{n!} e^{-\lambda} \, \forall n \in \mathbb{N}_{\geq 0}$. Then X has a *Poisson distribution* with parameter λ (denoted by $X \sim Poi(\lambda)$).

2.3. Stochastic Calculus

In this section we introduce some basics from stochastic calculus. The first definitions of this chapter are taken from Klenke (2013).

Definition 2.4 (Stochastic process)**.** Let $I \subset \mathbb{R}$. A family of random variables $X = (X_t)_{t \in I}$ (on $(\Omega, \mathcal{A}, \mathbb{P})$) with values in the measurable space $(E, \mathcal{E})$ is called a *stochastic process* with index set (or time set) I and range E.

All stochastic process we consider have $E = \mathbb{R}$. Such processes are called *real-valued*. Usually I is $[0, \infty)$. Thus $(X_t)_{t \geq 0}$ is an alternative way of expressing X.

Note that other views on stochastic processes are possible. As mentioned in Andersen et al. (1996) one can consider a stochastic process as a function of two arguments $t, \omega \in I \times \Omega$. With ω fixed we have a *sample path* of the stochastic process.

Definition 2.5. Let X be a random vaiable (or a stochastic process). Then we denote the distribution of X by $\mathcal{L}[X]$. For a σ-algebra $\mathcal{G} \subset \mathcal{A}$ we define $\mathcal{L}[X|\mathcal{G}]$ as the conditional distribution of X given $\mathcal{G}$.

Definition 2.6 (Filtration)**.** Let $(\Omega, \mathcal{A}, \mathbb{P})$be a probability space and let $\mathbb{F} = (\mathcal{F}_t)_{t \in I}$ be a family of σ-algebras with $\mathcal{F}_t \subset \mathcal{A}$ for all $t \in I$. $\mathbb{F}$ is called a *filtration* if $\mathcal{F}_s \subset \mathcal{F}_t$ for all $s, t \in I$ with $s \leq t$. The space $(\Omega, \mathcal{A}, \mathbb{F}, \mathbb{P})$is a *filtrated probability space*.

We can consider a filtration $\mathbb{F}$ as our history, i.e. the σ-algebra $\mathcal{F}_t$ contains all the information we have until time t. Since by definition a filtration's σ-algebras are increasing, we gain more and more information as time passes.

Definition 2.7. A stochastic process $(X_t)_{t\in I}$ is called *adapted* to the filtration $\mathbb{F} = (\mathcal{F}_t)_{t\in I}$ if X_t is $\mathcal{F}_t$-measurable for all $t \in I$. If $\mathbb{F} = (\mathcal{F}_t)_{t\in I}$ and $\mathcal{F}_t = \sigma((X_s)_{s\leq t})$ for all $t \in I$, then we call the filtration *generated by* X and write $\mathbb{F} =: \sigma(X)$.

Viewing a filtration as our history allows us to give an intuitive explanation of what definition 2.7 means. Aalen et al. (2008) point out that X being adapted to the filtration $\mathbb{F} = (\mathcal{F}_t)_{t\in I}$ means that our history $\mathbb{F}$ is generated by the process X and maybe some additional external information. So all information about X is contained in $\mathbb{F}$. Thus, for example, we can construct the conditional expectation $E(X_t|\mathcal{F}_t) = X_t$ and still recover the original random variable X_t without losing information about it. Since the σ-algebras are increasing, this property even holds for all X_s where $s \leq t$.

Definition 2.8. We call a filtration $\mathbb{F} = (\mathcal{F}_t)_{t\geq 0}$ $\mathbb{P}$-*complete*, if $\mathcal{F}_0$ contains all subsets $A \subseteq \Omega$ with $\mathbb{P}(A) = 0$. Note that this property means that all σ-algebras $\mathcal{F}_t$ with $t > 0$ contain the null sets as well because a filtration's σ-algebras are increasing.

Definition 2.9. We define $\mathcal{F}_{t+} := \cap_{s>t}\mathcal{F}_s$ and call a filtration $\mathbb{F} = (\mathcal{F}_t)_{t\geq 0}$ *right-continuous* if $\mathcal{F}_{t+} = \mathcal{F}_t$ for all $t \geq 0$.

Definition 2.10. A filtration $\mathbb{F}$ is said to satisfy the *usual conditions* if it is $\mathbb{P}$-complete and right-continuous. In this thesis we always assume this property to hold.

Next, we define a counting process basing our definition on Rüschendorf (2014). In the following we use the terms counting process and point process interchangeably.

Definition 2.11 (Counting process)**.** Let $(X_n)_{n\in\mathbb{N}}$ be a sequence of random variables $X_n \in (0, \infty)$ with $X_n > X_{n-1}$ $\mathbb{P}$-almost-surely. We define

$$N_t := \sum_{n=1}^{\infty} I_{\{X_n\leq t\}}$$

and call N a *counting process* or *point process*.

Daley and Vere-Jones (2003) point out that there are also other ways to define a point process. They introduce point processes using a counting measure. This approach has also been chosen by Resnick (1992).

If the realizations of two consecutive points X_n and X_{n+1} are t_n and t_{n+1}, respectively, then we can see from definition 2.11 that the corresponding counting process N is constant on $[t_n, t_{n+1})$. The occurrence of a new point immediately causes a discontinuity in N, i.e. $N_{t_{n+1}} = \lim_{s \uparrow t_{n+1}} N_s + 1 \neq \lim_{s \uparrow t_{n+1}} N_s$. Note that the limit $\lim_{s \uparrow t_{n+1}} N_s = N_{t_n}$ exists and that N is right-continuous because of $lim_{s \downarrow t_{n+1}} N_s = N_{t_{n+1}}$. Such processes, i.e. right-continuous with a limit on the left, are called *càdlàg*, an acronym of the French *continue à droite, limite à gauche*.

Definition 2.12 (Waiting times)**.** Let us again consider the sequence $(X_n)_{n \in \mathbb{N}}$ from definition 2.11. The random variables $T_n := X_n - X_{n-1}$ are called *waiting times*.

According to Daley and Vere-Jones (2003) waiting times are an alternative way of defining a point process.

When talking about a counting process N we always assume a filtrated probability space $(\Omega, \mathcal{A}, \mathbb{F}, \mathbb{P})$such that N is adapted to $\mathbb{F}$. In order to introduce the concept of the intensity of a counting process we first have to discuss a few more fundamentals of stochastic calculus, as also shown in Aalen et al. (2008).

Definition 2.13. Let X be a stochastic process.

1. X is called *integrable* if it is real-valued and $\forall t \in I : \mathbb{E}\left[|X_t|\right] < \infty$.
2. X is called *square integrable* if it is real-valued and $\forall t \in I : \mathbb{E}\left[X_t^2\right] < \infty$.

Definition 2.14 (Martingale)**.** Let $(\Omega, \mathcal{A}, \mathbb{F}, \mathbb{P})$be a filtrated probability space. A stochastic process X is called a *martingale* with respect to the filtration $\mathbb{F}$ and the probability measure $\mathbb{P}$ if

1. X is integrable,
2. X is adapted to $\mathbb{F}$, and
3. the martingale condition

$$\mathbb{E}[X_t | \mathcal{F}_s] = X_s \tag{2.3}$$

holds $\mathbb{P}$-almost surely for all $s \in [0, t]$.

If instead of (2.3) only

$$\mathbb{E}[X_t|\mathcal{F}_s] \geq X_s \tag{2.4}$$

holds $\mathbb{P}$-almost surely for all $s \in [0,t]$, then we call X a *submartingale*.

As indicated by the inequality sign in (2.4) submartingales tend to grow over time. Note that the inequality (2.4) holds for counting processes since they are monotonically increasing. This means that all counting processes are submartingales.

We now introduce a feature of stochastic processes called predictability as defined in Andersen et al. (1996).

Definition 2.15 (predictable)**.** Let $(\Omega, \mathcal{A}, \mathbb{F}, \mathbb{P})$ be a filtered probability space. A stochastic process is called *predictable* if, seen as a function of $(t,\omega) \in I \times \Omega$, it is measurable with respect to the σ-algebra on $I \times \Omega$ generated by the left-continuous processes that are adapted to $\mathbb{F}$.

A submartingale X can be split into two processes using the Doob-Meyer decomposition. From it we obtain $X = X^* + M$. In this equation X^* is nondecreasing and predictable and M is a martingale with $E(M_t) = 0$ for all $t \in I$. Since a counting process N is a submartingale we may write

$$N_t = \Lambda_t + M_t \tag{2.5}$$

where Λ_t is a predictable and nondecreasing.

Definition 2.16. We call Λ in equation (2.5) the *cumulative intensity* or *compensator* of N.

From (2.5) we obtain $M_t = N_t - \Lambda_t$ which means that a counting process can be turned into a zero-mean martingale by subtracting Λ_t. For our purposes we assume that Λ_t is absolutely continuous which means that there exists a process λ_t (which is also predictable) such that

$$\Lambda_t = \int_0^t \lambda_s \, \mathrm{d}s \,. \tag{2.6}$$

Thus

$$M_t = N_t - \int_0^t \lambda_s \, \mathrm{d}s \tag{2.7}$$

is a martingale with $\mathbb{E}[M_t] = 0$ for all $t \in I$.

Definition 2.17. We call λ in equation (2.6) the *intensity process* or *intensity* of the counting process N.

Sometimes, the intensity λ is defined in a different way. For example, Laub et al. (2015) as well as Bacry et al. (2015) use

$$\lambda_t = \lim_{\Delta t \downarrow 0} \frac{\mathbb{E}[N_{t+\Delta t} - N_t | \mathcal{F}_t]}{\Delta t} \tag{2.8}$$

as a definition. As argued above, the probability of multiple events between t and $t + \Delta t$ converges to zero with $\Delta t \to 0$. This is why Aalen et al. (2008) write

$$\lambda_t = \lim_{\Delta t \downarrow 0} \frac{\mathbb{P}[N_{t+\Delta t} = N_t + 1 | \mathcal{F}_t]}{\Delta t} \tag{2.9}$$

where the expectation in (2.8) is replaced by a probability. They also point out that in equation (2.5) and its equivalent $dN_t = \lambda_t dt + dM_t$ the left side represents our observation whereas the right side is a sum of the signal and noise.

Next, we define the integration with respect to a counting process. Often in the literature, the more general integral with respect to a function with locally finite variation is defined. As each path of a counting process is a function of locally finite variation, such an approach is certainly correct. However, we skip the theory on a function's variation and just present the important result.

Definition 2.18 (Integral w.r.t. a counting process)**.** Let N be a counting process and let t_j be the time of the j^{th} jump event of N. Let $f : \mathbb{R}_{\geq 0} \to \mathbb{R}$ be a Lebesgue-integrable function. Then we define

$$\int_0^t f(s) \, dN(s) := \sum_{t_j < t} f(t_j).$$

This definition is intuitive in the sense that $dN(t_j) = 1$ since the counting process N experiences a jump of size 1 at time t_j. If there is no jump at time t, we have $dN(t) = 0$. Thus, f is only considered at the jump times of the integrator N.

3. Compartmental Epidemic Models

3.1. Introduction

In this chapter we discuss how the spread of contagious diseases can be modeled using two compartmental epidemic models. One of the two is the SIR model where the population is divided into the three compartments S, I, and R. Compartment S contains all the individuals that are not ill but are *susceptible* to the disease. Compartment I contains those individuals that are already *infected*, while those that have *recovered* from the disease are in compartment R. Susceptible individuals can become infected by individuals from compartment I. In this case they move from S to I. Infected individuals may recover after some time has passed in which case they move from I to R. Recovered individuals cannot infect anybody and they cannot be infected anymore. Figure 3.1 summarizes the compartments and possible transitions.

The second model we discuss in this chapter is the SEIR model. It is related to the SIR and extends it by a fourth compartment. This compartment, denoted by E, comprises all the individuals that are *exposed*, meaning that these individuals are infected but not are not able to transmit the disease yet.

Figure 3.1.: Compartments and possible transitions in an SIR model.

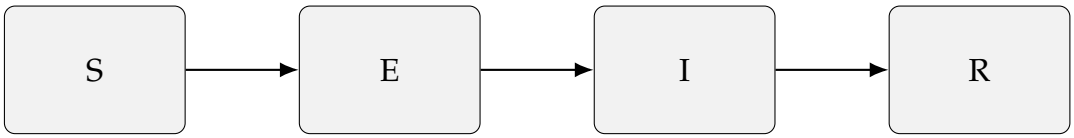


Figure 3.2.: Compartments and possible transitions in an SEIR model.

Only after the transition from E to I they become *infectious* or *infective*. So in this model there is no direct transition from S to I but rather from S to E and then to I. The SEIR model inherits the transition from I to R from the SIR model. Figure 3.2 gives an overview of the SEIR model.

For both epidemic models we distinguish a deterministic from a stochastic model. In sections 3.2 and 3.3 we discuss the deterministic and the stochastic SIR model respectively. The deterministic SEIR model follows in section 3.4. Section 3.5 about the stochastic SEIR model concludes this chapter.

3.2. The Deterministic SIR Model

In this section we introduce the deterministic SIR model which is also known as Kermack McKendrick epidemic model. Martcheva (2015) served as a basis for this section.

In the following, we denote the number of individuals in compartments S, I, and R at time t by $S(t)$, $I(t)$, and $R(t)$ respectively. Starting with initial values $S(0), I(0), R(0) \geq 0$ the (deterministic) processes $S(t)$, $I(t)$, and $R(t)$ evolve over time following the differential equations

$$\frac{dS(t)}{dt} = -\beta \frac{S(t)}{N} I(t), \tag{3.1}$$

$$\frac{dI(t)}{dt} = \left(\beta \frac{S(t)}{N} - \gamma\right) I(t), \quad \text{and} \tag{3.2}$$

$$\frac{dR(t)}{dt} = \gamma I(t). \tag{3.3}$$

The total population at time $t = 0$ is $N(0) = S(0) + I(0) + R(0)$. Since births and deaths are not modeled and the population is considered to be

closed (i.e. no individuals are allowed to enter or leave the population), the population's size does not change over time. This fact is also reflected by the derivative

$$N'(t) = S'(t) + I'(t) + R'(t) = 0. \tag{3.4}$$

Since $N(t)$ is constant we may define $N := N(t)$ for any $t \geq 0$.

The constants $\beta, \gamma \geq 0$ in (3.1), (3.2), and (3.3) denote the infection rate and recovery rate, respectively. The spread of a disease is determined by these two constants as well as the initial distribution. Changing one of these model parameters results in a different pattern of the spread of the disease.

By equation (3.1) $S(t)$ is monotonically decreasing. $R(t)$, on the other hand, is monotonically increasing because of (3.3). If $\beta\frac{S(t)}{N} - \gamma \leq 0$ holds at the beginning of the process, then the number of infected individuals is monotonically decreasing due to (3.2). Otherwise, the number of individuals in I is increasing until $\beta\frac{S(t)}{N} - \gamma \leq 0$ holds. This will happen at some point in time – let us call it u – since $S(t)$ is monotonically decreasing. For $t \geq u$ the number of infected individuals $I(t)$ is monotonically decreasing.

These two cases are depicted in figures 3.3 and 3.4. In both situations we have $S(0) = 80$, $I(0) = 20$, and $R(0) = 0$. In figure 3.3 the infection rate is 0.5 while the recovery rate amounts to 0.7. Thus, $\beta\frac{S(0)}{N} - \gamma = 0.5 * \frac{80}{100} - 0.7 = -0.3 < 0$, which means that I is monotonically decreasing due to equation (3.2).

In figure 3.4 we have $\beta = 1$ and $\gamma = 0.3$. Hence, $\beta\frac{S(0)}{N} - \gamma = 1 * \frac{80}{100} - 0.3 = 0.5 > 0$. This explains why in this figure I is increasing at first.

Often, one wants to know the final size of an epidemic, i.e. the total number of individuals that have become sick over time. This amount is expressed by the limit $\lim_{t\to\infty} R(t) =: R(\infty)$ because $I(t) \overset{t\to\infty}{\longrightarrow} 0$ holds according to theorem 2 in Allen (2008). It turns out that we can derive a relation involving only $R(\infty)$, the starting values $S(0)$ and $I(0)$, as well as the model parameters β and γ. To this end we use $R(\infty) = N - S(\infty)$, where $S(\infty)$ denotes the limit $\lim_{t\to\infty} S(t)$ and derive the corresponding relation involving $S(\infty)$ instead of $R(\infty)$. This procedure can also be seen in Brauer

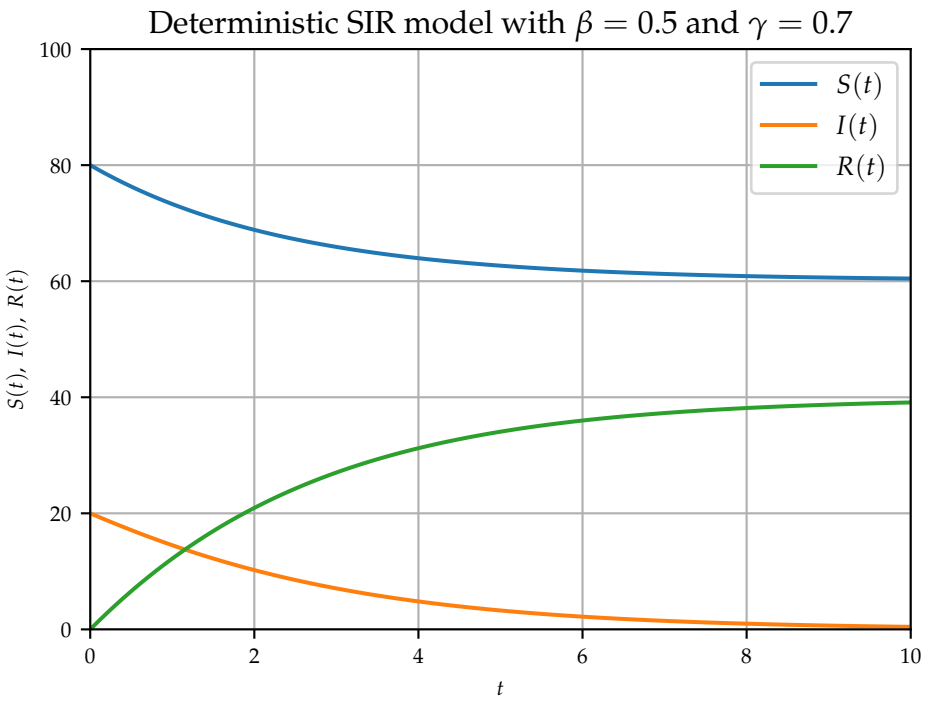


Figure 3.3.: A deterministic SIR model with monotonically decreasing number of infected.

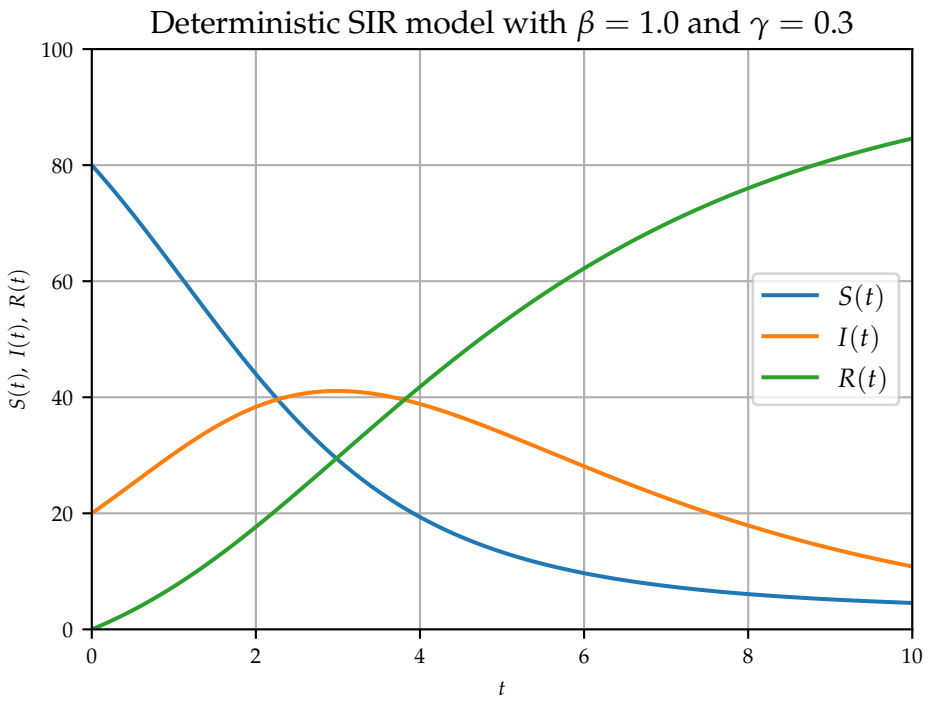


Figure 3.4.: A deterministic SIR model where the number of infected is increasing at first.

(2008) and Martcheva (2015). We start by using the equations (3.1) and (3.2) which leads us to

$$\frac{I'(t)}{S'(t)} = \frac{\frac{\mathrm{d}I(t)}{\mathrm{d}t}}{\frac{\mathrm{d}S(t)}{\mathrm{d}t}} = \frac{\left(\beta\frac{S(t)}{N} - \gamma\right) I(t)}{-\beta\frac{S(t)}{N} I(t)} = -1 + \frac{\gamma}{\beta\frac{S(t)}{N}} = -1 + \frac{\gamma N}{\beta S(t)}. \tag{3.5}$$

Multiplying (3.5) by S' we obtain

$$I'(t) = -S'(t) + \frac{S'(t)\gamma N}{\beta S(t)}$$

and integrating leads to

$$\begin{aligned} I(t) - I(0) &= \int_0^t I'(u)\,\mathrm{d}u \\ &= \int_0^t \left(-S'(u) + \frac{S'(u)\gamma N}{\beta S(u)}\right) \mathrm{d}u \\ &= -\int_0^t S'(u)\,\mathrm{d}u + \frac{\gamma}{\beta} N \int_0^t \frac{S'(u)}{S(u)}\,\mathrm{d}u \\ &= -S(t) + S(0) + \frac{\gamma}{\beta} N \left(\log(S(t)) - \log(S(0))\right). \end{aligned} \tag{3.6}$$

Since equation (3.6) holds for all $t \geq 0$, it holds for the limit $t \to \infty$. Using $\lim_{t\to\infty} I(t) = 0$ we get

$$-I(0) = -S(\infty) + S(0) + \frac{\gamma}{\beta} N \left(\log(S(\infty)) - \log(S(0))\right). \tag{3.7}$$

We can rearrange equation (3.7) to obtain

$$S(\infty) = I(0) + S(0) + \frac{\gamma}{\beta} N \left(\log(S(\infty)) - \log(S(0))\right). \tag{3.8}$$

(3.8) is an implicit equation for $S(\infty)$. On the left side we have $S(\infty)$ and on the right side there is a function of it. So in order to determine the value of

$S(\infty)$ we have to find a fixed point of the function on the right side of (3.8) which is

$$f(x) = I(0) + S(0) + \frac{\gamma}{\beta} N \left(\log(x) - \log(S(0))\right). \tag{3.9}$$

With the fixed point $S(\infty)$ we also get the final size of the epidemic $R(\infty) = N - S(\infty)$.

For the example depicted in figure 3.3 we get a result of $S(\infty) \approx 60.21$. This does not contradict the figure which indicates a value near 60. In the example of figure 3.4 the corresponding result is around 3.17.

The code for generating the figures as well as for calculating $S(\infty)$ resides in the Jupyter notebook for this section in Karakaš (2019).

3.3. The Stochastic SIR Model

3.3.1. Introduction

Now, we turn to a stochastic version of the SIR model. Allen (2008) discusses several models which introduce randomness to the SIR model. Here we only discuss the Continuous Time Markov Chain SIR model (CTMC model). By continuous time we mean that we model our three compartments for each point in time $t \in \mathbb{R}_{\geq 0}$ (as opposed to the Discrete Time Markov Chain Model where $t \in \mathbb{N}_0$). We discuss the implications of the Markov chain assumptions after introducing the notation we use. Allen (2008) and Yan (2008) served as the basis for this section.

So far the number of individuals in the respective compartments was denoted by $S(t)$, $I(t)$, and $R(t)$. Note that these variables are deterministic functions of the time. In the CTMC model these quantities are expressed as stochastic processes. To stay in line with our discussion on stochastic processes we use the notation S_t, I_t, and R_t now. Note that in the deterministic case we had $S(t), I(t), R(t) \in \mathbb{R}$ whereas in the stochastic model we have $S_t, I_t, R_t \in \mathbb{N}_0$.

Instead of modeling all three processes, we follow Yan (2008) focusing only on R_t as well as

$$C_t := N - S_t. \tag{3.10}$$

These two processes in connection with starting values allow the reconstruction of S_t and I_t using $S_t = S_0 - (C_t - C_0)$ and $I_t = C_t - R_t$. This is why modeling solely R_t and C_t suffices.

Let t_j^C be the point in time at which the j^{th} individual becomes infected. The superscript ^{C} shall indicate that the process C increases at this point in time. Using this notation we can write

$$C_t = C_0 + \sum_{0 < t_j^C \leq t} 1. \tag{3.11}$$

So C_t represents a counting process (adjusted by the constant $C_0 = N - S_0 = I_0 + R_0$) and counts the number of infections that have occurred until time t.

Let t_j^R be the point in time at which the j^{th} individual recovers. Then R_t can be expressed as

$$R_t = R_0 + \sum_{0 < t_j^R \leq t} 1,$$

making it a counting process (adjusted by the constant R_0) as well. Thus the tuple (C_t, R_t) constitutes a 2-dimensional counting process.

The processes representing the compartments of susceptible and infected individuals, S_t and I_t, can be written as

$$S_t = S_0 - \sum_{0 < t_j^C \leq t} 1 \quad \text{and} \tag{3.12}$$

$$I_t = I_0 + \sum_{0 < t_j^C \leq t} 1 - \sum_{0 < t_j^R \leq t} 1, \tag{3.13}$$

respectively.

We can use the variables t_j^C and t_j^R to express for how long individual j suffers from the disease. This duration is denoted by

$$\tau_j := t_j^R - t_j^C. \tag{3.14}$$

So far we have only clarified our notation. We did not mention according to which dynamics our process evolves. Given a state (C_t, R_t) the process can change in different ways. Since the CTMC assumes a Markov process, we have $\mathbb{P}((C_{t+\Delta t}, R_{t+\Delta t}) = (c,r)|\mathcal{F}_t) = \mathbb{P}((C_{t+\Delta t}, R_{t+\Delta t}) = (c,r)|(C_t, R_t))$ for all states $(c,r) \in {\mathbb{N}_0}^2$. This means that the distribution of a new state depends only on the last known state rather than the whole history of the process.

Note that when the disease dies out, i.e. t with $I_t = 0$ is reached, the process never changes again. In this case we have $\mathbb{P}((C_{t+s}, R_{t+s}) = (N - S_t, N - S_t)|I_t = 0) = 1$ for all $s \geq 0$. We call such a state *absorbing*. An absorbing state in our model is characterized by our two processes of interest, C and R, being equal.

Unless we have reached an absorbing state $(C_t, R_t) = (c,c)$, the state will change eventually when time passes. The waiting times, that is the times between two successive state changes are then exponentially distributed, since the exponential distribution is the only memory-less continuous distribution. A waiting time of zero thus has a probability of zero which implies that there are almost surely no two events at the same time. We conclude that the process has only three possibilities (with a positive probability) to evolve over an infinitesimally small time interval.

The first possibility is that an infection takes place. This means that the process changes from $(C_t, R_t) = (c,r)$ to $(C_{t+\Delta t}, R_{t+\Delta t}) = (c+1, r)$. Alternatively, a recovery is possible, i.e. $(C_{t+\Delta t}, R_{t+\Delta t}) = (c, r+1)$. Of course, it is also possible that nothing happens in the time interval $[t, t+\Delta t]$, that is $(C_{t+\Delta t}, R_{t+\Delta t}) = (c,r)$. We neglect the possibility of multiple events during $[t, t+\Delta t]$ because we may choose Δt arbitrarily small and – as argued above – the probability of multiple events goes to 0 with $\Delta t \to 0$.

A more convenient notation for the probabilities of these three possibilities is achieved by defining

$$p_{(c+j,r+k),(c,r)}(\Delta t) := \mathbb{P}((C_{t+\Delta t}, R_{t+\Delta t}) = (c+j, r+k) | (C_t, R_t) = (c,r)), \tag{3.15}$$

where $(j,k) \in \{(1,0),(0,1),(0,0)\}$. According to Allen (2008) the equations

$$p_{(c+1,r),(c,r)}(\Delta t) = \beta \frac{S_t}{N} I_t \Delta t + o(\Delta t), \tag{3.16}$$

$$p_{(c,r+1),(c,r)}(\Delta t) = \gamma I_t \Delta t + o(\Delta t), \text{ and} \tag{3.17}$$

$$p_{(c,r),(c,r)}(\Delta t) = 1 - \beta \frac{S_t}{N} I_t \Delta t - \gamma I_t \Delta t + o(\Delta t) \tag{3.18}$$

hold.

Using the equations (2.9) and (3.16) we see that the intensity λ^C of the counting process C is

$$\lambda_t^C = \beta \frac{S_t}{N} I_t. \tag{3.19}$$

Combining (2.9) with (3.17) we see that

$$\lambda_t^R = \gamma I_t \tag{3.20}$$

holds where λ^R denotes the intensity of the counting process R.

3.3.2. Simulation

A simulation algorithm can be derived from the exponential distribution of the waiting times. Given a state of the three variables S_t, I_t, and R_t we can compute the intensities λ_t^C and λ_t^R. As argued above, both, the infection process C and the recovery process R have exponentially distributed waiting times. As shown in theorem 2.2 the waiting time until either an infection or a recovery occurs is again exponentially distributed with parameter

$$\lambda_t = \lambda_t^C + \lambda_t^R. \tag{3.21}$$

So we can simulate the waiting time until the next event. Note that $\frac{\lambda_t^C}{\lambda_t^C+\lambda_t^R}$ gives that probability that the next event is an infection (see e.g. Kingman (1992)). Thus the probability of the alternative event (the next event is a recovery) is $1 - \frac{\lambda_t^C}{\lambda_t^C+\lambda_t^R} = \frac{\lambda_t^R}{\lambda_t^C+\lambda_t^R}$. By simulating a uniformly distributed random variable on $[0,1]$ we can decide of which type the next event is. After updating the state accordingly we can go back to calculating the new intensities of C and R. Algorithm 3.1 is intended to clarify this procedure.

Algorithm 3.1 Simulation of a stochastic SIR process until time t_{max}.

Input: $S_0,\ I_0,\ R_0,\ \beta,\ \gamma \geq 0$ and $t_{max} > 0$
Output: A sample SIR process for $t \in [0, t_{max}]$

$N \leftarrow S_0 + I_0 + R_0$
$t \leftarrow 0$
state $\leftarrow \{S_0, I_0, R_0, t\}$
states $\leftarrow$ [state,] {states is a list with one entry after this assignment}
while $t < t_{max}$ **do**
  state $\leftarrow$ copy of last entry in states
  Compute λ_t^C and λ_t^R using state, β, and γ.
  $\lambda \leftarrow \lambda_t^C + \lambda_t^R$
  if $\lambda > 0$ **then**
    $u \leftarrow$ realization of a uniform random variable in $[0,1]$
    $t \leftarrow t - \frac{\log(u)}{\lambda}$
    The next event is an infection with prob. $\frac{\lambda_t^C}{\lambda_t^C+\lambda_t^R}$, otherwise a recovery.
    Update entries in state that represent S, I, and R accordingly.
  else
    $t \leftarrow t_{max}$ {absorbing state is reached}
  end if
  entry in state representing time $\leftarrow t$
  Append state to states.
end while
return states

Rizoiu et al. (2018b) provide a program for simulating stochastic SIR processes at Rizoiu et al. (2018a). It is written in the programming language

R. We have developed a Python 3 program also capable of completing this task. It can be found at **gh-meins**. This repository also contains a Jupyter notebook related to this section about stochastic SIR models. In this notebook one can see how to use our Python program to calculate the results reported here and how to generate the plots shown in this section.

Each of the figures 3.5 and 3.6 show one sample path of an SIR process. Both figures also serve as a comparison to the deterministic SIR model since the parameters used in the simulations equal those used in the figures 3.3 and 3.4, respectively.

In figure 3.5 we see that a disease-free state is reached before $t = 5$. Since this is the absorbing state, the processes S, I, and R do not change after the disease has died out. Note that I tends to decrease throughout the simulation until it reaches zero. The tendency to decline in the process I_t is in line with the monotonically decreasing (deterministic) $I(t)$ in figure 3.3 where the same model parameters were used.

Instead of reaching a disease-free state quickly, the number of infected individuals is even increasing at first in figure 3.6. Comparing figures 3.6 with 3.4 we can see again a similar behavior between the deterministic and stochastic SIR model when the same parameters are provided. To show that this similarity is not only present in the sample path shown, we have simulated 1000 paths in figure 3.7.

3.3.3. Fitting a Stochastic SIR Model

Next, we fit a stochastic SIR model as defined in subsection 3.3.1 using the Maximum Likelihood approach. To this end we first discuss the model's likelihood function.

In the CTMC model we have two observations attached to each event. The first one is the time of the event and the second one is the event's type. We consider the former by using the exponential distribution of the waiting times. We index the events chronologically, such that y_i denotes the i^{th} event ($i \in \{1, \ldots, n\}$). Event y_i occurs at time t_i and can either be an infection or a recovery. Equations (3.19), (3.20), and (3.21) imply that the intensities λ^C,

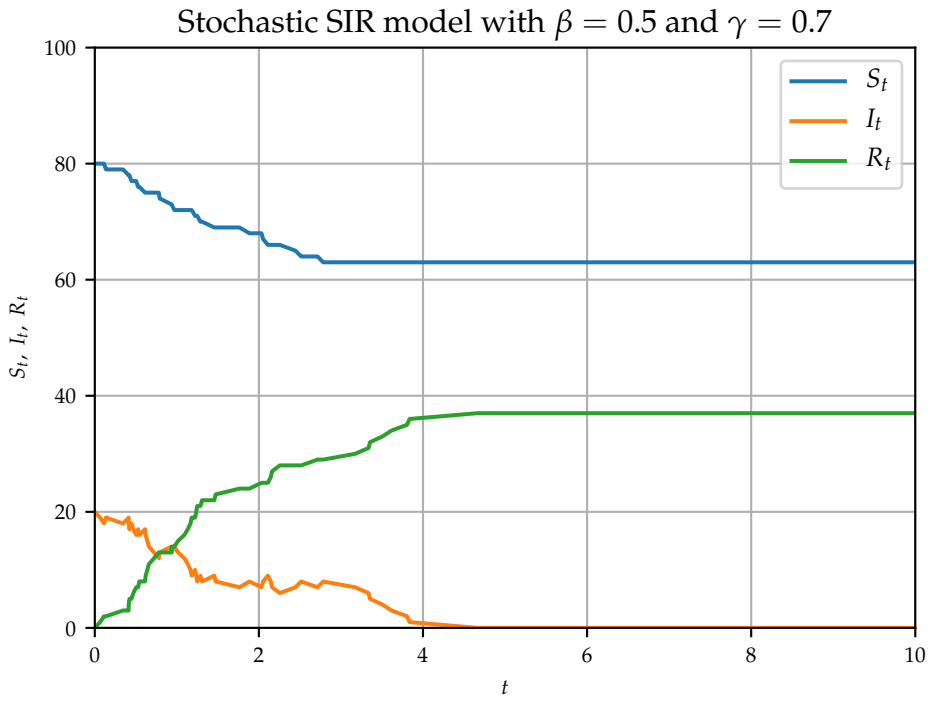


Figure 3.5.: A simulated stochastic SIR process where the absorbing state is reached before $t = 10$.

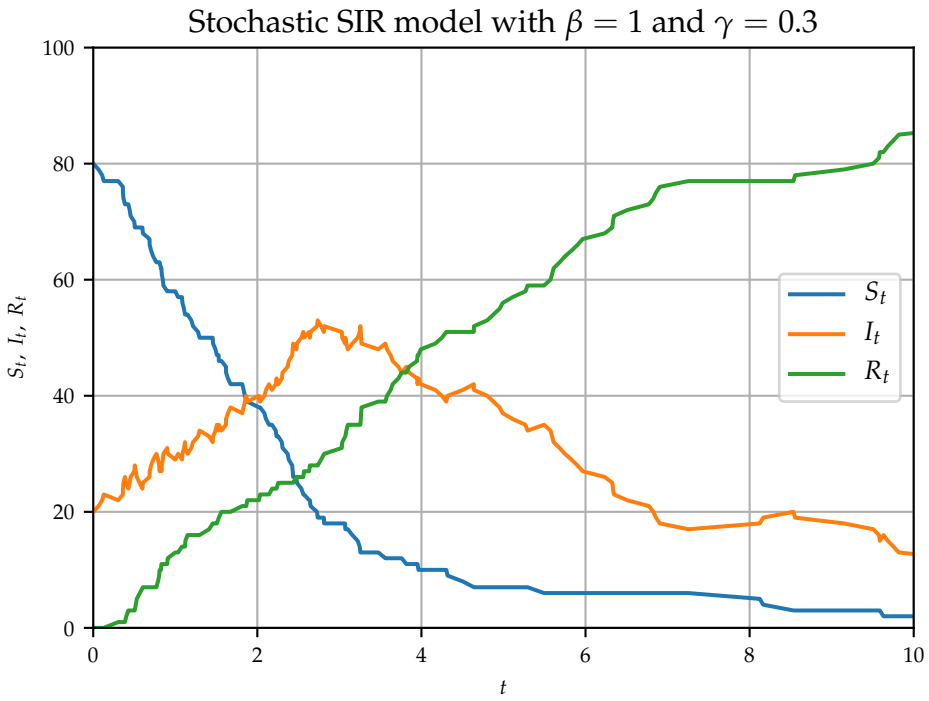


Figure 3.6.: A simulated stochastic SIR process where the absorbing state is not reached before $t = 10$.

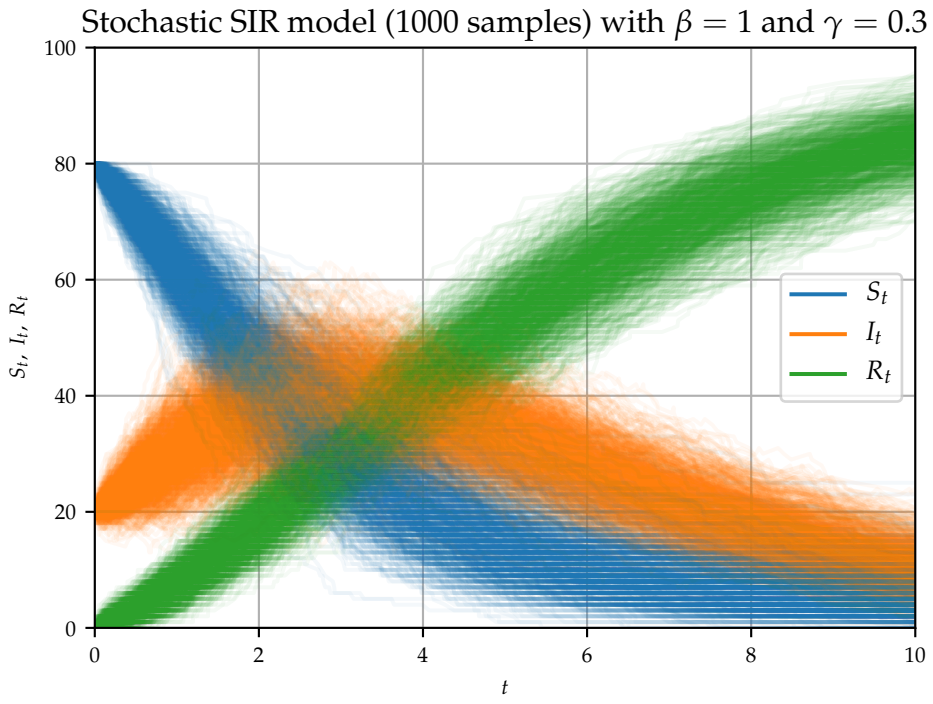


Figure 3.7.: 1000 simulated stochastic SIR processes with the same parameters as the sample in figure 3.6.

λ^R, and λ are constant between two events. We denote the intensities in $[t_i, t_{i+1})$, i.e. those resulting from y_i, by $\lambda^C(t_i)$, $\lambda^R(t_i)$, and $\lambda(t_i)$. With this notation and $\Delta t_i := t_i - t_{i-1}$ as waiting time we get $\lambda(t_{i-1})\,\mathrm{e}^{-\lambda(t_{i-1})\Delta t_i}$ as the the probability density function of Δt_i.

To compute the likelihood, we also have to incorporate the type of the observation. As discussed in 3.3.2 the probability that the next event is an infection is given by $\frac{\lambda^C(t_{i-1})}{\lambda^C(t_{i-1})+\lambda^R(t_{i-1})}$ and the probability of a recovery is $\frac{\lambda^R(t_{i-1})}{\lambda^C(t_{i-1})+\lambda^R(t_{i-1})}$.

The model's event times and types of events are independent, so their contributions to the likelihood function are multiplied. Since the events are independent as well, each observation constitutes a factor. The product of these factors represents the likelihood. With

$$\mathbb{1}_I(y_i) := \begin{cases} 1 & \dots\ y_i \text{ is an infection} \\ 0 & \dots\ y_i \text{ is a recovery} \end{cases}$$

and $\mathbb{1}_R(y_i) := 1 - \mathbb{1}_I(y_i)$ we obtain

$$\prod_{i=2}^{n} \lambda(t_{i-1})\,\mathrm{e}^{-\lambda(t_{i-1}\Delta t_i)} \left(\frac{\lambda^C(t_{i-1})}{\lambda^C(t_{i-1}) + \lambda^R(t_{i-1})} \mathbb{1}_I(y_i) + \frac{\lambda^R(t_{i-1})}{\lambda^C(t_{i-1}) + \lambda^R(t_{i-1})} \mathbb{1}_R(y_i) \right) \tag{3.22}$$

as the likelihood function of the CTMC model. Note that the first observation is neglected because it is assumed to be given. With (3.19) and (3.20) we can rewrite (3.22) and get

$$L(\beta, \gamma, N) = \prod_{i=2}^{n} \lambda(t_{i-1})\,\mathrm{e}^{-\lambda(t_{i-1}\Delta t_i)} \left(\frac{\beta S(t_{i-1})}{\beta S(t_{i-1}) + N\gamma} \mathbb{1}_I(y_i) + \frac{N\gamma}{\beta S(t_{i-1}) + N\gamma} \mathbb{1}_R(y_i) \right), \tag{3.23}$$

where $S(t_i)$ denotes the number of susceptible individuals in the interval $[t_i, t_{i+1})$. Note that the likelihood function's parameters are β, γ, and N –

in accordance with Rizoiu et al. (2018b). Note also that the formula for the likelihood in Rizoiu et al. (2018b) is incorrect. Their equation (34) contains a product of $\mathbb{1}_I(y_i)$ and $\mathbb{1}_R(y_i)$ which is always zero, regardless of the class y_i of the event because either $\mathbb{1}_I(y_i) = 0$ or $\mathbb{1}_R(y_i) = 0$ holds.

In Rizoiu et al. (2018b) N is fitted with the values of β and γ assumed to be known and thus fixed. In this setting Rizoiu et al. (2018b) show that the estimation of N works well with the medians of the estimated values for N being close to the true value of N – even if the time horizons of the simulations are limited (cf. figure 5 in Rizoiu et al. (2018b)).

We have conducted a similar analysis based on 20 simulations. To stay in line with Rizoiu et al. (2018a) – which we consider in more detail later – we used $S_0 = 1000$, $I_0 = 300$, and $R_0 = 0$ as starting values and the parameters $\beta = 1$ and $\gamma = 0.2$. The starting values imply that $N = 1300$. All of the 20 simulations were run until time $t_{max} = 11$. We fitted N for each simulation and for different time horizons $\{1, 2, \ldots, t_{max}\}$ per simulation. For each time horizon we calculated the median of the 20 estimations for N as well as an upper and a lower quantile such that there are three estimations above and below the upper and lower quantile, respectively. Our result is shown in figure 3.8. As in figure 5 in Rizoiu et al. (2018b) we can see that the median of our estimations is near the true value of N and that the difference between the upper and lower plotted quantile tends to decrease with larger time horizons, i.e. with more data.

The estimations in figure 3.8 as well as many other estimation results presented in this thesis involve the application of the L-BFGS algorithm. In our Python program we used the implementation by Zhu et al. (1997).

Although the estimation results in figure 3.8 might look promising, we have to keep in mind that we assumed that β and γ are known, which often is not realistic. This is why we now turn to the case in which all three parameters of the likelihood function are unknown. Rizoiu et al. (2018a) present their fitting procedure for this case in a Jupyter notebook. First they simulate 20 realizations of a stochastic SIR model. Each simulation starts with $S_0 = 1000$, $I_0 = 300$, and $R_0 = 0$ and has the parameters $\beta = 1$, and $\gamma = 0.2$. Then they estimate the parameters N, β, and γ. According to them this step is achieved by maximizing the log-likelihood function. They present their estimates for the parameters β, γ, and N by showing

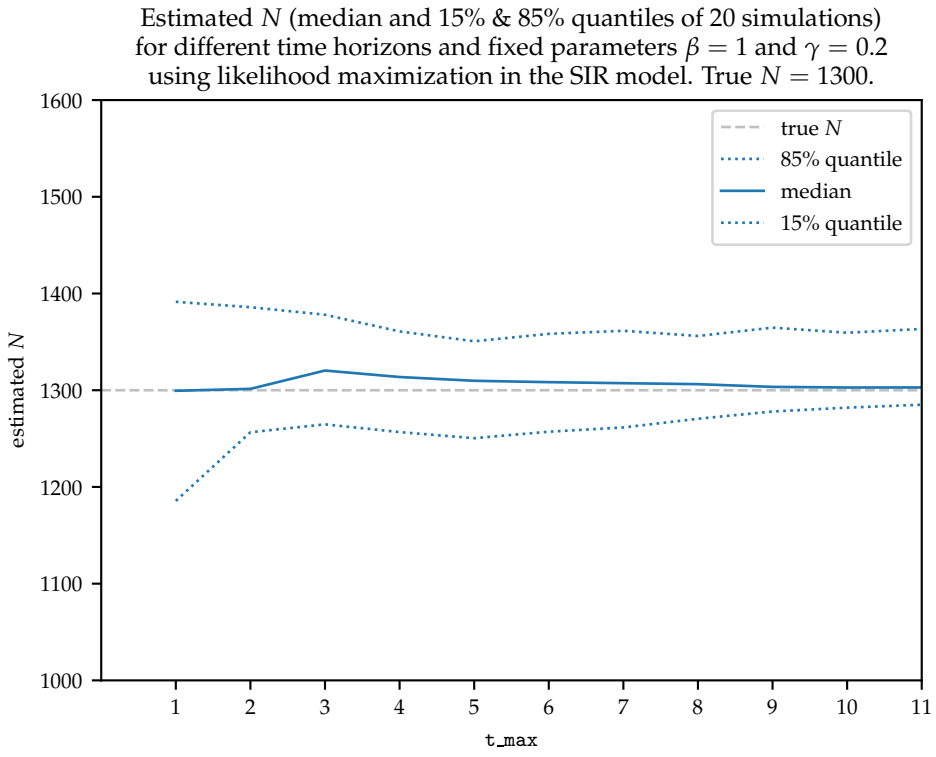


Figure 3.8.: Estimation results (median, and 15% & 85% quantiles) for N.

1. the median values as well as
2. the sample standard deviation

of the 20 experiments for each parameter. Their results can be seen in table 3.1. Note that they did not use a random seed, thus preventing the reproduction of their experiments.

	N	β	γ
theoretical	1300	1.000	0.200
median	1283.5	0.986	0.200
standard deviation	3.6	0.026	0.006

Table 3.1.: Estimation results presented by Rizoiu et al. (2018a).

While Rizoiu et al. (2018a) offer their implementation in the programming language R, we aimed at providing the equivalent procedure in form of Python 3 code. When provided with the same initial guesses as in Rizoiu et al. (2018a), i.e. $N = 1000$, $\beta = 0.1$, and $\gamma = 0.1$, our fitting procedure produces the results shown in table 3.2. Each row of this table shows the estimation results of one simulation. The table's first column represents the index numbering the 20 simulations. The other three values in each row show the estimation results for N, β, and γ in the simulation specified by the index.

Table 3.3 summarizes the estimation results of table 3.2 with median values as well as sample standard deviations. We see that our estimation results for γ are close to the true value of 0.2. However, the estimation of β does not work as desired. Instead of the true value 1 we get estimations around 0.775. And even worse, the estimation of the parameter N basically does not differ from the starting value (1000) we provided although the true value was 1300.

When we compare our results with those of Rizoiu et al. (2018a) in table 3.1 we can clearly see that our results are much worse. So why is this the case? To answer this question we compared the R code to our Python code. It turns out that the fitting procedure `fit.stochastic.sir` in R neglects the starting value for N provided to it. The first three lines of this function are the following.

	N	beta	gamma
0	1000.0	0.752	0.211
1	1000.0	0.753	0.198
2	1000.0	0.778	0.197
3	1000.0	0.810	0.194
4	1000.0	0.816	0.203
5	1000.0	0.744	0.191
6	1000.0	0.803	0.214
7	1000.0	0.771	0.198
8	1000.0	0.734	0.190
9	1000.0	0.762	0.198
10	1000.0	0.787	0.201
11	1000.0	0.808	0.199
12	1000.0	0.775	0.198
13	1000.0	0.787	0.195
14	1000.0	0.739	0.201
15	1000.0	0.811	0.204
16	1000.0	0.779	0.200
17	1000.0	0.738	0.199
18	1000.0	0.763	0.202
19	1000.0	0.776	0.195

Table 3.2.: Estimation results of our Python code.

	N	β	γ
theoretical	1300	1.000	0.200
median	1000.00011	0.77536	0.19852
standard deviation	0.00002	0.02617	0.00577

Table 3.3.: Summary of estimation results of our Python code.

```
if (params.start["N"] <  max(mysim[, -1])) {
  params.start["N"] <-  max(mysim[, -1])
}
```

In these lines of code we have a `data.frame` called `mysim` which represents one simulation. The first column of this `data.frame` consists of the event times. Thus, `mysim[, -1]` is the `data.frame` without the time column. The remaining columns represent S_t, I_t, R_t, and C_t. So the R code uses the maximum value contained in these four columns if it is greater than the starting value provided to the fitting procedure. Since in the example shown by Rizoiu et al. (2018a) – which is also used here – most individuals are either infected or recovered at the end of the observation time, the maximum value in `mysim[, -1]` is the last entry in the column corresponding to C_t. In this example, this value is already close to N and it is not changed by the L-BFGS algorithm. So the estimations of N returned by the R code (which are much better than our results in table 3.2) are not the result of the L-BFGS algorithm but rather the result from setting the estimation to the largest observed value.

For experiments with a shorter time horizon, i.e. a smaller value for `Tmax` in the R code, or with a smaller infection rate the R code would not produce estimates of similar quality because in these two cases $C_{\texttt{Tmax}}$ is smaller. This is proven in appendix B.1.

Besides, Rizoiu et al. (2018b) state in their appendix (E.1) that they use the number of events as lower bound for N. Such a bound could lead to incorrect results since each individual can become infected *and* recover and thus the number of events is not limited by N (but by $2N - 1$ as discussed in subsection 3.3.4). However, they do not use this bound in their code in Rizoiu et al. (2018a). Rizoiu et al. (2018b) mention the incorrect lower bound in connection with their equation (34) which gives a formula for the likelihood of a stochastic SIR model. As mentioned above, this formula is also incorrect since it always yields zero.

For a time horizon of 11 and an infection rate of 1 – as used in Rizoiu et al. (2018a) – tables 3.1 and 3.3 show that the R code produces better results than our Python code. This not only applies to N but to β and γ as well. Hence, we have also implemented a variation of the fitting procedure in our code

which mimics the behavior of the R code by setting the lower bound for the parameter N to the number of observed infections. The corresponding estimation result is contained in table 3.4. Again we provide a summary of this table. It can be found in table 3.5.

	N	beta	gamma
0	1274.0	0.958	0.211
1	1283.0	0.966	0.198
2	1289.0	1.003	0.197
3	1285.0	1.040	0.194
4	1282.0	1.046	0.203
5	1282.0	0.953	0.191
6	1285.0	1.032	0.214
7	1275.0	0.983	0.198
8	1287.0	0.945	0.190
9	1281.0	0.976	0.198
10	1281.0	1.008	0.201
11	1289.0	1.042	0.199
12	1287.0	0.997	0.198
13	1290.0	1.015	0.195
14	1280.0	0.946	0.201
15	1288.0	1.045	0.204
16	1287.0	1.003	0.200
17	1278.0	0.943	0.199
18	1288.0	0.983	0.202
19	1282.0	0.995	0.195

Table 3.4.: Estimation results of our Python code using better starting values for N.

Comparing tables 3.1 and 3.5, we can see that now the results produced by the R and the Python code do not differ considerably. The slight differences probably result from the fact that the 20 simulations are not equal. Note that using good starting values for N improved the estimation results not only for N but also for β which was not well estimated previously (with a starting value of $N = 1000$).

However, when we instruct our `fit` method to mimic the behavior of the

	N	β	γ
theoretical	1300.0	1.000	0.200
median	1284.0	0.99607	0.19852
standard deviation	4.637	0.03520	0.00577

Table 3.5.: Summary of estimation results of our Python code using better starting values for N.

R code (via the argument `n_start=None`), our Python code also produces worse results in case of shorter time horizons or lower infection rates – just as the R code. Using `t_max = 5` instead of `t_max = 11` yields the results shown in table 3.6. Table 3.7 shows the results if we again use a time horizon of 11 for our simulations but reduce the infection rate to 0.5.

	N	β	γ
theoretical	1300.0	1.000	0.200
median	1200.0	0.93977	0.19765
standard deviation	16.5	0.04746	0.00918

Table 3.6.: Summary of estimation results of our Python code with a shorter time horizon.

	N	β	γ
theoretical	1300.0	0.500	0.200
median	1096.0	0.42071	0.20054
standard deviation	23.3	0.02672	0.00664

Table 3.7.: Summary of estimation results of our Python code with a reduced infection rate.

When we compare tables B.1 and B.2 with tables 3.6 and 3.7, respectively, we see that our Python code again produces results that are similar to those generated by the R code. The slight differences probably stem from the simulations which differ between the two languages.

To see how several different sets of parameters affect the estimation result for N we generated the heat map in figure 3.9. In the example of Rizoiu et al.

(2018a) the ratio $\frac{\beta}{\gamma} = \frac{1}{0.2}$ equals 5. This ratio allows the disease to spread across a wide proportion of the population. In figure 3.9 we include the ratio used in Rizoiu et al. (2018a) but also consider ratios of $4, 3, 2, 1, 1/2, 1/3, 1/4$, and $1/5$. Furthermore, we decided to set the sum $\beta + \gamma$ to 1.2 in order to make β and γ unambiguous. This is in accordance with the example in Rizoiu et al. (2018a) where $\beta + \gamma = 1 + 0.2 = 1.2$. The heat map includes the estimation results for different β/γ combinations as well as for different time horizons.

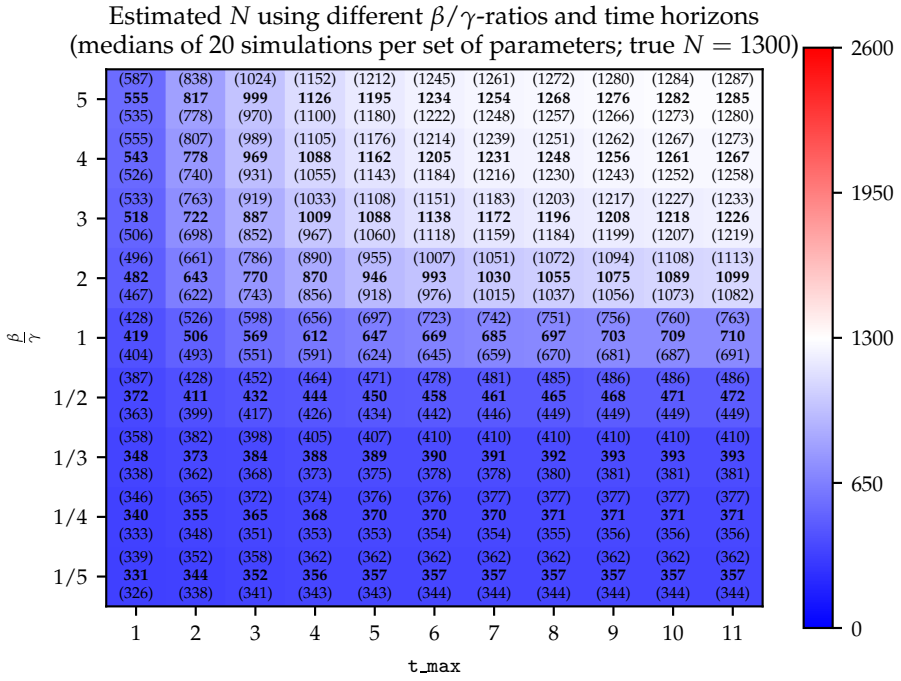


Figure 3.9.: Estimated N for different sets of parameters. In bold we have the median of estimations on 20 simulations. In parentheses above (below) we have the 85% (15%) quantile.

The heat map clearly shows that estimations of N are near the true value of 1300 only for the largest β/γ ratios and time horizons considered in figure 3.9. Other parameter combinations yield estimation results that considerably fall short of the true value. For example, $\beta = \gamma = 0.6$ and `t_max` $= 5$ lead

to an estimation that is about half of the true N. In case of $\beta = 0.3$, $\gamma = 0.9$ the estimation is less than $\frac{N}{3}$ even if the observation comprises 11 units of time.

Note that we have only used our Python program for the analysis involving the heat map. Since in some simulations the disease dies out, the R program would not be capable of calculating the heat map correctly. We prove this statement in appendix B.2.

Before we move on with the next section we compare the bad estimation results shown in the heat map in figure 3.9 with corresponding results for the case where β and γ are assumed to be known. Figure 3.10 summarizes the estimation results in this case.

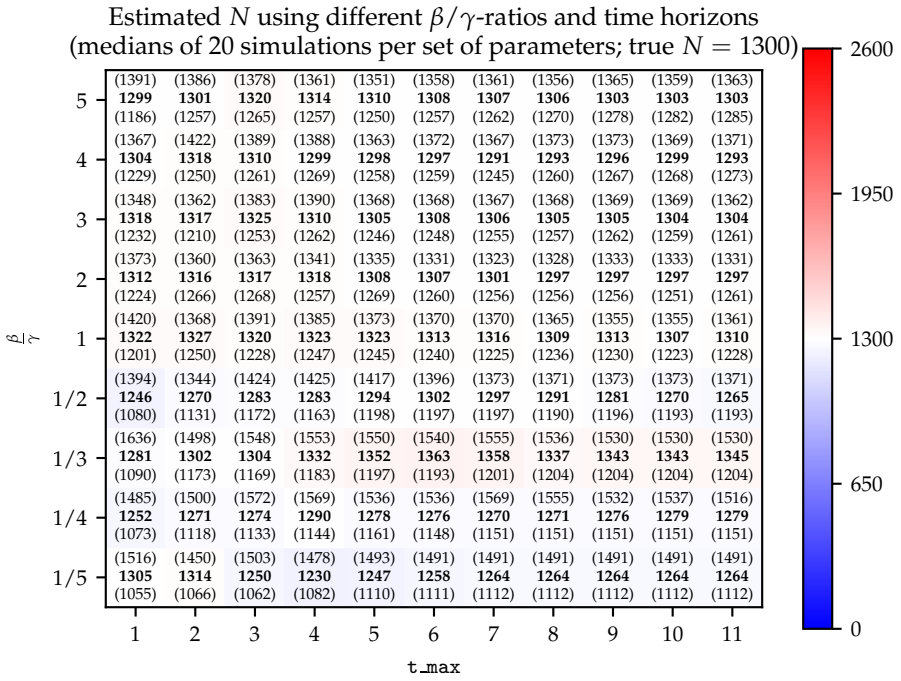


Figure 3.10.: Estimated N for different sets of parameters with known β and γ. In bold we have the median of estimations on 20 simulations. In parentheses above (below) we have the 85% (15%) quantile.

We can see that the estimation medians are now much closer to the true value of N. Now even for smaller β/γ combinations and time horizons the estimation medians are near the true value. Furthermore, there is no systematic underestimation of N when only this parameter is estimated. Larger intervals between the upper and lower quantile represent the only drawback.

3.3.4. The Final Distribution in an SIR Model

In this subsection, we express a state of the SIR model as a pair (S_t, I_t) since we do not use any properties of counting processes anymore. Let Σ be the set of possible states (S_t, I_t). By *possible* we mean that $S_t, I_t \geq 0$ and $S_t + I_t \leq N$ hold. For $N = 2$ we have

$$\Sigma = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \end{pmatrix} \right\}. \tag{3.24}$$

As discussed before, there are several possibilities for the SIR model to evolve given a state $(S_t, I_t) = (s, i)$. If the next event following time t is an infection, the system moves to the state $(s-1, i+1)$. If it is a recovery, the next state is $(s, i-1)$. If $i = 0$ holds, we are in an absorbing state that will not change after time t.

In the equations (3.16), (3.17), and (3.18) we have also looked at the probabilities for the evolution of the SIR model in a short time frame. In this subsection, we neglect the time aspect and only focus on the type of change in our variables S_t and I_t and the corresponding probabilities.

Let these probabilities be contained in a transition matrix $U = (u_{kl})_{\substack{1 \leq k \leq N \\ 1 \leq l \leq N}}$ such that u_{kl} is the probability that we move from state l to state k. If we

choose $\beta = 1$, $\gamma = 0.5$ and stick to our example of $N = 2$ we have

$$U = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 1 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.25}$$

as the transition matrix, where the rows and columns represent the states in the same ordering as shown in (3.24). Let us go through each column of our example U to gain an understanding for it.

In the first column we start in the state $(S_t, I_t) = (0,0)$. Since we are in an absorbing state ($I_t = 0$) we stay in this state with probability 1, hence the entry in the first row. In the second column we start in state $(S_t, I_t) = (0,1)$. Since there are no susceptible individuals in this state, no infection can take place. Thus, a recovery occurs next and we move to the state $(0,0)$. This explains the entry of 1 in the first row. The third column represents the state $(S_t, I_t) = (0,2)$ where again only a recovery can take place. Thus we must move to $(0,1)$ which explains the 1 in the second row. The fourth column means that we start from $(S_t, I_t) = (1,0)$. This state is absorbing and thus we have a 1 in the fourth row. The fifth column is different from the others. Here we start with $(S_t, I_t) = (1,1)$. So we can either have a recovery (leading us to $(1,0)$ (fourth row) with probability $\frac{\gamma I_t}{\beta \frac{S_t}{N} I_t + \gamma I_t} = \frac{0.5 \cdot 1}{1 \cdot \frac{1}{2} \cdot 1 + 0.5 \cdot 1} = 0.5$) or an infection (leading us to $(0,2)$ (third row) with probability $\frac{\beta \frac{S_t}{N} I_t}{\beta \frac{S_t}{N} I_t + \gamma I_t} = \frac{1 \cdot \frac{1}{2} \cdot 1}{1 \cdot \frac{1}{2} \cdot 1 + 0.5 \cdot 1} = 0.5$). The last column represents the starting state $(S_t, I_t) = (2,0)$. It is absorbing, hence we have a 1 in the last row.

The Jupyter notebook dedicated to this section in Karakaš (2019) shows how to compute Σ and U using our code. Figure 3.11 serves as a visualization of all possible states and the transition probabilities contained in the matrix U. The former are represented as dots, whereas the arrows indicate possible transitions. The number next to each arrow shows the probability of the corresponding transition.

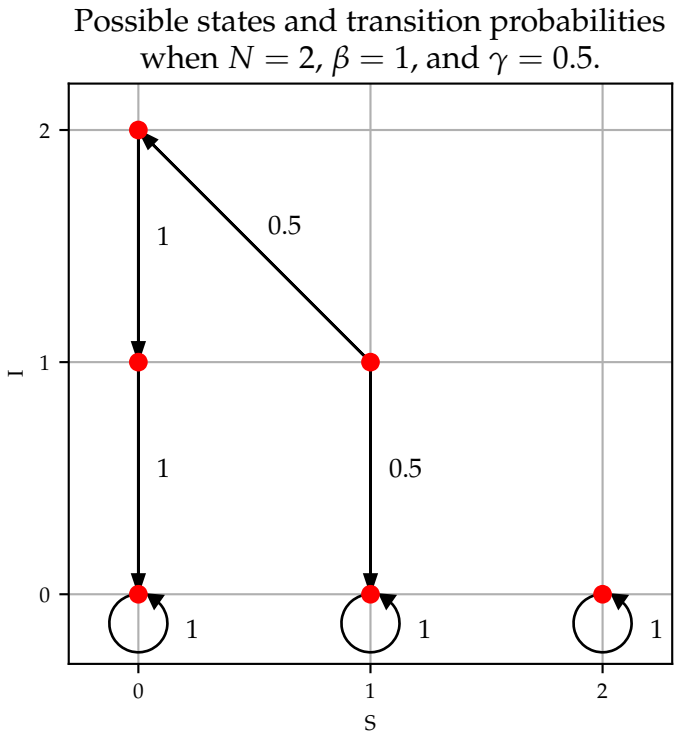


Figure 3.11.: Possible states and transitions with corresponding probabilities in an example SIR model.

To see the relation between the transition matrix U and the distribution of (S_t, I_t) after the last event has occurred, we also need a vector $\pi \in \mathbb{N}_0^{|\Sigma|}$ which represents the current state $\sigma \in \Sigma$. The vector π has one entry equal to one and all other entries are zero. The entry of 1 indicates the state we are in. If we again use our example with $N = 2$ and assume an ordering of states as in (3.24), then $\pi = (0,0,0,1,0,0)^T$ means that we are in the state $(S_t, I_t) = (1,0)$. The product $U\pi$ then gives us a new vector which for each state contains the probability that this state is reached after being in $(1,0)$. Since $U\pi = U \cdot (0,0,0,1,0,0)^T = (0,0,0,1,0,0)^T = \pi$ we stay in the same state with probability 1, which makes this state an absorbing state.

Starting with $\pi = (0,0,0,0,1,0)^T$ – which is equivalent to the state $(S_t, I_t) = (1,1)$ – is more interesting. Now the distribution for the next state is

$$U\pi = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 1 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0.5 \\ 0.5 \\ 0 \\ 0 \end{pmatrix}.$$

So we have a 50 % chance to move to the third state which is $(0,2)$ and a 50 % chance to transition to the fourth state which is $(1,0)$. The distribution for the processes S and I after one more event is $U \cdot (0,0,0.5,0.5,0,0)^T = (0,0.5,0,0.5,0,0)^T$. After one more transition we arrive at the distribution $U \cdot (0,0.5,0,0.5,0,0)^T = (0.5,0,0,0.5,0,0)^T$ which means that we get to the state $(0,0)$ with a probability of 0.5 and to $(1,0)$ with the same probability. Both states are absorbing and this is why the distribution does not change after further multiplication by U, i.e. $U \cdot (0.5,0,0,0.5,0,0)^T = (0.5,0,0,0.5,0,0)^T$.

In both examples – the one with starting vector $\pi = (0,0,0,1,0,0)^T$ and the one with $\pi = (0,0,0,0,1,0)^T$ – the sequence $(U^n\pi)_{n\in\mathbb{N}}$ converged to a *stationary distribution* after a few steps. The convergence will take place for any SIR model since there can only be $S_0 < N$ infections and N recoveries at most. So only $2N-1$ events can occur at most and thus $U^n\pi$ results in the stationary distribution for all $n \geq 2N-1$.

After the demonstration example with $N = 2$ we can again turn to the example discussed above. There we had $S_0 = 1000$, $I_0 = 300$, and $R_0 = 0$ which means that $N = 1300$. Multiplying the starting vector representing our initial state with the transition matrix from the left until we reach convergence, leads us to the stationary distribution depicted in figure 3.12.

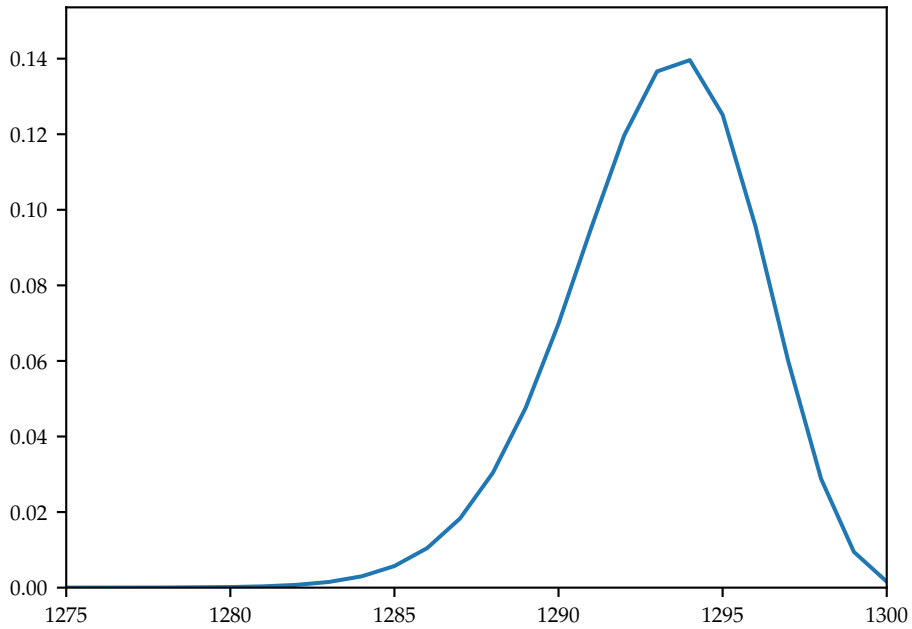


Figure 3.12.: A priori distribution of the final size of an epidemic.

The distribution shown in figure 3.12 reflects all the information available at time 0. This is why it is also called the *a priori* distribution. After observing several events we can update the vector π and recalculate the distribution of the final size of the epidemic. This is called an *a posteriori* distribution.

We plot it for five simulations in figure 3.13 with the starting state being the state reached after half of the corresponding simulation's time. The figure also contains the true final size of each simulation as a vertical line colored in the same way as the corresponding a posteriori distribution. There we

see how the updated information in π leads to distributions that are more precise, i.e. show less variance. Despite the decrease in variance all of the plotted a posteriori distributions assign positive probability mass to the true value of the final size. Also note that for each of the five simulations considered in figure 3.13 the a posteriori distribution places more probability mass on the true value than the a priori distribution.

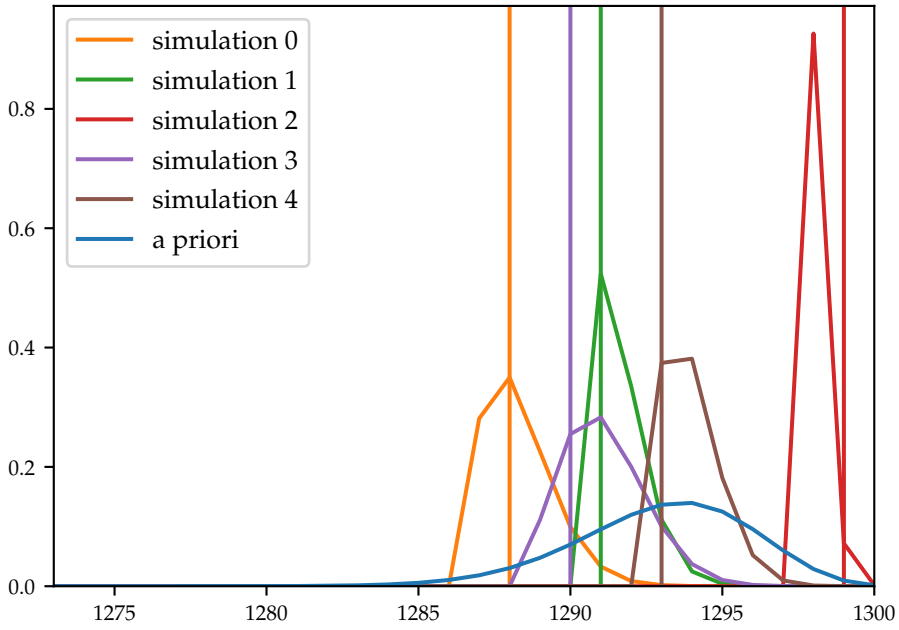


Figure 3.13.: A posteriori distributions of the final size of an epidemic modeled by a SIR model. The blue line shows the a priori distribution. The vertical lines represent the final size of the epidemic in the first five simulations. The distributions in the corresponding colors are the a posteriori distributions of the final size after observing a simulation until half of the simulation's time.

In this section we have seen that knowing an SIR model's parameters N, β, and γ as well as the iniatial number of infected individuals allows us to generate the a priori distribution of the final size of an epidemic. Observing the infection and recovery events yields further information which results in

a posteriori distributions that are more precise than the a priori distribution, thus allowing better predictions. In practice however we often lack the knowledge on the model's parameters and cannot observe recovery events. In these cases we can resort to HawkesN processes which are introduced in the next chapter.

3.4. The Deterministic SEIR model

As mentioned in the introduction the SEIR model extends the SIR model by one additional compartment. Susceptible individuals that become infected move to the new compartment E which comprises all *exposed* individuals. They are infected but are not able to infect others yet. After some time individuals from compartment E move to compartment I which means they become *infectious* or *infective*. Eventually, individuals recover, i.e. they move from I to R. The model's compartments as well as the possible transitions are summarized in figure 3.2.

As in the case of the deterministic SIR model, the transitions in the deterministic SEIR model are governed by differential equations. The evolution of the disease follows

$$\frac{dS(t)}{dt} = -\beta \frac{S(t)}{N} I(t), \tag{3.26}$$

$$\frac{dE(t)}{dt} = \beta \frac{S(t)}{N} I(t) - \sigma E(t), \tag{3.27}$$

$$\frac{dI(t)}{dt} = \sigma E(t) - \gamma I(t), \quad \text{and} \tag{3.28}$$

$$\frac{dR(t)}{dt} = \gamma I(t). \tag{3.29}$$

There are further extensions to the model in the literature, e.g. Feng et al. (2007) or Li et al. (1999).

As in De la Sen et al. (2011), we call the times an individual spends in compartment E and I the *latent* and *infective period*, respectively. Li et al. (1999) recognize that for $\sigma \to \infty$ the mean latent period goes to zero, i.e.

$1/\sigma \xrightarrow{\sigma \to \infty} 0$. This is why they argue "the SEIR model becomes a SIR model" in this case.

Examples of epidemics modeled by the SEIR model are depicted in figures 3.14 and 3.15. The former shows the case where the size of compartment I is monotonically decreasing – as in figure 3.3.

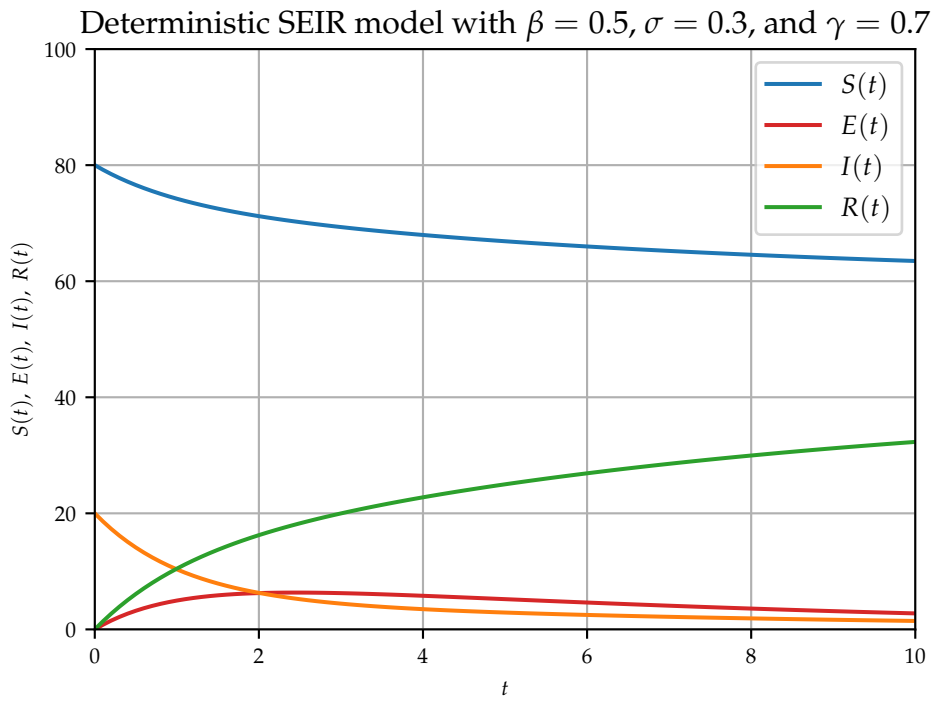


Figure 3.14.: A deterministic SEIR model with monotonically decreasing number of infected.

When we compare figure 3.15 to figure 3.4 we can see that the additional compartment in the SEIR model allows for more complicated functions $I(t)$. In the SEIR model in figure 3.15 $I(t)$ first decreases and then increases before it starts to decrease again. The code for generating the two mentioned figures resides in the Jupyter notebook corresponding to this section (see Karakaš (2019)).

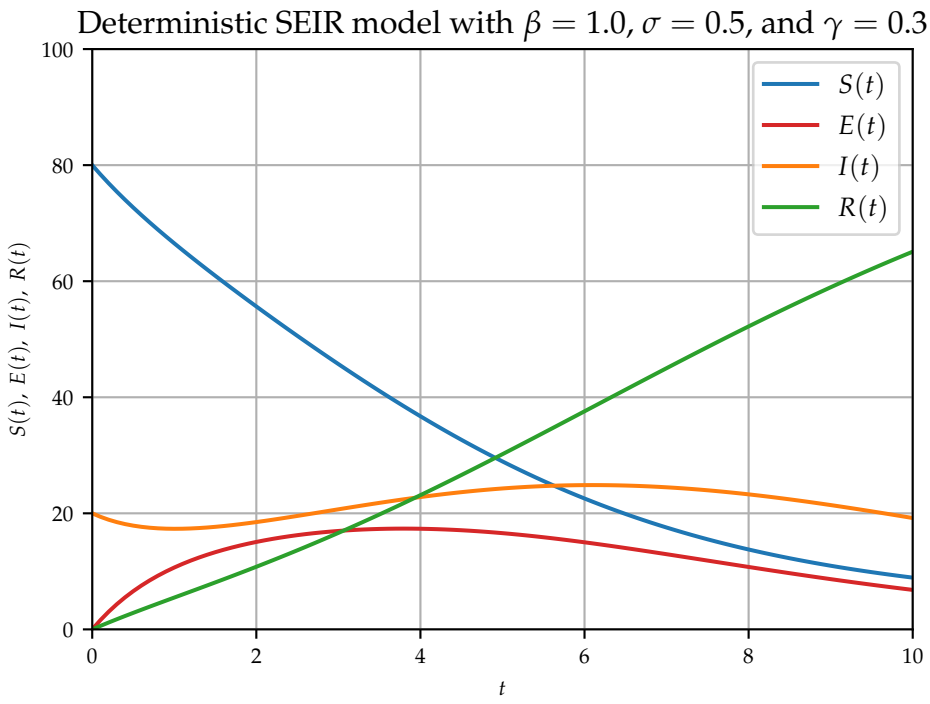


Figure 3.15.: A deterministic SEIR model where the number of infected is not monotonic.

3.5. The Stochastic SEIR Model

3.5.1. Simulation

As in the case of the SIR model we can formulate a stochastic version of the SEIR model. In subsection 3.3.1 we argued that instead of modeling all three compartments of an SIR model it suffices to focus on the two counting processes C_t and R_t. Analogously, we can focus on three counting processes in an SEIR setting. These three counting processes are

$$\begin{aligned} &B_t = E_t + I_t + R_t, \\ &C_t = I_t + R_t, \text{ and} \\ &R_t. \end{aligned} \tag{3.30}$$

Note that the latter two are the same as in our discussion of the SIR model. The three counting processes allow the reconstruction of $S_t = N - B_t$, $E_t = B_t - C_t$, and $I_t = C_t - R_t$.

Next, we formulate the transition probabilities. Since we now are considering three transitions instead of two, there is one more transition probability to compute than in the SIR model. The three dimensional counting process $(B_t, C_t, R_t) = (b, c, r)$ can now move to $(B_{t+\Delta t}, C_{t+\Delta t}, R_{t+\Delta t}) = (b + j, c + k, r + l)$ with $(j, k, l) \in \{(1,0,0), (0,1,0), (0,0,1), (0,0,0)\}$ if we assume that $\left\|(j,k,l)^T\right\|_1 = j + k + l \leq 1$. This assumption will be true for Δt sufficiently small. The first three entries in the set of possible (j, k, l) combinations correspond to the transitions from S to E, from E to I, and from I to R, respectively. The last entry, $(0, 0, 0)$, corresponds to the possibility that the state does not change in the interval $(t, t + \Delta t]$. Analogously to equations (3.16)–(3.18) we conclude from equations 3.26–3.29 that

$$p_{(b+1,c,r),(b,c,r)}(\Delta t) = \beta \frac{S_t}{N} I_t \Delta t + o(\Delta t), \tag{3.31}$$

$$p_{(b,c+1,r),(b,c,r)}(\Delta t) = \sigma E_t \Delta t + o(\Delta t), \tag{3.32}$$

$$p_{(b,c,r+1),(b,c,r)}(\Delta t) = \gamma I_t \Delta t + o(\Delta t), \text{ and} \tag{3.33}$$

$$p_{(b,c,r),(b,c,r)}(\Delta t) = 1 - \left(\beta \frac{S_t}{N} I_t + \sigma E_t + \gamma I_t\right) \Delta t + o(\Delta t) \tag{3.34}$$

holds. Here, $p_{(b+j,c+k,r+l),(b,c,r)}(\Delta t)$ is defined analogously to $p_{(c+j,r+k),(c,r)}(\Delta t)$ in equation (3.15).

From the equations (3.31)–(3.34) we can obtain the intensities of the three counting processes B_t, C_t, and R_t and analogously to the SIR model we can use these intensities to simulate an epidemic.

Each of the plots in figure 3.16 and 3.17 shows a simulation of the stochastic SEIR model. These models have the same parameters as their deterministic counterparts in the figures 3.14 and 3.15, respectively.

As in the SIR model in figure 3.5, a disease-free state is reached before time 10 in figure 3.16, i.e. $E(t) = I(t) = 0$ for $t >= 10$.

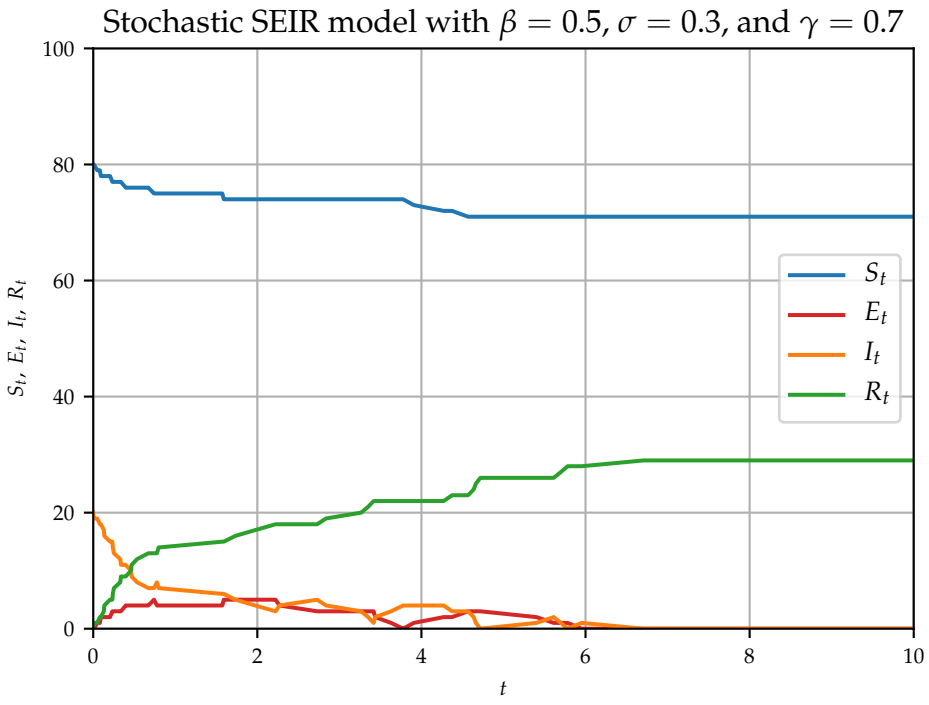


Figure 3.16.: A simulated stochastic SEIR process where the absorbing state is reached before $t = 10$.

In figure 3.17, on the other hand, the compartments E and I collectively still comprise about one fifth of the whole population at time 10.

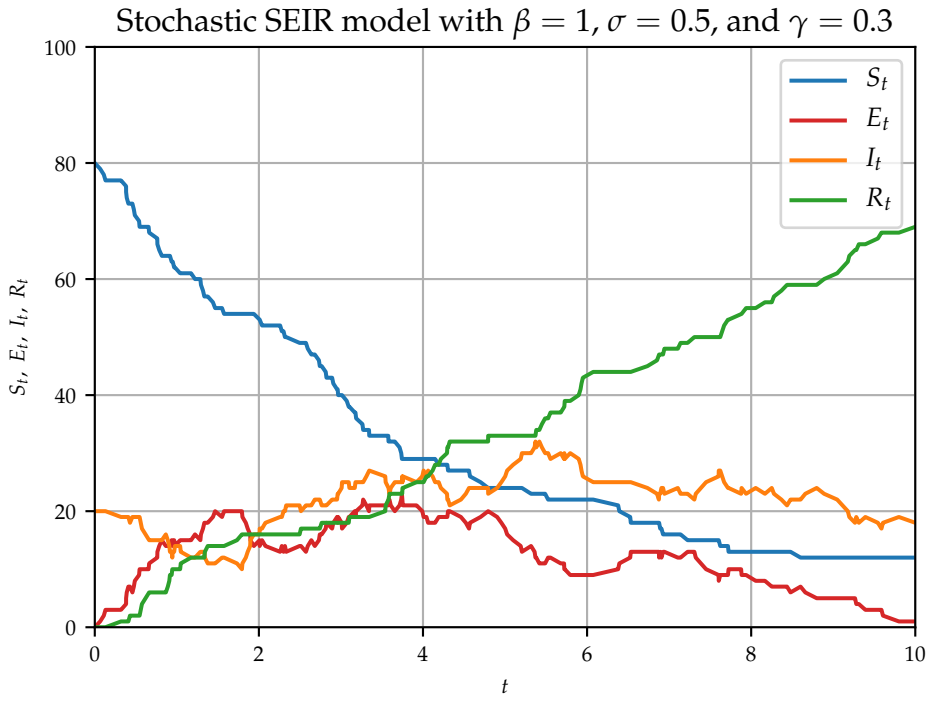


Figure 3.17.: A simulated stochastic SEIR process where the absorbing state is not reached before $t = 10$.

The code for generating the two plots can be found in the Github repository Karakaš (2019) in the Jupyter notebook for this section.

3.5.2. The Final Distribution in an SEIR Model

As in the case of the SIR model the calculation of the final distribution is based on multiplying a state vector by a transition matrix until convergence is reached. As in subsection 3.3.4 we denote the transition matrix and the set of all possible states by U and Σ, respectively.

One difference to subsection 3.3.4 is that we now have to deal with one more compartment, i.e. E, so we need one more variable to specify a state. Thus we identify a state by (S_t, E_t, I_t) instead of (S_t, I_t). This has a large impact on the set of possible states. In order to be considered possible a state now has to satisfy $S_t + E_t + I_t \leq N$. With the additional variable E_t many more combinations become possible. In the case of $N = 2$ we have

$$\Sigma = \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} \right\}.$$

As $U \in \mathbb{R}^{|\Sigma| \times |\Sigma|}$ this means that the transition matrix is much larger in the SEIR setting than in the context of SIR models.

If the i-th column of U corresponds to an absorbing state, then its i-th entry of this column is 1 and all the other entries are 0. This is the same as in the case of the SIR-related transition matrix.

Columns that do not correspond to an absorbing state can now have up to three non-zero entries – one for each of the three transitions depicted in figure 3.2. The entries can be calculated using the equations (3.31)–(3.33). So given the non-absorbing state (s, e, i) the column of U corresponding to (s, e, i) has the values $\beta \frac{s}{N} i/D$, $\sigma e/D$, and $\gamma i/D$ in the rows corresponding to $(s-1, e+1, i)$, $(s, e-1, i+1)$, and $(s, e, i-1)$, respectively, with $D = \beta \frac{s}{N} i + \sigma e + \gamma i$. All other entries are 0.

Given a starting vector π we can obtain the stationary distribution by multiplying the vector with the matrix U until the result does not change anymore. This will be the case after $3N - 1$ multiplications at the latest since each individual can move to a new compartment up to three times and at the beginning one individual already has to be either exposed or infectious, otherwise the starting state would be absorbing. So $U^n \pi$ is the final distribution for $n \geq 3N - 1$ irrespective of the state π.

If π reflects the state at time $t = 0$, then $U^n \pi$ with $n \geq 3N - 1$ is the a priori distribution. If π corresponds to an observed state after some time has passed, then $U^n \pi$ is the a posteriori distribution for $n \geq 3N - 1$.

We have generated a plot with the a priori distribution for the SEIR model with $S_0 = 350$, $E_0 = 0$, $I_0 = 105$, $R_0 = 0$, $\beta = 1$, $\sigma = 0.5$, and $\gamma = 0.3$ in

figure 3.18. This figure also contains a posteriori distributions of five simulations. These distributions are based on observing half of each simulation's time until it reaches convergence. Furthermore, for each simulation we have highlighted the actual final size of the epidemic by a vertical line in the corresponding color. We see that the a posteriori distribution tends to place more probability mass on the actual outcome. Note that, compared to figure 3.13, we have reduced the initial population by 65 % in order to compensate for the computational burden resulting from the larger set Σ in the SEIR model.

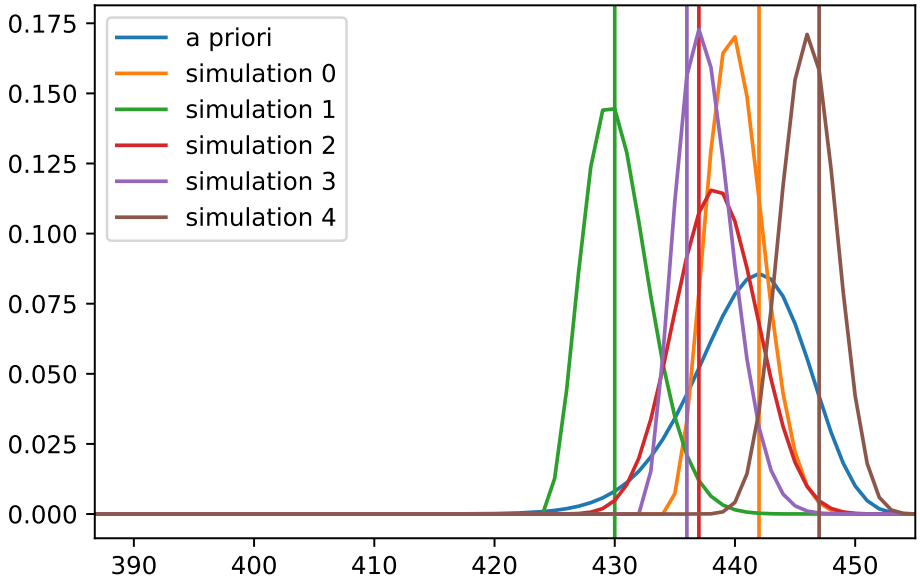


Figure 3.18.: A posteriori distributions of the final size of an epidemic modeled by a SEIR model. The blue line shows the a priori distribution. The vertical lines represent the final size of the epidemic in the first five simulations. The distributions in the corresponding colors are the a posteriori distributions of the final size after observing a simulation until half of the simulation's time.

4. The Hawkes Process

4.1. Introduction

The first goal of this chapter is the definition of the Hawkes process. To this end, we first describe the more simple Poisson process in section 4.2 and then move on to the Hawkes process in section 4.3. In section 4.4 we present the HawkesN process – a generalization of the Hawkes process – as proposed by Rizoiu et al. (2018b). Following these authors we then show the connection between the stochastic SIR and HawkesN processes in section 4.5. In section 4.6 we introduce the link between the SEIR model and the HawkesN process and show how it implies two new kernel functions, one of which generalizes the excitation function of the SIR-related HawkesN process. In sections 4.7 and 4.8 we discuss an approach for fitting the parameters of a HawkesN process. This procedure involves likelihood maximization.

4.2. The Poisson Process

The following introduction of the Poisson process is based on Klenke (2013).

Definition 4.1 ((Stationary) Poisson process)**.** A family $(N_t, t \geq 0)$ of $\mathbb{N}_0$-valued random variables is called a *(stationary) Poisson process* with intensity $\alpha \geq 0$ if $N_0 = 0$ and if:

1. For any $n \in \mathbb{N}$ and any choice of $n+1$ numbers $0 = t_0 < t_1 < \ldots < t_n$, the family $(N_{t_i} - N_{t_{i-1}}, i = 1, \ldots, n)$ is independent.
2. For $t > s \geq 0$, the difference $N_t - N_s$ is Poisson-distributed with parameter $\alpha(t-s)$; that is $N_t - N_s \sim Poi(\alpha(t-s))$.

The last condition in definition 4.1 states that

$$\forall k \in \mathbb{N}_0 : \quad \Pr[N_t - N_s = k] = \frac{(\lambda(t-s))^k}{k!} \mathrm{e}^{-\lambda(t-s)}$$

which also ensures that the process is monotonically increasing. Taking into account that also $N_0 = 0$ holds, we see that $N_t \in \mathbb{N}_0$ for all t. It can be shown that if there is a jump in the Poisson process, i.e. $\lim_{u \uparrow t} N_u \neq N_t$, then $\lim_{u \uparrow t} N_u = N_t - 1$ holds almost surely, i.e. the size of the process' jumps is 1. Daley and Vere-Jones (2003) call such processes *orderly*. This property implies that N_t represents the total count of its jumps until time t. This means that the Poisson process is a counting process as defined in definition 2.11.

From the last condition in definition 2.3 we also obtain

$$\mathbb{P}[N_{s+t} - N_s = 0] = \mathrm{e}^{-\alpha t}. \tag{4.1}$$

This resembles equation (2.1). Let us assume that at time s in (4.1) a jump occurs. Let W be the waiting time starting from s. Equation (4.1) is then equivalent to $\mathbb{P}[W > t] = \mathrm{e}^{-\alpha t}$, which means that the waiting times of a Poisson process are exponentially distributed with parameter α. Furthermore, the waiting times are independent because of the independence condition in definition 4.1. We can use this property to easily simulate a Poisson process. Figure 4.1 shows a simulated Poisson process with intensity 0.5.

Since the Poisson process is a counting process we can transform it in order to obtain a martingale. We replace λ_s in (2.7) by α to see if we get our desired martingale. Because α is constant we get

$$M_t = N_t - \int_0^t \alpha \, \mathrm{d}s = N_t - \alpha t$$

and thus

$$\mathbb{E}[M_t | \mathcal{F}_s] = \mathbb{E}\left[N_t | \mathcal{F}_s\right] - \mathbb{E}\left[\alpha t | \mathcal{F}_s\right] = N_s + \alpha(t-s) - \alpha t = N_s - \alpha s = M_s,$$

which is the martingale condition. This means that calling α the process' intensity in definition 4.1 was in line with our discussion on counting processes.

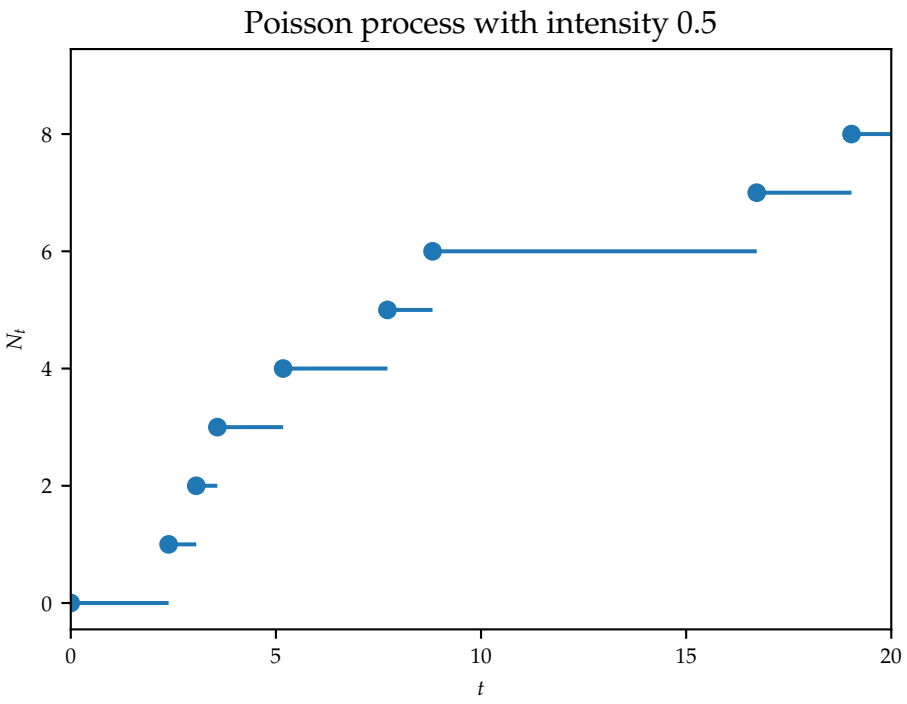


Figure 4.1.: A simulated Poisson process with intensity $\alpha = 0.5$.

One possible extension of the Poisson process is the *inhomogeneous* (sometimes also called nonhomogeneous) Poisson process. Such a process also starts with $N(0) = 0$ and it also features the independence condition of definition 4.1. However, the intensity $\alpha(t)$ in this generalization is not constant but varies deterministically over time. So instead of $N_t - N_s \sim Poi(\alpha(t-s))$ in definition 4.1 we have for the inhomogeneous Poisson process $N_t - N_s \sim Poi(\int_s^t \alpha(v)\,\mathrm{d}v)$ for $0 \leq s < t$. Daley and Vere-Jones (2003) point out that an inhomogeneous Poisson process N can be transformed into a Poisson process $\widetilde{N}$ using $u(t) := \Lambda_t = \int_0^t \alpha(v)\,\mathrm{d}v$ and setting $\widetilde{N}_t := N_{u^{-1}(t)}$.

4.3. The Hawkes Process

In this section we introduce the Hawkes Process. There are several ways to define this process. Our definition follows Laub et al. (2015).

Definition 4.2. Let $(N_t)_{t\geq 0}$ be a counting process with intensity

$$\lambda_t = \lambda + \int_0^t \mu(t-u)\,\mathrm{d}N_u\,, \tag{4.2}$$

where $\lambda > 0$ and $\mu : (0,\infty) \to [0,\infty)$ Then N is called a *Hawkes process*.

We assume that $\exists t \in \mathbb{R}_{\geq 0} : \mu(t) \neq 0$. Otherwise, we would have a Poisson process with constant intensity λ. In the context of Hawkes processes we call λ the *background intensity*.

In definition 4.2 we can rewrite the integral using definition 2.18. Denoting the time of the j^{th} jump event by t_j, we get

$$\int_0^t \mu(t-u)\,\mathrm{d}N_u = \sum_{t_j<t} \mu(t-t_j).$$

We see that whenever a jump event occurs, it increases the intensity (if $\mu(0) > 0$), making further jumps even more likely. This is why Hawkes processes are also called *self-exciting* and μ is called *excitation function*. Another

name for μ is *kernel*. Often an exponential excitation function is chosen, i.e. $\mu(t) = \kappa\theta\, e^{-\theta t}$, where κ and θ are called the scaling and the decay parameter, respectively. Both parameters are assumed to be positive such that the kernel is always positive and decreasing. The intensity of a Hawkes process with an exponential excitation function has the form of

$$\lambda_t = \lambda + \sum_{t_j < t} \kappa\theta\, e^{-\theta(t-t_j)}\,.$$

Note that Hawkes processes are different from inhomogeneous Poisson processes. While the intensity of the latter is a deterministic function, Hawkes processes feature a random intensity since it is the result of an integration with respect to a stochastic process.

4.4. The HawkesN Process

This section as well as the following ones in this chapter deal with the HawkesN process. It was defined by Rizoiu et al. (2018b) which served as the main source for the rest of this chapter. The HawkesN process was introduced in order to model actions in a community where each participant is able to act once at most. Retweet cascades on Twitter represent an example of such phenomena. To this end Rizoiu et al. (2018b) have adjusted the intensity of a Hawkes process by a factor which is intended to account for the finite population $N < \infty$ in a community like Twitter. So N represents an upper bound for the potential number of retweets. The HawkesN process $(N_t)_{t\geq 0}$ is defined as a counting process with intensity

$$\lambda_t^H = \left(1 - \frac{N_t}{N}\right)\left(\lambda + \sum_{t_j < t} \mu(t - t_j)\right). \tag{4.3}$$

We see that the intensity of a HawkesN process in equation (4.3) converges to that of a Hawkes process (see equation (4.2)) for $N \to \infty$.

When the process $(N_t)_{t\geq 0}$ reaches the population size N, the intensity λ_t^H becomes zero. From this point on, the probability of another jump in the

HawkesN process is zero which is the desired behavior for processes that count non-repeating actions of individuals.

4.5. The Relation between HawkesN and Stochastic SIR

Rizoiu et al. (2018b) show that there is a close relationship between HawkesN processes and the stochastic SIR model (more precisely, the CTMC model) discussed in section 3.3. This relationship can be explained intuitively by using again the example of Twitter.

If a retweet cascade is seen as a HawkesN process $(N_t)_{t\geq 0}$, then each retweet increments the process. This event also affects the intensity λ^H. On the one hand it has an increasing effect by adding a new addend in the sum in equation (4.3), on the other hand it has a decreasing effect by reducing the first factor, $\left(1 - \frac{N_t}{N}\right)$, in the same equation. As time passes and more and more events are registered, the new addends become smaller and smaller because they are adjusted by the decreasing first factor. So even if the increasing effect of a new event in the beginning of the observation period outweighs the decreasing effect, the latter will prevail eventually, making new retweets less likely.

Retweet cascades can also be modeled using a stochastic SIR model, where each infection represents a retweet. Given a state with $S_t \gg I_t$ a new infection (retweet) increases the probability of another retweet due to equation (3.16). As discussed in chapter 3, the infection process I_t declines after some time because of recoveries and because new infections become rarer.

So in both models we have forces that cause a retweet to generate further retweets and both models feature a mechanism that causes the retweet cascade to decelerate and stop eventually. However, there is also a key difference. While the infection in the SIR model has an equivalent in the HawkesN process, a recovery in the SIR context lacks such an equivalent. Recoveries are not observed in the HawkesN setting.

Before we discuss how to bridge this gap between the two models we need to specify the HawkesN process in more detail. Equation (4.3) does not assume any particular form of the excitation function. To see the link between our two models of interest this function has to be the exponential kernel, thus we focus on the HawkesN process with intensity

$$\lambda_t^H = \left(1 - \frac{N_t}{N}\right)\left(\lambda + \sum_{t_j^C < t} \kappa\theta \, \mathrm{e}^{-\theta(t - t_j^C)}\right). \tag{4.4}$$

Note that in this equation and in the rest of this thesis t_j^C denotes the j^{th} jump to underline the connection to the SIR model.

Now we can present the relationship between the two models in a formal way. It is the main result in Rizoiu et al. (2018b) and forms their theorem 3.1.

Theorem 4.3. *Let $(C_t)_{t\geq 0}$ be a counting process which counts the infection events in a stochastic SIR model and let the corresponding intensity be $(\lambda_t^C)_{t\geq 0}$. The parameters of the stochastic SIR model are N, β, and γ. Let λ, κ, θ, and (again) N be the parameters of a HawkesN process $(N_t)_{t\geq 0}$ with an intensity $(\lambda_t^H)_{t\geq 0}$ of the same form as in equation (4.4). As in equation (3.14) we denote the duration the j^{th} individual's infection by τ_j. Let $\mathcal{T} := \{\tau_1, \tau_2, \ldots\}$ be the set of all individuals' time to recovery. If $\beta = \kappa\theta$, $\gamma = \theta$, and the HawkesN process' background intensity $\lambda = 0$ then*

$$\mathbb{E}_{\mathcal{T}}\left[\lambda_t^C\right] = \lambda_t^H \tag{4.5}$$

holds for all but finitely many points t.

Proof. Since there is no background intensity, i.e. $\lambda = 0$, the right side of equation (4.5) is

$$\lambda_t^H = \left(1 - \frac{N_t}{N}\right) \sum_{t_j^C < t} \kappa\theta \, \mathrm{e}^{-\theta(t - t_j^C)}. \tag{4.6}$$

Now we start with the left side of the equation and work our way towards the right side. The left side is

$$\begin{aligned}\mathbb{E}_{\mathcal{T}}\left[\lambda_t^C\right] &= \mathbb{E}_{\mathcal{T}}\left[\beta\frac{S_t}{N}I_t\right] \\ &= \mathbb{E}_{\mathcal{T}}\left[\beta\frac{S_t}{N}\sum_{t_j^C\le t}\mathbb{1}_{(t,\infty)}\left(t_j^C+\tau_j\right)\right] \\ &= \sum_{t_j^C\le t}\mathbb{E}_{\mathcal{T}}\left[\beta\frac{S_t}{N}\mathbb{1}_{(t,\infty)}\left(t_j^C+\tau_j\right)\right].\end{aligned}$$

For all but finitely many points in time we can replace the $\le$ sign by $<$ and write

$$\begin{aligned}\mathbb{E}_{\mathcal{T}}\left[\lambda_t^C\right] &= \sum_{t_j^C\le t}\mathbb{E}_{\mathcal{T}}\left[\beta\frac{S_t}{N}\mathbb{1}_{(t,\infty)}\left(t_j^C+\tau_j\right)\right] \\ &= \sum_{t_j^C< t}\mathbb{E}_{\mathcal{T}}\left[\beta\frac{S_t}{N}\mathbb{1}_{(t,\infty)}\left(t_j^C+\tau_j\right)\right] \quad \forall t\in\mathbb{R}\setminus\{t_j^C|j\in\{1,2,\dots,N\}\}.\end{aligned}$$

Due to equation (3.16) the waiting time τ_j is exponentially distributed with parameter γ. Thus, the probability density function of τ_j is $\gamma\,\mathrm{e}^{-\gamma\tau_j}$, which leads to

$$\begin{aligned}\mathbb{E}_{\mathcal{T}}\left[\lambda_t^C\right] &= \sum_{t_j^C< t}\mathbb{E}_{\mathcal{T}}\left[\beta\frac{S_t}{N}\mathbb{1}_{(t,\infty)}\left(t_j^C+\tau_j\right)\right] \\ &= \sum_{t_j^C< t}\int_0^\infty \beta\frac{S_t}{N}\mathbb{1}_{(t,\infty)}\left(t_j^C+x\right)\gamma\,\mathrm{e}^{-\gamma x}\,\mathrm{d}x \\ &= \sum_{t_j^C< t}\beta\frac{S_t}{N}\int_{t-t_j^C}^\infty \gamma\,\mathrm{e}^{-\gamma x}\,\mathrm{d}x \quad \forall t\in\mathbb{R}\setminus\{t_j^C|j\in\{1,2,\dots,N\}\}.\end{aligned}$$

The integral represents the tail of an exponential distribution. We have derived its formula which can be seen in equation (2.1). Thus, we have

$$\begin{aligned}\mathbb{E}_{\mathcal{T}}\left[\lambda_t^C\right] &= \sum_{t_j^C<t} \beta\frac{S_t}{N}\int_{t-t_j^C}^{\infty} \gamma\, \mathrm{e}^{-\gamma x}\,\mathrm{d}x \\ &= \sum_{t_j^C<t} \beta\frac{S_t}{N}\,\mathrm{e}^{-\gamma(t-t_j^C)} \quad \forall t \in \mathbb{R}\setminus\{t_j^C | j \in \{1,2,\ldots,N\}\}.\end{aligned}$$

Using equation (3.10) we get

$$\begin{aligned}\mathbb{E}_{\mathcal{T}}\left[\lambda_t^C\right] &= \sum_{t_j^C<t} \beta\frac{S_t}{N}\,\mathrm{e}^{-\gamma(t-t_j^C)} \\ &= \sum_{t_j^C<t} \beta\frac{N-C_t}{N}\,\mathrm{e}^{-\gamma(t-t_j^C)} \\ &= \left(1-\frac{C_t}{N}\right)\sum_{t_j^C<t} \beta\,\mathrm{e}^{-\gamma(t-t_j^C)} \quad \forall t \in \mathbb{R}\setminus\{t_j^C | j \in \{1,2,\ldots,N\}\}.\end{aligned}$$

Translating this result into the notation of HawkesN processes (C_t of the SIR model corresponds to N_t) and using the theorem's assumptions we obtain

$$\begin{aligned}\mathbb{E}_{\mathcal{T}}\left[\lambda_t^C\right] &= \left(1-\frac{C_t}{N}\right)\sum_{t_j^C<t} \beta\,\mathrm{e}^{-\gamma(t-t_j^C)} \\ &= \left(1-\frac{N_t}{N}\right)\sum_{t_j^C<t} \beta\,\mathrm{e}^{-\gamma(t-t_j^C)} \\ &= \left(1-\frac{N_t}{N}\right)\sum_{t_j^C<t} \kappa\theta\,\mathrm{e}^{-\theta(t-t_j^C)} = \lambda_t^H \quad \forall t \in \mathbb{R}\setminus\{t_j^C | j \in \{1,2,\ldots,N\}\}\end{aligned} \tag{4.7}$$

which confirms that equation (4.5) holds. □

Now that we have derived the intensity of the HawkesN process corresponding to an SIR model, we can visualize it. An example is shown in figure 4.2.

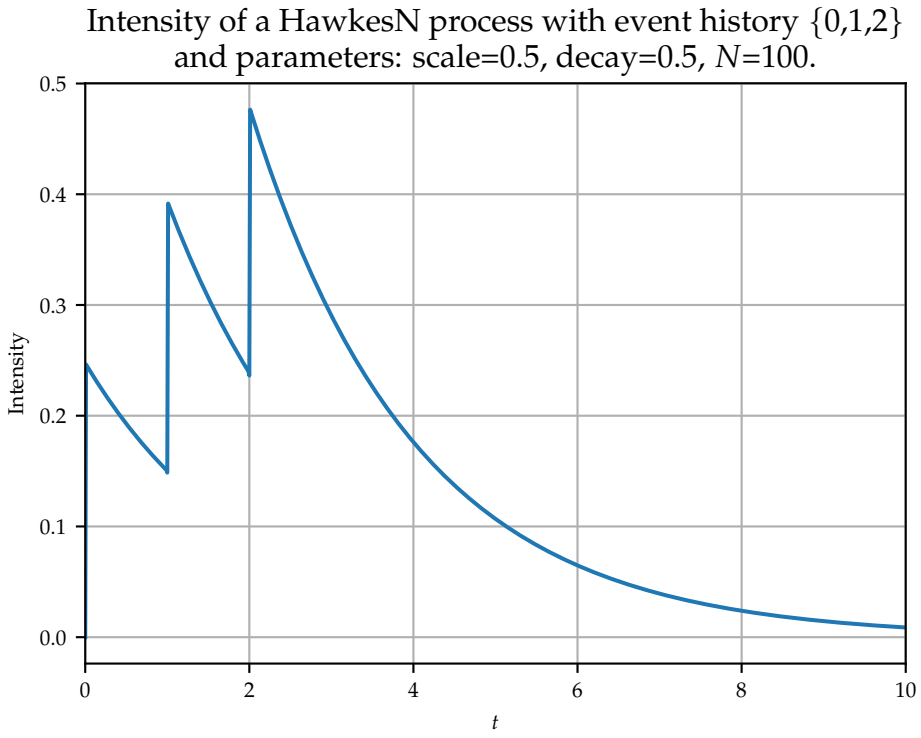


Figure 4.2.: Intensity of an example HawkesN Process calculated with our Python code.

4.6. The Relation between HawkesN and Stochastic SEIR

The HawkesN processes and the connection to the SIR model was developed in order to model information diffusion in communities like Twitter. In

this section, we are about to generalize this link to the SEIR model. Such a framework could prove useful if one wants to model information diffusion in moderated communities. In such communities a post by one user can trigger reactions by other users that are not public immediately. These other users are then considered exposed. In other words, they have a latent infection. After the moderator has approved a reaction post and made it public, the corresponding user moves from compartment E to I. Thus such a model could be used to analyze e.g. moderated forums or citation networks where papers are published after an approval process.

Rizoiu et al. (2018b) have formulated their HawkesN process using an exponential kernel. They consider other kernels as future work. In this thesis we derive a generalization of the exponential kernel. We show that a HawkesN process with the generalized excitation function still corresponds to an epidemic model – namely the SEIR model. For SEIR models where the parameters σ and γ are unequal, theorem (4.4) shows the link to a HawkesN process.

Theorem 4.4. *Let $(B_t)_{t\geq 0}$ be the number of individuals in the compartments E, I, and R at time t – as defined in (3.30). This means that B_t is a counting process which counts the transitions from S to E in a stochastic SEIR model. Let the corresponding intensity be $(\lambda_t^B)_{t\geq 0}$. The parameters of the stochastic SEIR model are N, β, σ, and γ. We assume $\sigma \neq \gamma$. Let $(N_t)_{t\geq 0}$ be a HawkesN process with intensity in the form of*

$$\lambda_t^H = \left(1 - \frac{N_t}{N}\right) \sum_{t_j^B < t} \beta \frac{\sigma}{\gamma - \sigma} \left(\mathrm{e}^{-\sigma(t-t_j^B)} - \mathrm{e}^{-\gamma(t-t_j^B)}\right). \tag{4.8}$$

We denote the duration the j^{th} individual's status as exposed (infectious) by τ_j^E (τ_j^I). Let $\mathcal{T} := \{\tau_1^E, \tau_1^I, \tau_2^E, \tau_2^I, \ldots\}$ be the set of all individuals' times in compartment E and I. Then

$$\mathbb{E}_{\mathcal{T}}\left[\lambda_t^B\right] = \lambda_t^H \tag{4.9}$$

holds for all but finitely many points t.

Proof. As in the proof of theorem 4.3 we work our way from the left to the

right side of the equation we want to prove. The equation

$$\begin{aligned}\mathbb{E}_{\mathcal{T}}\left[\lambda_t^B\right] &= \mathbb{E}_{\mathcal{T}}\left[\beta\frac{S_t}{N}I_t\right] \\ &= \mathbb{E}_{\mathcal{T}}\left[\beta\frac{N-B_t}{N}I_t\right] \\ &= \mathbb{E}_{\mathcal{T}}\left[\beta\left(1-\frac{B_t}{N}\right)I_t\right]\end{aligned}$$

also holds in the SEIR model. Since B_t is observed we may write

$$\mathbb{E}_{\mathcal{T}}\left[\lambda_t^B\right] = \beta\left(1-\frac{B_t}{N}\right)\mathbb{E}_{\mathcal{T}}\left[I_t\right], \tag{4.10}$$

so again we are interested in the expectation of I_t. In contrast to the SIR model the transition observed in the HawkesN process corresponding to the SEIR model is from S to E instead of S to I. This means that we observe the times of this transition and we denote the transition time for the j^{th} individual by t_j^B. This time together with the random variables τ_j^E and τ_j^I allows us to express I_t as $\sum_{t_j^B \le t} \mathbb{1}_{[0,t)}\left(t_j^B+\tau_j^E\right)\cdot\mathbb{1}_{(t,\infty)}\left(t_j^B+\tau_j^E+\tau_j^I\right)$. Thus we obtain

$$\begin{aligned}\mathbb{E}_{\mathcal{T}}\left[I_t\right] &= \mathbb{E}_{\mathcal{T}}\left[\sum_{t_j^B\le t}\mathbb{1}_{[0,t)}\left(t_j^B+\tau_j^E\right)\mathbb{1}_{(t,\infty)}\left(t_j^B+\tau_j^E+\tau_j^I\right)\right] \\ &= \sum_{t_j^B\le t}\mathbb{E}_{\mathcal{T}}\left[\mathbb{1}_{[0,t)}\left(t_j^B+\tau_j^E\right)\mathbb{1}_{(t,\infty)}\left(t_j^B+\tau_j^E+\tau_j^I\right)\right].\end{aligned}$$

Since τ_j^E and τ_j^I are exponentially distributed with parameters σ and γ,

respectively, we can rewrite the expectation as

$$\begin{aligned}
\mathbb{E}_{\mathcal{T}}\left[I_t\right] &= \sum_{t_j^B \le t} \mathbb{E}_{\mathcal{T}}\left[\mathbb{1}_{[0,t)}\left(t_j^B + \tau_j^E\right)\mathbb{1}_{(t,\infty)}\left(t_j^B + \tau_j^E + \tau_j^I\right)\right] \\
&= \sum_{t_j^B \le t} \int_0^\infty \int_0^\infty \mathbb{1}_{[0,t)}\left(t_j^B + x\right)\mathbb{1}_{(t,\infty)}\left(t_j^B + x + y\right)\sigma\,\mathrm{e}^{-\sigma x}\,\gamma\,\mathrm{e}^{-\gamma y}\,\mathrm{d}y\,\mathrm{d}x \\
&= \sum_{t_j^B \le t} \int_0^\infty \mathbb{1}_{[0,t)}\left(t_j^B + x\right)\int_0^\infty \mathbb{1}_{(t,\infty)}\left(t_j^B + x + y\right)\gamma\,\mathrm{e}^{-\gamma y}\,\mathrm{d}y\,\sigma\,\mathrm{e}^{-\sigma x}\,\mathrm{d}x
\end{aligned} \tag{4.11}$$

We can calculate the inner integral which results in

$$\begin{aligned}
\int_0^\infty \mathbb{1}_{(t,\infty)}\left(t_j^B + x + y\right)\gamma\,\mathrm{e}^{-\gamma y}\,\mathrm{d}y &= \int_{t-t_j^B-x}^\infty \gamma\,\mathrm{e}^{-\gamma y}\,\mathrm{d}y \\
&= -\int_{-\gamma(t-t_j^B-x)}^{-\infty} \mathrm{e}^{u}\,\mathrm{d}u \\
&= \int_{-\infty}^{-\gamma(t-t_j^B-x)} \mathrm{e}^{u}\,\mathrm{d}u \\
&= \mathrm{e}^{-\gamma(t-t_j^B-x)}\,.
\end{aligned}$$

Plugging this result into equation (4.11) yields

$$\begin{aligned}
\mathbb{E}_{\mathcal{T}}\left[I_t\right] &= \sum_{t_j^B \le t} \int_0^\infty \mathbb{1}_{[0,t)}\left(t_j^B + x\right)\int_0^\infty \mathbb{1}_{(t,\infty)}\left(t_j^B + x + y\right)\gamma\,\mathrm{e}^{-\gamma y}\,\mathrm{d}y\,\sigma\,\mathrm{e}^{-\sigma x}\,\mathrm{d}x \\
&= \sum_{t_j^B \le t} \int_0^\infty \mathbb{1}_{[0,t)}\left(t_j^B + x\right)\mathrm{e}^{-\gamma(t-t_j^B-x)}\,\sigma\,\mathrm{e}^{-\sigma x}\,\mathrm{d}x \\
&= \sum_{t_j^B \le t} \int_0^{t-t_j^B} \mathrm{e}^{-\gamma(t-t_j^B-x)}\,\sigma\,\mathrm{e}^{-\sigma x}\,\mathrm{d}x \\
&= \sigma \sum_{t_j^B \le t} \mathrm{e}^{-\gamma(t-t_j^B)} \int_0^{t-t_j^B} \mathrm{e}^{(\gamma-\sigma)x}\,\mathrm{d}x\,.
\end{aligned}$$

The integral in this equation is

$$\begin{aligned}\int_0^{t-t_j^B} \mathrm{e}^{(\gamma-\sigma)x}\,\mathrm{d}x &= \frac{1}{\gamma-\sigma}\int_0^{(\gamma-\sigma)(t-t_j^B)} \mathrm{e}^{u}\,\mathrm{d}u \\ &= \frac{1}{\gamma-\sigma}\left(\mathrm{e}^{(\gamma-\sigma)(t-t_j^B)}-1\right)\end{aligned}$$

which gives us

$$\begin{aligned}\mathbb{E}_{\mathcal{T}}\left[I_t\right] &= \sigma\sum_{t_j^B\le t}\mathrm{e}^{-\gamma(t-t_j^B)}\int_0^{t-t_j^B}\mathrm{e}^{(\gamma-\sigma)x}\,\mathrm{d}x \\ &= \sigma\sum_{t_j^B\le t}\mathrm{e}^{-\gamma(t-t_j^B)}\frac{1}{\gamma-\sigma}\left(\mathrm{e}^{(\gamma-\sigma)(t-t_j^B)}-1\right) \\ &= \frac{\sigma}{\gamma-\sigma}\sum_{t_j^B\le t}\left(\mathrm{e}^{-\sigma(t-t_j^B)}-\mathrm{e}^{-\gamma(t-t_j^B)}\right).\end{aligned}$$

For all but finitely many points in time we get the relation

$$\begin{aligned}\mathbb{E}_{\mathcal{T}}\left[I_t\right] &= \frac{\sigma}{\gamma-\sigma}\sum_{t_j^B\le t}\left(\mathrm{e}^{-\sigma(t-t_j^B)}-\mathrm{e}^{-\gamma(t-t_j^B)}\right) \\ &= \frac{\sigma}{\gamma-\sigma}\sum_{t_j^B< t}\left(\mathrm{e}^{-\sigma(t-t_j^B)}-\mathrm{e}^{-\gamma(t-t_j^B)}\right) \quad \forall t\in\mathbb{R}\setminus\{t_j^B|j\in\{1,2,\ldots,N\}\}.\end{aligned}$$

This – in connection with (4.10) and the fact that B_t in the SEIR model corresponds to N_t in the HawkesN process – concludes our proof. □

Recall that according to Li et al. (1999) the deterministic SEIR model converges to the SIR model with $\sigma\to\infty$. This relation extends to the intensities (4.7) and (4.8) of HawkesN processes. To show this we denote the HawkesN intensities corresponding to the SIR and the SEIR model by $\lambda_t^{H,SIR}$ and $\lambda_t^{H,SEIR}$, respectively. From equation (4.7) we know

$$\lambda_t^{H,SIR} = \left(1-\frac{N_t}{N}\right)\sum_{t_j^C\le t}\beta\,\mathrm{e}^{-\gamma(t-t_j^C)}.$$

Since τ_j^E is exponentially distributed with parameter σ, we have $\mathbb{E}\left[\tau_j^E\right] \xrightarrow{\sigma\to\infty} 0$. So in the limit we have $\tau_j^E = 0$ almost surely which is equivalent to $t_j^B = t_j^C$ almost surely. In other words, the transition from E to I happens immediately. Furthermore, $\frac{\sigma}{\gamma-\sigma} \xrightarrow{\sigma\to\infty} -1$ and $\mathrm{e}^{-\sigma(t-t_j^B)} \xrightarrow{\sigma\to\infty} 0$ for $t > t_j^B$. Thus we have

$$\begin{aligned}\lambda_t^{H,SEIR} &= \left(1-\frac{N_t}{N}\right)\sum_{t_j^B<t}\beta\frac{\sigma}{\gamma-\sigma}\left(\mathrm{e}^{-\sigma(t-t_j^B)}-\mathrm{e}^{-\gamma(t-t_j^B)}\right)\\ &\xrightarrow{\sigma\to\infty}\left(1-\frac{N_t}{N}\right)\sum_{t_j^C\le t}\beta\,\mathrm{e}^{-\gamma(t-t_j^C)} = \lambda_t^{H,SIR}.\end{aligned}$$

which means that the relation between an SIR and an SEIR model is translated to the corresponding HawkesN processes.

Theorem 4.4 does not consider the case $\sigma = \gamma$. And in fact, the theorem's intensity in (4.8) is not defined in this case. It turns out that for SEIR models with $\sigma = \gamma$ we can also find a corresponding HawkesN process by defining its kernel function accordingly. This is shown in the following theorem.

Theorem 4.5. *Let $(B_t)_{t\ge0}$ be the number of individuals in the compartments E, I, and R at time t. This means that B_t is a counting process which counts the transitions from S to E in a stochastic* SEIR *model. Let the corresponding intensity be $(\lambda_t^B)_{t\ge0}$. The parameters of the stochastic* SEIR *model are N, β, σ, and γ. We assume $\sigma = \gamma$. Let $(N_t)_{t\ge0}$ be a HawkesN process with intensity in the form of*

$$\lambda_t^H = \left(1-\frac{N_t}{N}\right)\sum_{t_j^B<t}\beta\gamma(t-t_j^B)\,\mathrm{e}^{-\gamma(t-t_j^B)}. \tag{4.12}$$

We denote the duration the j^{th} individual's status as exposed (infectious) by τ_j^E (τ_j^I). Let $\mathcal{T} := \{\tau_1^E, \tau_1^I, \tau_2^E, \tau_2^I, \ldots\}$ be the set of all individuals' times in compartment E and I. Then

$$\mathbb{E}_{\mathcal{T}}\left[\lambda_t^B\right] = \lambda_t^H \tag{4.13}$$

holds for all but finitely many points t.

Proof. As in shown in the proof of theorem 4.4 we have

$$\mathbb{E}_{\mathcal{T}}\left[\lambda_t^B\right] = \beta\left(1 - \frac{B_t}{N}\right)\mathbb{E}_{\mathcal{T}}\left[I_t\right], \tag{4.14}$$

and from (4.11) we know that

$$\mathbb{E}_{\mathcal{T}}\left[I_t\right] = \sum_{t_j^B \le t} \int_0^\infty \mathbb{1}_{[0,t)}\left(t_j^B + x\right) \int_0^\infty \mathbb{1}_{(t,\infty)}\left(t_j^B + x + y\right) \gamma\, \mathrm{e}^{-\gamma y}\, \mathrm{d}y\, \sigma\, \mathrm{e}^{-\sigma x}\, \mathrm{d}x$$

holds. Since now our assumption is $\sigma = \gamma$ we can rewrite this to

$$\mathbb{E}_{\mathcal{T}}\left[I_t\right] = \sum_{t_j^B \le t} \int_0^\infty \mathbb{1}_{[0,t)}\left(t_j^B + x\right) \int_0^\infty \mathbb{1}_{(t,\infty)}\left(t_j^B + x + y\right) \gamma\, \mathrm{e}^{-\gamma y}\, \mathrm{d}y\, \gamma\, \mathrm{e}^{-\gamma x}\, \mathrm{d}x\,.$$

As shown in the proof of theorem 4.4 the inner integral equals $\mathrm{e}^{-\gamma(t - t_j^B - x)}$. This implies

$$\begin{aligned}
\mathbb{E}_{\mathcal{T}}\left[I_t\right] &= \sum_{t_j^B \le t} \int_0^\infty \mathbb{1}_{[0,t)}\left(t_j^B + x\right) \mathrm{e}^{-\gamma(t - t_j^B - x)}\, \gamma\, \mathrm{e}^{-\gamma x}\, \mathrm{d}x \\
&= \sum_{t_j^B \le t} \int_0^{t - t_j^B} \mathrm{e}^{-\gamma(t - t_j^B - x)}\, \gamma\, \mathrm{e}^{-\gamma x}\, \mathrm{d}x \\
&= \gamma \sum_{t_j^B \le t} \mathrm{e}^{-\gamma(t - t_j^B)} \int_0^{t - t_j^B} \mathrm{d}x \\
&= \gamma \sum_{t_j^B \le t} \mathrm{e}^{-\gamma(t - t_j^B)}(t - t_j^B).
\end{aligned}$$

For points in time where no transition from S to E occurs we have

$$\begin{aligned}
\mathbb{E}_{\mathcal{T}}\left[I_t\right] &= \gamma \sum_{t_j^B \le t} \mathrm{e}^{-\gamma(t - t_j^B)}(t - t_j^B) \\
&= \gamma \sum_{t_j^B < t} \mathrm{e}^{-\gamma(t - t_j^B)}(t - t_j^B) \quad \forall t \in \mathbb{R} \setminus \{t_j^B | j \in \{1, 2, \ldots, N\}\}.
\end{aligned} \tag{4.15}$$

If we consider that B_t in the SEIR model corresponds to N_t in the HawkesN process, then equations (4.15) and (4.14) prove that (4.13) holds. □

Note that also the intensity in equation (4.12) can be written as a limit. To this end, we write the intensity from (4.8) and (4.12) as $\lambda_t^{H,\sigma,\gamma}$ and $\lambda_t^{H,\gamma}$, respectively. Starting from the case $\gamma = \sigma + \Delta$ with $\Delta \neq 0$ and letting $\Delta \to 0$ we see that

$$\begin{aligned}
\lambda_t^{H,\gamma+\Delta,\gamma} &= \left(1 - \frac{N_t}{N}\right) \sum_{t_j^B < t} \beta \frac{\gamma + \Delta}{\gamma - (\gamma + \Delta)} \left(\mathrm{e}^{-(\gamma+\Delta)(t-t_j^B)} - \mathrm{e}^{-\gamma(t-t_j^B)}\right) \\
&= -\left(1 - \frac{N_t}{N}\right) \sum_{t_j^B < t} \beta(\gamma + \Delta) \frac{\mathrm{e}^{-(\gamma+\Delta)(t-t_j^B)} - \mathrm{e}^{-\gamma(t-t_j^B)}}{\Delta} \\
&\xrightarrow{\Delta \to 0} -\left(1 - \frac{N_t}{N}\right) \sum_{t_j^B < t} \beta\gamma \frac{\mathrm{d}\, \mathrm{e}^{-\gamma(t-t_j^B)}}{\mathrm{d}\gamma} \\
&= \left(1 - \frac{N_t}{N}\right) \sum_{t_j^B < t} \beta\gamma (t - t_j^B)\, \mathrm{e}^{-\gamma(t-t_j^B)} = \lambda_t^{H,\gamma}
\end{aligned}$$

holds. So although difference in the kernels in $\lambda_t^{H,\gamma}$ and $\lambda_t^{H,\sigma,\gamma}$ might seem large, the former is just the limit of the latter for $\sigma \to \gamma$.

In figure 4.3 we have plotted the intensity of a HawkesN process that corresponds to an SEIR model. It differs considerably from the corresponding plot in 4.2 which is related to the SIR model. There are no more jumps when a new event happens. If we neglect the factor $(1 - N_t/N)$, then an observed event lets the intensity increase continuously. After some time the intensity starts to decrease again.

This is the desired behavior in the SEIR model because an infection does not immediately increase compartment I which triggers further infections, or in other words, events. As time goes by, exposed individuals move from compartment E to I, so the increase in intensity is delayed. Since we cannot observe the transition from E to I in the HawkesN setting, the increase in intensity is not abrupt but gradual.

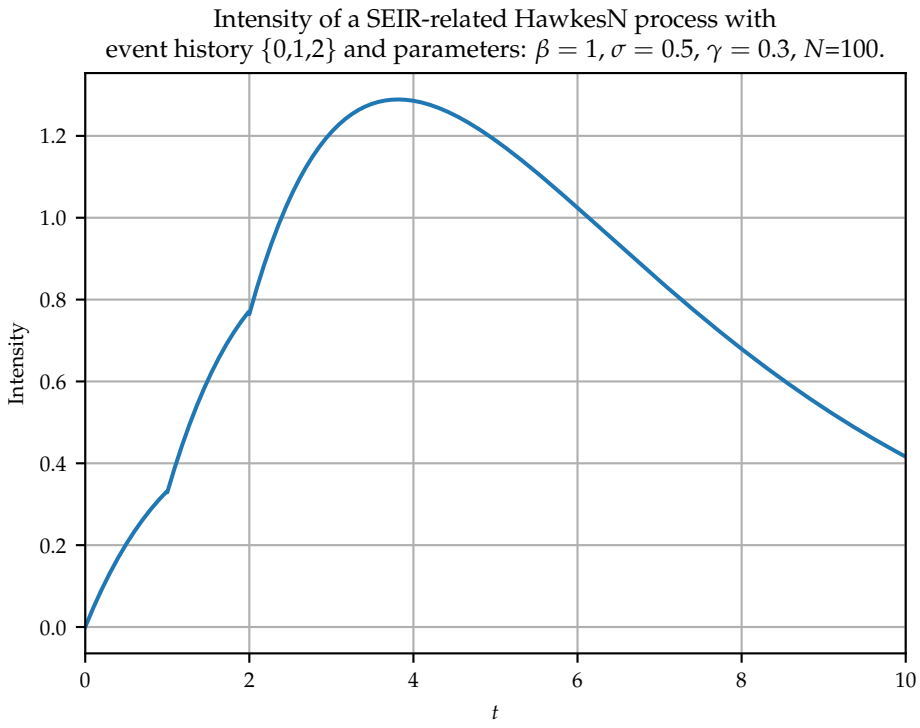


Figure 4.3.: Intensity of an SEIR-related HawkesN Process.

Note that for the intensity of the SEIR-related HawkesN process it does not matter whether the summation index is $t_j^B < t$ or $t_j^B \leq t$ because the potential additional addend in the latter case is always zero. For $\sigma \neq \gamma$ this is the case because for $t = t_j^B$ we have $\mathrm{e}^{-\sigma(t-t_j^B)} - \mathrm{e}^{-\gamma(t-t_j^B)} = 0$. In the case $\sigma = \gamma$ the factor $t - t_j^B$ in the intesity function makes it irrelevant if we sum over $t_j^B < t$ or $t_j^B \leq t$. In the SIR context, on the other hand, the distinction between $<$ and $\leq$ has implications for the intensity of the HawkesN process. This is discussed in more detail in appendix C.1.

4.7. Fitting a HawkesN Process Related to an SIR Model

4.7.1. Maximum Likelihood Estimation

In this subsection we discuss how the unknown parameters of a HawkesN process can be estimated after having observed the event times $t_1^C, t_1^C, \ldots, t_n^C$. We focus on the Maximum Likelihood approach, so the log-likelihood plays a key role in this subsection.

In their proposition 7.2.III, Daley and Vere-Jones (2003) provide the formula of the likelihood function. It is

$$L = \left[\prod_{i=1}^{n} \lambda^H_{t_i^C}\right] \exp\left(-\int_0^{t_n^C} \lambda_u^H \,\mathrm{d}u\right).$$

Taking the logarithm of the likelihood L we arrive at the log-likelihood

$$\ell(\kappa, \theta, N) = \sum_{i=1}^{n} \log\left(\lambda^H_{t_i^C}\right) - \int_0^{t_n^C} \lambda_u^H \,\mathrm{d}u\,. \tag{4.16}$$

Inserting the SIR-related HawkesN intensity with our assumptions of no background intensity and an exponential kernel, i.e. equation (4.6), we obtain

$$\ell(\kappa,\theta,N) = \sum_{i=1}^{n} \log\left(\left(1-\frac{i}{N}\right)\sum_{t_j^C<t_i^C} \kappa\theta\, \mathrm{e}^{-\theta(t_i^C-t_j^C)}\right) - \int_0^{t_n^C}\left(1-\frac{N_u}{N}\right)\sum_{t_j^C<u}\kappa\theta\,\mathrm{e}^{-\theta(u-t_j^C)}\,\mathrm{d}u\,. \tag{4.17}$$

Let us consider the integral in (4.17). If we denote the time of the first event by t_1, then we see that the sum inside the integral is empty for $u < t_1$. Using this we get

$$\int_0^{t_n^C}\left(1-\frac{N_u}{N}\right)\sum_{t_j^C<u}\kappa\theta\,\mathrm{e}^{-\theta(u-t_j^C)}\,\mathrm{d}u = \int_{t_1}^{t_n^C}\left(1-\frac{N_u}{N}\right)\sum_{t_j^C<u}\kappa\theta\,\mathrm{e}^{-\theta(u-t_j^C)}\,\mathrm{d}u\,.$$

Similarly, we conclude that

$$\int_{t_1}^{t_n^C}\left(1-\frac{N_u}{N}\right)\sum_{t_j^C<u}\kappa\theta\,\mathrm{e}^{-\theta(u-t_j^C)}\,\mathrm{d}u = \sum_{j=1}^{n-1}\int_{t_j^C}^{t_n^C}\left(1-\frac{N_u}{N}\right)\kappa\theta\,\mathrm{e}^{-\theta(u-t_j^C)}\,\mathrm{d}u\,.$$

Taking advantage of the fact that $N_u = l$ for $u \in \left[t_l^C, t_{l+1}^C\right)$, we obtain

$$\sum_{j=1}^{n-1}\int_{t_j^C}^{t_n^C}\left(1-\frac{N_u}{N}\right)\kappa\theta\,\mathrm{e}^{-\theta(u-t_j^C)}\,\mathrm{d}u = \sum_{j=1}^{n-1}\sum_{l=j}^{n-1}\left(1-\frac{l}{N}\right)\int_{t_l^C}^{t_{l+1}^C}\kappa\theta\,\mathrm{e}^{-\theta(u-t_j^C)}\,\mathrm{d}u\,,$$

where we can easily solve the integral. So the integral in equation (4.17) equals

$$\kappa\sum_{j=1}^{n-1}\sum_{l=j}^{n-1}\left(1-\frac{l}{N}\right)\left(\mathrm{e}^{-\theta(t_l^C-t_j^C)}-\mathrm{e}^{-\theta(t_{l+1}^C-t_j^C)}\right).$$

With this simplification equation (4.17) becomes

$$\ell(\kappa,\theta,N) = \sum_{i=1}^{n} \log\left(\left(1-\frac{i}{N}\right)\sum_{t_j^C<t_i^C} \kappa\theta\, \mathrm{e}^{-\theta(t_i^C-t_j^C)}\right) - \kappa\sum_{j=1}^{n-1}\sum_{l=j}^{n-1}\left(1-\frac{l}{N}\right)\left(\mathrm{e}^{-\theta(t_l^C-t_j^C)}-\mathrm{e}^{-\theta(t_{l+1}^C-t_j^C)}\right). \tag{4.18}$$

Even though there is no integral left in (4.18) anymore, we refer to the first part on the right side as the sum part and to the second part as the integral part.

We have implemented the likelihood formula in our Python program. Section C.2.1 demonstrates its use and shows how it differs from the R program written by Rizoiu et al. (2018a).

Now that we have a function for determining the log-likelihood, we are interested in the function's maximum. In analogy to Rizoiu et al. (2018b) we utilize the L-BFGS algorithm to achieve this goal. In order to avoid the numerical computation of the function's gradient, we calculate the partial derivatives of the log-likelihood symbolically with respect to all three parameters κ, θ, and N as in Rizoiu et al. (2018b). The partial derivative with respect to κ is

$$\frac{\partial\ell(\kappa,\theta,N)}{\partial\kappa} = \sum_{i=1}^{n}\frac{\left(1-\frac{i}{N}\right)\sum_{t_j^C<t_i^C}\theta\,\mathrm{e}^{-\theta(t_i^C-t_j^C)}}{\left(1-\frac{i}{N}\right)\sum_{t_j^C<t_i^C}\kappa\theta\,\mathrm{e}^{-\theta(t_i^C-t_j^C)}} - \sum_{j=1}^{n-1}\sum_{l=j}^{n-1}\left(1-\frac{l}{N}\right)\left(\mathrm{e}^{-\theta(t_l^C-t_j^C)}-\mathrm{e}^{-\theta(t_{l+1}^C-t_j^C)}\right)$$

Note, that the numerator and the denominator in the sum part largely cancel out, leaving us with $\sum_{i=1}^{n}\frac{1}{\kappa} = \frac{n}{\kappa}$. So we get the derivative in the simplified form as in

$$\frac{\partial\ell(\kappa,\theta,N)}{\partial\kappa} = \frac{n}{\kappa} - \sum_{j=1}^{n-1}\sum_{l=j}^{n-1}\left(1-\frac{l}{N}\right)\left(\mathrm{e}^{-\theta(t_l^C-t_j^C)}-\mathrm{e}^{-\theta(t_{l+1}^C-t_j^C)}\right). \tag{4.19}$$

Taking the partial derivative of the log-likelihood with respect to θ we get

$$\frac{\partial \ell}{\partial \theta} = \sum_{i=1}^{n} \frac{\frac{\partial \left(\left(1-\frac{i}{N}\right) \sum_{t_j^C < t_i^C} \kappa\theta \, \mathrm{e}^{-\theta(t_i^C - t_j^C)} \right)}{\partial \theta}}{\left(1-\frac{i}{N}\right) \left(\sum_{t_j^C < t_i^C} \kappa\theta \, \mathrm{e}^{-\theta(t_i^C - t_j^C)} \right)} - \kappa \sum_{j=1}^{n-1} \sum_{l=j}^{n-1} \left(1-\frac{l}{N}\right) \left(-(t_l^C - t_j^C) \, \mathrm{e}^{-\theta(t_l^C - t_j^C)} + (t_{l+1}^C - t_j^C) \, \mathrm{e}^{-\theta(t_{l+1}^C - t_j^C)} \right).$$

Also in this derivative we can simplify the sum part. Due to

$$\begin{aligned} &\sum_{i=1}^{n} \frac{\frac{\partial \left(\left(1-\frac{i}{N}\right) \sum_{t_j^C < t_i^C} \kappa\theta \, \mathrm{e}^{-\theta(t_i^C - t_j^C)} \right)}{\partial \theta}}{\left(1-\frac{i}{N}\right) \left(\sum_{t_j^C < t_i^C} \kappa\theta \, \mathrm{e}^{-\theta(t_i^C - t_j^C)} \right)} \\ &= \sum_{i=1}^{n} \frac{\sum_{t_j^C < t_i^C} \left(\mathrm{e}^{-\theta(t_i^C - t_j^C)} - \theta(t_i^C - t_j^C) \, \mathrm{e}^{-\theta(t_i^C - t_j^C)} \right)}{\sum_{t_j^C < t_i^C} \theta \, \mathrm{e}^{-\theta(t_i^C - t_j^C)}} \\ &= \sum_{i=1}^{n} \frac{\sum_{t_j^C < t_i^C} \left(1 - \theta(t_i^C - t_j^C) \right) \mathrm{e}^{-\theta(t_i^C - t_j^C)}}{\sum_{t_j^C < t_i^C} \theta \, \mathrm{e}^{-\theta(t_i^C - t_j^C)}} \end{aligned}$$

we obtain

$$\frac{\partial \ell}{\partial \theta} = \sum_{i=1}^{n} \frac{\sum_{t_j^C < t_i^C} \left(1 - \theta(t_i^C - t_j^C) \right) \mathrm{e}^{-\theta(t_i^C - t_j^C)}}{\sum_{t_j^C < t_i^C} \theta \, \mathrm{e}^{-\theta(t_i^C - t_j^C)}} - \kappa \sum_{j=1}^{n-1} \sum_{l=j}^{n-1} \left(1-\frac{l}{N}\right) \left(-(t_l^C - t_j^C) \, \mathrm{e}^{-\theta(t_l^C - t_j^C)} + (t_{l+1}^C - t_j^C) \, \mathrm{e}^{-\theta(t_{l+1}^C - t_j^C)} \right). \tag{4.20}$$

As partial derivative with respect to N we get

$$\frac{\partial \ell(\kappa,\theta,N)}{\partial N} = \sum_{i=1}^{n} \frac{\frac{\partial\left(\left(1-\frac{i}{N}\right)\sum_{t_j^C<t_i^C}\kappa\theta\,\mathrm{e}^{-\theta(t_i^C-t_j^C)}\right)}{\partial N}}{\left(1-\frac{i}{N}\right)\left(\sum_{t_j^C<t_i^C}\kappa\theta\,\mathrm{e}^{-\theta(t_i^C-t_j^C)}\right)} - \kappa\sum_{j=1}^{n-1}\sum_{l=j}^{n-1}\frac{l}{N^2}\left(\mathrm{e}^{-\theta(t_l^C-t_j^C)}-\mathrm{e}^{-\theta(t_{l+1}^C-t_j^C)}\right).$$

Again, we can simplify the first sum, reducing it to

$$\begin{aligned}
&\sum_{i=1}^{n} \frac{\frac{\partial\left(\left(1-\frac{i}{N}\right)\sum_{t_j^C<t_i^C}\kappa\theta\,\mathrm{e}^{-\theta(t_i^C-t_j^C)}\right)}{\partial N}}{\left(1-\frac{i}{N}\right)\sum_{t_j^C<t_i^C}\kappa\theta\,\mathrm{e}^{-\theta(t_i^C-t_j^C)}} \\
&= \sum_{i=1}^{n} \frac{\frac{i}{N^2}\sum_{t_j^C<t_i^C}\kappa\theta\,\mathrm{e}^{-\theta(t_i^C-t_j^C)}}{\frac{N-i}{N}\sum_{t_j^C<t_i^C}\kappa\theta\,\mathrm{e}^{-\theta(t_i^C-t_j^C)}} \\
&= \sum_{i=1}^{n} \frac{i}{N(N-i)}.
\end{aligned} \tag{4.21}$$

Thus, the partial derivative of the log-likelihood with respect to N is

$$\frac{\partial \ell(\kappa,\theta,N)}{\partial N} = \sum_{i=1}^{n}\frac{i}{N(N-i)} - \frac{\kappa}{N^2}\sum_{j=1}^{n-1}\sum_{l=j}^{n-1} l\left(\mathrm{e}^{-\theta(t_l^C-t_j^C)}-\mathrm{e}^{-\theta(t_{l+1}^C-t_j^C)}\right). \tag{4.22}$$

Rizoiu et al. (2018b) recognize that there is a lower bound for this partial derivative. Using $N-i \leq N$ for all $i \geq 1$ and $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$ we obtain

$$\frac{\partial \ell(\kappa,\theta,N)}{\partial N} \geq \frac{1}{N^2} \underbrace{\left(\frac{n(n+1)}{2} - \kappa \sum_{j=1}^{n-1} \sum_{l=j}^{n-1} l \left(e^{-\theta(t_l^C - t_j^C)} - e^{-\theta(t_{l+1}^C - t_j^C)} \right) \right)}_{\mathcal{S}(\kappa,\theta,\mathcal{T})} \tag{4.23}$$

which serves as lower bound for the derivative of $\ell(\kappa, \theta, N)$ with respect to N. We see that only the first factor in (4.23) contains N. We call the second factor in this equation $\mathcal{S}(\kappa, \theta, \mathcal{T})$. It is independent of N and depends only on κ, θ, and the event times $\mathcal{T}$. As stated by Rizoiu et al. (2018b) $\mathcal{S}(\kappa, \theta, \mathcal{T}) > 0$ implies $\frac{\partial \ell(\kappa,\theta,N)}{\partial N} > 0$ which means that there is no maximum of $\ell(\kappa, \theta, N)$ given $(\kappa, \theta, \mathcal{T})$.

We have implemented the functions for calculating the gradient of the log-likelihood function in our Python program. Subsection C.2.2 demonstrates how to use these functions and highlights differences to the R program of Rizoiu et al. (2018a).

4.7.2. Estimation Results

As with the SIR model we first fit a HawkesN process assuming that two of three model parameters are known. With κ and θ fixed, we estimate the value of N. In figure 4.4 we see that after a few time units this approach leads to estimations that are close to the true value of N.

To see that such an estimation procedure works reasonably well also for other κ-θ-combinations we have generated the heat map in figure 4.5 . Similar to its counterpart in the SIR model (figure 3.10) it shows decent estimations of N for the majority of parameter combinations.

Of course, it is often unrealistic to assume that the parameters κ and θ are known. If we have to maximize the model's likelihood with respect to all three parameters we get much worse results. They are summarized in figure 4.6. This heat map shows that for many combinations of κ and θ the estimations of N are prone to an exploding behavior. The rightmost cell at

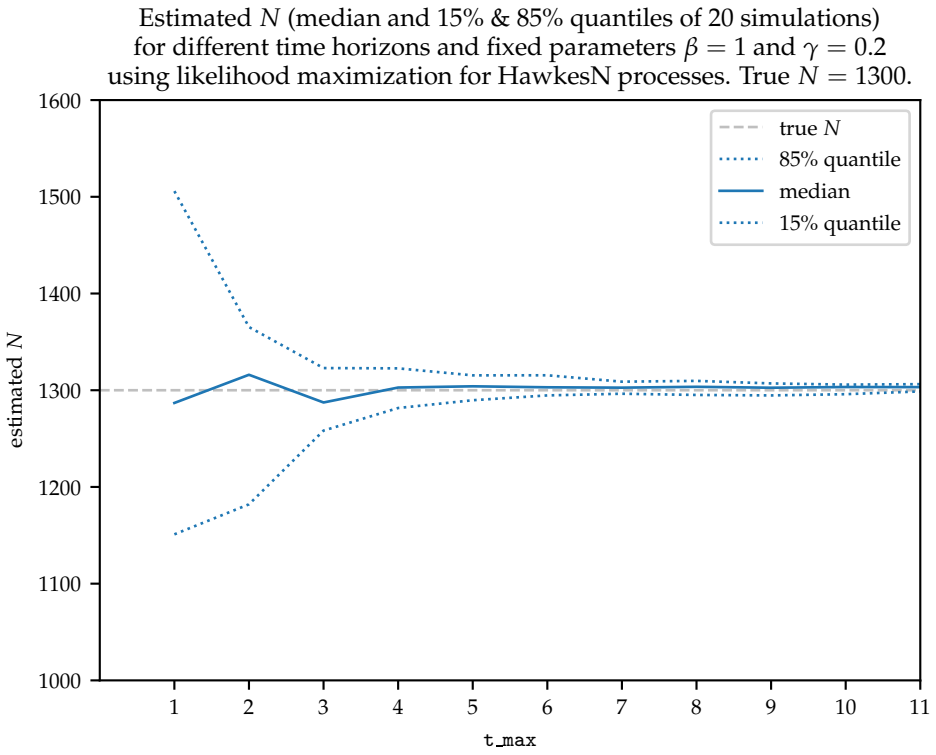
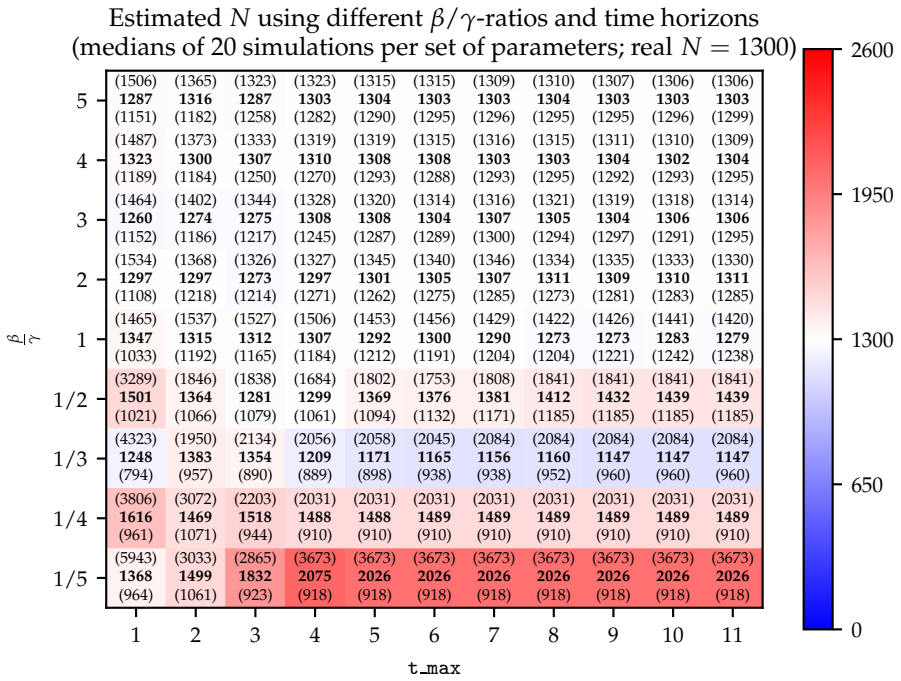


Figure 4.4.: Fitting N of the HawkesN process with known β and γ.



$\frac{\beta}{\gamma}$	1	2	3	4	5	6	7	8	9	10	11
5	(1506) **1287** (1151)	(1365) **1316** (1182)	(1323) **1287** (1258)	(1323) **1303** (1282)	(1315) **1304** (1290)	(1315) **1303** (1295)	(1309) **1303** (1296)	(1310) **1304** (1295)	(1307) **1303** (1295)	(1306) **1303** (1296)	(1306) **1303** (1299)
4	(1487) **1323** (1189)	(1373) **1300** (1184)	(1333) **1307** (1250)	(1319) **1310** (1270)	(1319) **1308** (1293)	(1315) **1308** (1288)	(1316) **1303** (1293)	(1315) **1303** (1295)	(1311) **1304** (1292)	(1310) **1302** (1293)	(1309) **1304** (1295)
3	(1464) **1260** (1152)	(1402) **1274** (1186)	(1344) **1275** (1217)	(1328) **1308** (1245)	(1320) **1308** (1287)	(1314) **1304** (1289)	(1316) **1307** (1300)	(1321) **1305** (1294)	(1319) **1304** (1297)	(1318) **1306** (1291)	(1314) **1306** (1295)
2	(1534) **1297** (1108)	(1368) **1297** (1218)	(1326) **1273** (1214)	(1327) **1297** (1271)	(1345) **1301** (1262)	(1340) **1305** (1275)	(1346) **1307** (1285)	(1334) **1311** (1273)	(1335) **1309** (1281)	(1333) **1310** (1283)	(1330) **1311** (1285)
1	(1465) **1347** (1033)	(1537) **1315** (1192)	(1527) **1312** (1165)	(1506) **1307** (1184)	(1453) **1292** (1212)	(1456) **1300** (1191)	(1429) **1290** (1204)	(1422) **1273** (1204)	(1426) **1273** (1221)	(1441) **1283** (1242)	(1420) **1279** (1238)
1/2	(3289) **1501** (1021)	(1846) **1364** (1066)	(1838) **1281** (1079)	(1684) **1299** (1061)	(1802) **1369** (1094)	(1753) **1376** (1132)	(1808) **1381** (1171)	(1841) **1412** (1185)	(1841) **1432** (1185)	(1841) **1439** (1185)	(1841) **1439** (1185)
1/3	(4323) **1248** (794)	(1950) **1383** (957)	(2134) **1354** (890)	(2056) **1209** (889)	(2058) **1171** (898)	(2045) **1165** (938)	(2084) **1156** (938)	(2084) **1160** (952)	(2084) **1147** (960)	(2084) **1147** (960)	(2084) **1147** (960)
1/4	(3806) **1616** (961)	(3072) **1469** (1071)	(2203) **1518** (944)	(2031) **1488** (910)	(2031) **1488** (910)	(2031) **1489** (910)	(2031) **1489** (910)	(2031) **1489** (910)	(2031) **1489** (910)	(2031) **1489** (910)	(2031) **1489** (910)
1/5	(5943) **1368** (964)	(3033) **1499** (1061)	(2865) **1832** (923)	(3673) **2075** (918)	(3673) **2026** (918)	(3673) **2026** (918)	(3673) **2026** (918)	(3673) **2026** (918)	(3673) **2026** (918)	(3673) **2026** (918)	(3673) **2026** (918)

Figure 4.5.: Heat map with estimations of N with known scale and decay parameter.

the bottom of the heat map, for example, shows an estimation median for N of 102124 which is far from the true value of 1300.

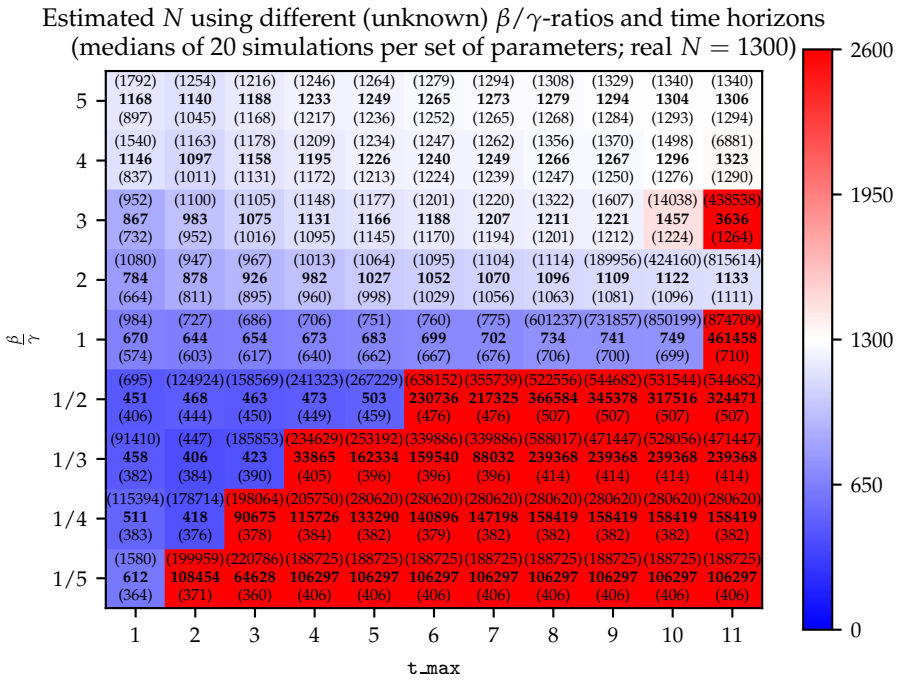


Figure 4.6.: Heat map with estimations of N when maximizing the HawkesN likelihood in all three parameters simultaneously.

In section 4.7.1 we have seen an explanation for the exploding behavior we observed when trying to estimate N. This explanation was based on the lower bound for the partial derivative of the log-likelihood with respect to N in (4.23).

4.8. Fitting a HawkesN Process Related to an SEIR Model

4.8.1. Fitting in the Case $\sigma \neq \gamma$

In this section we discuss the parameter estimation of a HawkesN process related to an SEIR model. As in subsection 4.7 we focus on the Maximum Likelihood approach. Note that the intensity of the HawkesN process plays a pivotal role and we have two of them depending on whether $\sigma \neq \gamma$ or $\sigma = \gamma$ holds. We treat the former case in this subsection and the latter in the following subsection.

In theorem 4.4 we have shown that the intensity of a HawkesN process corresponding to a SEIR model with $\sigma \neq \gamma$ equals

$$\left(1 - \frac{N_t}{N}\right) \sum_{t_j^B < t} \beta \frac{\sigma}{\gamma - \sigma} \left(\mathrm{e}^{-\sigma(t - t_j^B)} - \mathrm{e}^{-\gamma(t - t_j^B)}\right).$$

Using equation (4.16) we obtain

$$\begin{aligned} \ell(\beta, \sigma, \gamma, N) = \sum_{i=1}^{n} \log \left(\left(1 - \frac{i}{N}\right) \sum_{t_j^B < t_i^B} \beta \frac{\sigma}{\gamma - \sigma} \left(\mathrm{e}^{-\sigma(t_i^B - t_j^B)} - \mathrm{e}^{-\gamma(t_i^B - t_j^B)}\right) \right) \\ - \int_0^{t_n^B} \left(1 - \frac{N_u}{N}\right) \sum_{t_j^B < u} \beta \frac{\sigma}{\gamma - \sigma} \left(\mathrm{e}^{-\sigma(u - t_j^B)} - \mathrm{e}^{-\gamma(u - t_j^B)}\right) \mathrm{d}u \,. \end{aligned} \tag{4.24}$$

as log-likelihood. The integral in this equation can be simplified. First we

rearrange terms via

$$\begin{aligned}
&\int_0^{t_n^B} \left(1 - \frac{N_u}{N}\right) \sum_{t_j^B < u} \beta \frac{\sigma}{\gamma - \sigma} \left(\mathrm{e}^{-\sigma(u - t_j^B)} - \mathrm{e}^{-\gamma(u - t_j^B)}\right) \mathrm{d}u \\
&= \sum_{j=1}^{n-1} \int_{t_j^B}^{t_n^B} \left(1 - \frac{N_u}{N}\right) \beta \frac{\sigma}{\gamma - \sigma} \left(\mathrm{e}^{-\sigma(u - t_j^B)} - \mathrm{e}^{-\gamma(u - t_j^B)}\right) \mathrm{d}u \\
&= \sum_{j=1}^{n-1} \sum_{l=j}^{n-1} \left(1 - \frac{l}{N}\right) \beta \frac{\sigma}{\gamma - \sigma} \int_{t_l^B}^{t_{l+1}^B} \left(\mathrm{e}^{-\sigma(u - t_j^B)} - \mathrm{e}^{-\gamma(u - t_j^B)}\right) \mathrm{d}u \,. \qquad (4.25)
\end{aligned}$$

In equation (4.25) we can identify the two integrals

$$\int_{t_l^B}^{t_{l+1}^B} \mathrm{e}^{-\sigma(u - t_j^B)} \, \mathrm{d}u = \frac{-1}{\sigma} \left(\mathrm{e}^{-\sigma\left(t_{l+1}^B - t_j^B\right)} - \mathrm{e}^{-\sigma\left(t_l^B - t_j^B\right)}\right)$$

and

$$\int_{t_l^B}^{t_{l+1}^B} \mathrm{e}^{-\gamma(u - t_j^B)} \, \mathrm{d}u = \frac{-1}{\gamma} \left(\mathrm{e}^{-\gamma\left(t_{l+1}^B - t_j^B\right)} - \mathrm{e}^{-\gamma\left(t_l^B - t_j^B\right)}\right).$$

Substituting the integral in equation (4.24) by the simplified expression leads to

$$\begin{aligned}
\ell(\beta, \sigma, \gamma, N) = \sum_{i=1}^{n} \log &\left(\left(1 - \frac{i}{N}\right) \sum_{t_j^B < t_i^B} \beta \frac{\sigma}{\gamma - \sigma} \left(\mathrm{e}^{-\sigma(t_i^B - t_j^B)} - \mathrm{e}^{-\gamma(t_i^B - t_j^B)}\right) \right) \\
&- \sum_{j=1}^{n-1} \sum_{l=j}^{n-1} \left(1 - \frac{l}{N}\right) \beta \frac{\sigma}{\gamma - \sigma} \left(\frac{1}{\sigma} \left(\mathrm{e}^{-\sigma\left(t_l^B - t_j^B\right)} - \mathrm{e}^{-\sigma\left(t_{l+1}^B - t_j^B\right)}\right) \right. \\
&\left. - \frac{1}{\gamma} \left(\mathrm{e}^{-\gamma\left(t_l^B - t_j^B\right)} - \mathrm{e}^{-\gamma\left(t_{l+1}^B - t_j^B\right)}\right) \right)
\end{aligned}$$

as the log-likelihood of the HawkesN process.

In order to maximize the log-likelihood, we calculate the partial derivatives

with respect to its parameters. The partial derivative w.r.t. β is

$$\frac{\partial \ell(\beta,\sigma,\gamma,N)}{\partial \beta} = \sum_{i=1}^{n} \frac{\left(1-\frac{i}{N}\right)\sum_{t_j^B<t_i^B}\frac{\sigma}{\gamma-\sigma}\left(\mathrm{e}^{-\sigma(t_i^B-t_j^B)}-\mathrm{e}^{-\gamma(t_i^B-t_j^B)}\right)}{\left(1-\frac{i}{N}\right)\sum_{t_j^B<t_i^B}\beta\frac{\sigma}{\gamma-\sigma}\left(\mathrm{e}^{-\sigma(t_i^B-t_j^B)}-\mathrm{e}^{-\gamma(t_i^B-t_j^B)}\right)} - \sum_{j=1}^{n-1}\sum_{l=j}^{n-1}\left(1-\frac{l}{N}\right)\frac{\sigma}{\gamma-\sigma}\left(\frac{1}{\sigma}\left(\mathrm{e}^{-\sigma\left(t_l^B-t_j^B\right)}-\mathrm{e}^{-\sigma\left(t_{l+1}^B-t_j^B\right)}\right) - \frac{1}{\gamma}\left(\mathrm{e}^{-\gamma\left(t_l^B-t_j^B\right)}-\mathrm{e}^{-\gamma\left(t_{l+1}^B-t_j^B\right)}\right)\right),$$

where the sum part is $\sum_{i=1}^{n}\frac{1}{\beta}=\frac{n}{\beta}$. So

$$\frac{\partial \ell(\beta,\sigma,\gamma,N)}{\partial \beta} = \frac{n}{\beta} - \sum_{j=1}^{n-1}\sum_{l=j}^{n-1}\left(1-\frac{l}{N}\right)\frac{\sigma}{\gamma-\sigma}\left(\frac{1}{\sigma}\left(\mathrm{e}^{-\sigma\left(t_l^B-t_j^B\right)}-\mathrm{e}^{-\sigma\left(t_{l+1}^B-t_j^B\right)}\right) - \frac{1}{\gamma}\left(\mathrm{e}^{-\gamma\left(t_l^B-t_j^B\right)}-\mathrm{e}^{-\gamma\left(t_{l+1}^B-t_j^B\right)}\right)\right) \tag{4.26}$$

holds.

Due to space restrictions we split the derivative with respect to σ into multiple parts. First we analyze the sum part of the log-likelihood denoted by ℓ^{sum}. The outer derivative of the logarithm in the sum part is 1 divided by the logarithm's argument. The inner derivative is

$$\left(1-\frac{i}{N}\right)\beta\left[\frac{\gamma}{(\gamma-\sigma)^2}\sum_{t_j^B<t_i^B}\left(\mathrm{e}^{-\sigma(t_i^B-t_j^B)}-\mathrm{e}^{-\gamma(t_i^B-t_j^B)}\right) + \frac{\sigma}{\gamma-\sigma}\sum_{t_j^B<t_i^B}\left(-(t_i^B-t_j^B)\right)\mathrm{e}^{-\sigma(t_i^B-t_j^B)}\right]$$

where the product rule was used. Note that the factor $\left(1-\frac{i}{N}\right)\beta$ appears in both the inner and the outer derivative of the logarithm, so it cancels out. After removing further terms which occur on both sides of the fraction bar we obtain

$$\begin{aligned}\frac{\partial \ell^{\text{sum}}(\beta,\sigma,\gamma,N)}{\partial \sigma} &= \sum_{i=1}^{n}\left[\frac{\frac{\gamma}{(\gamma-\sigma)}}{\sigma}+\frac{\sum_{t_j^B<t_i^B}\left(-(t_i^B-t_j^B)\right)\mathrm{e}^{-\sigma(t_i^B-t_j^B)}}{\sum_{t_j^B<t_i^B}\left(\mathrm{e}^{-\sigma(t_i^B-t_j^B)}-\mathrm{e}^{-\gamma(t_i^B-t_j^B)}\right)}\right]\\ &= \frac{n\gamma}{\sigma(\gamma-\sigma)}+\sum_{i=1}^{n}\frac{\sum_{t_j^B<t_i^B}\left(-(t_i^B-t_j^B)\right)\mathrm{e}^{-\sigma(t_i^B-t_j^B)}}{\sum_{t_j^B<t_i^B}\left(\mathrm{e}^{-\sigma(t_i^B-t_j^B)}-\mathrm{e}^{-\gamma(t_i^B-t_j^B)}\right)}\end{aligned}$$

as the derivative of the log-likelihood's sum part w.r.t. σ. To calculate the corresponding derivative of the integral part – denoted by ℓ^{int} – we write it as

$$\begin{aligned}\ell^{\text{int}}(\beta,\sigma,\gamma,N) = \sum_{j=1}^{n-1}\sum_{l=j}^{n-1}\left(1-\frac{l}{N}\right)\beta\Bigg[&\frac{1}{\gamma-\sigma}\left(\mathrm{e}^{-\sigma\left(t_l^B-t_j^B\right)}-\mathrm{e}^{-\sigma\left(t_{l+1}^B-t_j^B\right)}\right)\\ &-\frac{\sigma}{\gamma(\gamma-\sigma)}\left(\mathrm{e}^{-\gamma\left(t_l^B-t_j^B\right)}-\mathrm{e}^{-\gamma\left(t_{l+1}^B-t_j^B\right)}\right)\Bigg],\end{aligned} \tag{4.27}$$

where the terms outside the squared brackets do not depend on σ. Using the product rule the partial derivative of

$$\frac{1}{\gamma-\sigma}\left(\mathrm{e}^{-\sigma\left(t_l^B-t_j^B\right)}-\mathrm{e}^{-\sigma\left(t_{l+1}^B-t_j^B\right)}\right) \tag{4.28}$$

w.r.t. σ is

$$\begin{aligned}&\frac{1}{(\gamma-\sigma)^2}\left(\mathrm{e}^{-\sigma\left(t_l^B-t_j^B\right)}-\mathrm{e}^{-\sigma\left(t_{l+1}^B-t_j^B\right)}\right)\\ &+\frac{1}{\gamma-\sigma}\left(\left(-\left(t_l^B-t_j^B\right)\right)\mathrm{e}^{-\sigma\left(t_l^B-t_j^B\right)}-\left(-\left(t_{l+1}^B-t_j^B\right)\right)\mathrm{e}^{-\sigma\left(t_{l+1}^B-t_j^B\right)}\right).\end{aligned}$$

The corresponding derivative of

$$\frac{\sigma}{\gamma(\gamma-\sigma)}\left(\mathrm{e}^{-\gamma\left(t_l^B-t_j^B\right)}-\mathrm{e}^{-\gamma\left(t_{l+1}^B-t_j^B\right)}\right) \tag{4.29}$$

is

$$\frac{1}{(\gamma-\sigma)^2}\left(\mathrm{e}^{-\gamma\left(t_l^B-t_j^B\right)}-\mathrm{e}^{-\gamma\left(t_{l+1}^B-t_j^B\right)}\right).$$

Now that we have formed the partial derivative w.r.t. σ for all parts of the log-likelihood, we turn to the partial derivative w.r.t. γ. Again, we first look at the sum part with the outer derivative of the logarithm therein being 1 divided by the logarithm's argument. The inner derivative of the logarithm is

$$\left(1-\frac{i}{N}\right)\beta\left[\frac{-\sigma}{(\gamma-\sigma)^2}\sum_{t_j^B<t_i^B}\left(\mathrm{e}^{-\sigma(t_i^B-t_j^B)}-\mathrm{e}^{-\gamma(t_i^B-t_j^B)}\right)\right.$$
$$\left.+\frac{\sigma}{\gamma-\sigma}\sum_{t_j^B<t_i^B}\left(-(t_i^B-t_j^B)\right)\mathrm{e}^{-\gamma(t_i^B-t_j^B)}\right]$$

and thus we get

$$\begin{aligned}\frac{\partial\ell^{\mathrm{sum}}(\beta,\sigma,\gamma,N)}{\partial\gamma}&=\sum_{i=1}^{n}\left[\frac{-1}{(\gamma-\sigma)}+\frac{\sum_{t_j^B<t_i^B}\left(-(t_i^B-t_j^B)\right)\mathrm{e}^{-\gamma(t_i^B-t_j^B)}}{\sum_{t_j^B<t_i^B}\left(\mathrm{e}^{-\sigma(t_i^B-t_j^B)}-\mathrm{e}^{-\gamma(t_i^B-t_j^B)}\right)}\right]\\&=\frac{-n}{\gamma-\sigma}+\sum_{i=1}^{n}\frac{\sum_{t_j^B<t_i^B}\left(-(t_i^B-t_j^B)\right)\mathrm{e}^{-\gamma(t_i^B-t_j^B)}}{\sum_{t_j^B<t_i^B}\left(\mathrm{e}^{-\sigma(t_i^B-t_j^B)}-\mathrm{e}^{-\gamma(t_i^B-t_j^B)}\right)}.\end{aligned}$$

For the derivative of the log-likelihood's integral part we start from equation (4.27). Only the expression inside the squared brackets depends on γ, so we only consider this part when forming the derivative. We also calculate the derivative separately for the first and the second half of the expression inside the squared brackets.

The partial derivatives of (4.28) and (4.29) w.r.t. γ are

$$\frac{-1}{(\gamma-\sigma)^2}\left(\mathrm{e}^{-\sigma\left(t_l^B-t_j^B\right)}-\mathrm{e}^{-\sigma\left(t_{l+1}^B-t_j^B\right)}\right)$$

and

$$\frac{\sigma(\sigma-2\gamma)}{\gamma^2(\gamma-\sigma)^2}\left(\mathrm{e}^{-\gamma\left(t_l^B-t_j^B\right)}-\mathrm{e}^{-\gamma\left(t_{l+1}^B-t_j^B\right)}\right)$$
$$+\frac{\sigma}{\gamma(\gamma-\sigma)}\left(\left(t_{l+1}^B-t_j^B\right)\mathrm{e}^{-\gamma\left(t_{l+1}^B-t_j^B\right)}-\left(t_l^B-t_j^B\right)\mathrm{e}^{-\gamma\left(t_l^B-t_j^B\right)}\right),$$

respectively.

The fourth parameter of the log-likelihood is N. The partial derivative of the log-likelihood's sum part w.r.t. N is the same as in (4.21). For the integral part we obtain

$$\sum_{j=1}^{n-1}\sum_{l=j}^{n-1}\frac{l}{N^2}\frac{\beta\sigma}{\gamma-\sigma}\left(\frac{1}{\sigma}\left(\mathrm{e}^{-\sigma\left(t_l^B-t_j^B\right)}-\mathrm{e}^{-\sigma\left(t_{l+1}^B-t_j^B\right)}\right)\right.$$
$$\left.-\frac{1}{\gamma}\left(\mathrm{e}^{-\gamma\left(t_l^B-t_j^B\right)}-\mathrm{e}^{-\gamma\left(t_{l+1}^B-t_j^B\right)}\right)\right)$$

as the derivative. Thus,

$$\frac{\partial\ell}{\partial N}=\sum_{i=1}^{n}\frac{i}{N(N-i)}-\sum_{j=1}^{n-1}\sum_{l=j}^{n-1}\frac{l}{N^2}\frac{\beta\sigma}{\gamma-\sigma}\left(\frac{1}{\sigma}\left(\mathrm{e}^{-\sigma\left(t_l^B-t_j^B\right)}-\mathrm{e}^{-\sigma\left(t_{l+1}^B-t_j^B\right)}\right)\right.$$
$$\left.-\frac{1}{\gamma}\left(\mathrm{e}^{-\gamma\left(t_l^B-t_j^B\right)}-\mathrm{e}^{-\gamma\left(t_{l+1}^B-t_j^B\right)}\right)\right) \tag{4.30}$$

holds true. With the same reasoning as in the step from eqution (4.22) to (4.23) we get the lower bound

$$\frac{\partial\ell}{\partial N}\geq\frac{1}{N^2}\left[\frac{n(n+1)}{2}-\sum_{j=1}^{n-1}\sum_{l=j}^{n-1}l\frac{\beta\sigma}{\gamma-\sigma}\left(\frac{1}{\sigma}\left(\mathrm{e}^{-\sigma\left(t_l^B-t_j^B\right)}-\mathrm{e}^{-\sigma\left(t_{l+1}^B-t_j^B\right)}\right)\right.\right.$$
$$\left.\left.-\frac{1}{\gamma}\left(\mathrm{e}^{-\gamma\left(t_l^B-t_j^B\right)}-\mathrm{e}^{-\gamma\left(t_{l+1}^B-t_j^B\right)}\right)\right)\right]. \tag{4.31}$$

So if the expression inside the square bracket in (4.31) is positive, we can – ceteris paribus – increase N in order to increase the likelihood.

4.8.2. Fitting in the Case $\sigma = \gamma$

In case of an SEIR model with $\sigma = \gamma$ the intensity of the corresponding HawkesN process equals

$$\left(1 - \frac{N_t}{N}\right) \sum_{t_j^B < t} \beta\gamma (t - t_j^B)\, \mathrm{e}^{-\gamma(t - t_j^B)}$$

according to theorem 4.5. Using this intensity in (4.16) leads to

$$\begin{aligned} \ell(\beta, \gamma, N) = \sum_{i=1}^{n} \log \left(\left(1 - \frac{i}{N}\right) \sum_{t_j^B < t_i^B} \beta\gamma \left(t_i^B - t_j^B\right) \mathrm{e}^{-\gamma(t_i^B - t_j^B)} \right) \\ - \int_0^{t_n^B} \left(1 - \frac{N_u}{N}\right) \sum_{t_j^B < u} \beta\gamma \left(u - t_j^B\right) \mathrm{e}^{-\gamma(u - t_j^B)}\, \mathrm{d}u \end{aligned} \tag{4.32}$$

as log-likelihood. As in the case $\sigma \neq \gamma$ we can simplify the integral. First we rearrange terms according to

$$\begin{aligned} &\int_0^{t_n^B} \left(1 - \frac{N_u}{N}\right) \sum_{t_j^B < u} \beta\gamma \left(u - t_j^B\right) \mathrm{e}^{-\gamma(u - t_j^B)}\, \mathrm{d}u \\ &= \sum_{j=1}^{n-1} \int_{t_j^B}^{t_n^B} \left(1 - \frac{N_u}{N}\right) \beta\gamma \left(u - t_j^B\right) \mathrm{e}^{-\gamma(u - t_j^B)}\, \mathrm{d}u \\ &= \sum_{j=1}^{n-1} \sum_{l=j}^{n-1} \left(1 - \frac{l}{N}\right) \beta \int_{t_l^B}^{t_{l+1}^B} \gamma \left(u - t_j^B\right) \mathrm{e}^{-\gamma(u - t_j^B)}\, \mathrm{d}u\,. \end{aligned}$$

Applying integration by parts to the remaining integral yields

$$\begin{aligned} \int_{t_l^B}^{t_{l+1}^B} \gamma \left(u - t_j^B\right) \mathrm{e}^{-\gamma(u - t_j^B)}\, \mathrm{d}u = \frac{1}{\gamma} \Big[&\mathrm{e}^{-\gamma(t_{l+1}^B - t_j^B)} \left(-\gamma(t_{l+1}^B - t_j^B) - 1\right) \\ &- \mathrm{e}^{-\gamma(t_l^B - t_j^B)} \left(-\gamma(t_l^B - t_j^B) - 1\right) \Big]\,. \end{aligned}$$

With this simpler form of the integral equation (4.32) reads

$$\ell(\beta,\gamma,N) = \sum_{i=1}^{n} \log\left(\left(1-\frac{i}{N}\right)\sum_{t_j^B<t_i^B}\beta\gamma\left(t_i^B-t_j^B\right)\mathrm{e}^{-\gamma(t_i^B-t_j^B)}\right) - \sum_{j=1}^{n-1}\sum_{l=j}^{n-1}\left(1-\frac{l}{N}\right)\frac{\beta}{\gamma}\left[\mathrm{e}^{-\gamma(t_{l+1}^B-t_j^B)}\left(-\gamma(t_{l+1}^B-t_j^B)-1\right) - \mathrm{e}^{-\gamma(t_l^B-t_j^B)}\left(-\gamma(t_l^B-t_j^B)-1\right)\right]. \quad (4.33)$$

Next, we calculate the partial derivatives of the log-likelihood in (4.33). The partial derivative of the sum part w.r.t. β is $\frac{n}{\beta}$ as in (4.26). Hence we have

$$\frac{\partial\ell(\beta,\gamma,N)}{\partial\beta} = \frac{n}{\beta} - \sum_{j=1}^{n-1}\sum_{l=j}^{n-1}\left(1-\frac{l}{N}\right)\frac{1}{\gamma}\left[\mathrm{e}^{-\gamma(t_{l+1}^B-t_j^B)}\left(-\gamma(t_{l+1}^B-t_j^B)-1\right) - \mathrm{e}^{-\gamma(t_l^B-t_j^B)}\left(-\gamma(t_l^B-t_j^B)-1\right)\right].$$

The derivatives of the log-likelihood's sum and integral part w.r.t. γ are

$$\begin{aligned}
&\sum_{i=1}^{n}\frac{\left(1-\frac{i}{N}\right)\sum_{t_j^B<t_i^B}\beta\left[\left(t_i^B-t_j^B\right)\mathrm{e}^{-\gamma(t_i^B-t_j^B)} - \gamma\left(t_i^B-t_j^B\right)^2\mathrm{e}^{-\gamma(t_i^B-t_j^B)}\right]}{\left(1-\frac{i}{N}\right)\sum_{t_j^B<t_i^B}\beta\gamma\left(t_i^B-t_j^B\right)\mathrm{e}^{-\gamma(t_i^B-t_j^B)}} \\
&= \sum_{i=1}^{n}\left[\frac{1}{\gamma} - \frac{\sum_{t_j^B<t_i^B}\left(t_i^B-t_j^B\right)^2\mathrm{e}^{-\gamma(t_i^B-t_j^B)}}{\sum_{t_j^B<t_i^B}\left(t_i^B-t_j^B\right)\mathrm{e}^{-\gamma(t_i^B-t_j^B)}}\right] \\
&= \frac{n}{\gamma} - \sum_{i=1}^{n}\frac{\sum_{t_j^B<t_i^B}\left(t_i^B-t_j^B\right)^2\mathrm{e}^{-\gamma(t_i^B-t_j^B)}}{\sum_{t_j^B<t_i^B}\left(t_i^B-t_j^B\right)\mathrm{e}^{-\gamma(t_i^B-t_j^B)}}
\end{aligned}$$

and

$$\sum_{j=1}^{n-1}\sum_{l=j}^{n-1}\left(1-\frac{l}{N}\right)\frac{\beta}{\gamma^2}\left\{\mathrm{e}^{-\gamma\left(t_{l+1}^B-t_j^B\right)}\left[\gamma\left(t_{l+1}^B-t_j^B\right)\left(1+\gamma\left(t_{l+1}^B-t_j^B\right)\right)+1\right]\right.$$
$$\left.-\mathrm{e}^{-\gamma\left(t_l^B-t_j^B\right)}\left[\gamma\left(t_l^B-t_j^B\right)\left(1+\gamma\left(t_l^B-t_j^B\right)\right)+1\right]\right\},$$

respectively. For both results the product rule was used. Additionally, we made use of the quotient rule for the latter result.

Next, we calculate the partial derivative w.r.t. the parameter N. For the sum part the derivative is the same as in (4.21), hence we obtain

$$\frac{\partial\ell(\beta,\gamma,N)}{\partial N}=\sum_{i=1}^{n}\frac{i}{N(N-i)}-\sum_{j=1}^{n-1}\sum_{l=j}^{n-1}\frac{l}{N^2}\frac{\beta}{\gamma}\left[\mathrm{e}^{-\gamma(t_{l+1}^B-t_j^B)}\left(-\gamma(t_{l+1}^B-t_j^B)-1\right)\right.$$
$$\left.-\mathrm{e}^{-\gamma(t_l^B-t_j^B)}\left(-\gamma(t_l^B-t_j^B)-1\right)\right] \tag{4.34}$$

as partial derivative w.r.t. N. A lower bound for (4.34) can be constructed in the same way as shown above for equations (4.22) and (4.30). In this case the lower bound is

$$\frac{\partial\ell(\beta,\gamma,N)}{\partial N}\geq\frac{1}{N^2}\left[\frac{n(n+1)}{2}-\sum_{j=1}^{n-1}\sum_{l=j}^{n-1}l\frac{\beta}{\gamma}\left[\mathrm{e}^{-\gamma(t_{l+1}^B-t_j^B)}\left(-\gamma(t_{l+1}^B-t_j^B)-1\right)\right.\right.$$
$$\left.\left.-\mathrm{e}^{-\gamma(t_l^B-t_j^B)}\left(-\gamma(t_l^B-t_j^B)-1\right)\right]\right].$$

4.8.3. Estimation Results

As in the SIR context we have tried to fit the SEIR-related HawkesN process to simulated data. Again, we have estimated the parameters for two cases. In the first case we assumed that all parameters except from N are known. In the second case all parameters were assumed to be unknown. For these two cases we have generated the figures 4.7 and 4.8, respectively.

For these plots the same β/γ combinations as in figures 4.5 and 4.6 were used. The simulations were started with only two populated compartments ($S_0 = 1000$ and $I_0 = 300$) and we set σ to 5. Although this value might seem high, it slows down the spread of the epidemic compared to the SIR model. Thus, we have fewer events in the SEIR simulations. We think that this could be the reason why most of the estimations of N in the two plots 4.7 and 4.8 are far from the true value.

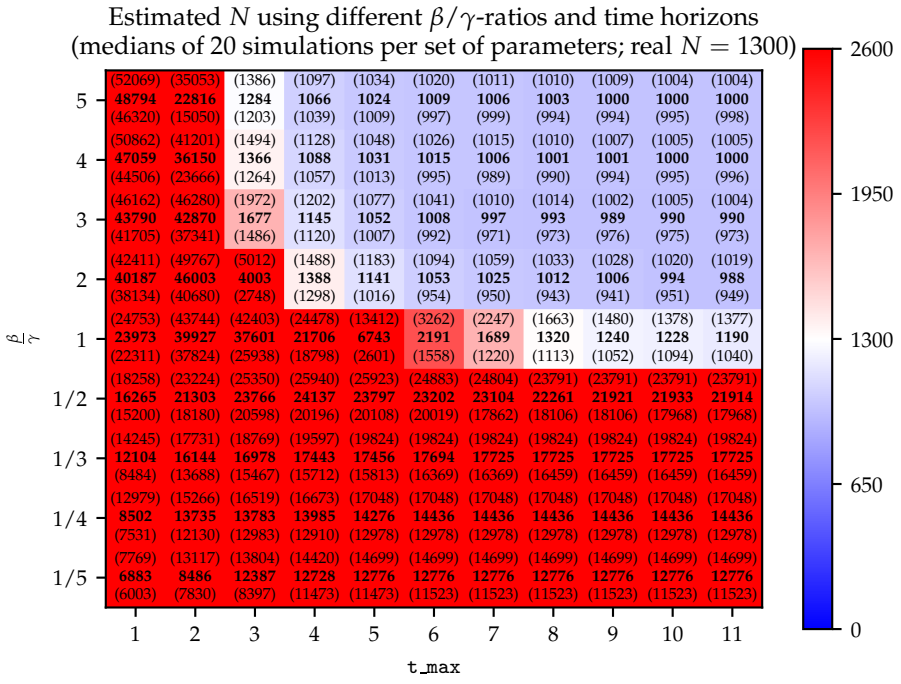


Figure 4.7.: Heat map with estimations of N with known β, σ, and γ.

If one, however, obtains good estimations for the HawkesN process, then the estimated parameters can also be used in the SEIR model. Using this model allows for the prediction of the final size of the epidemic as discussed in subsection 3.5.2.

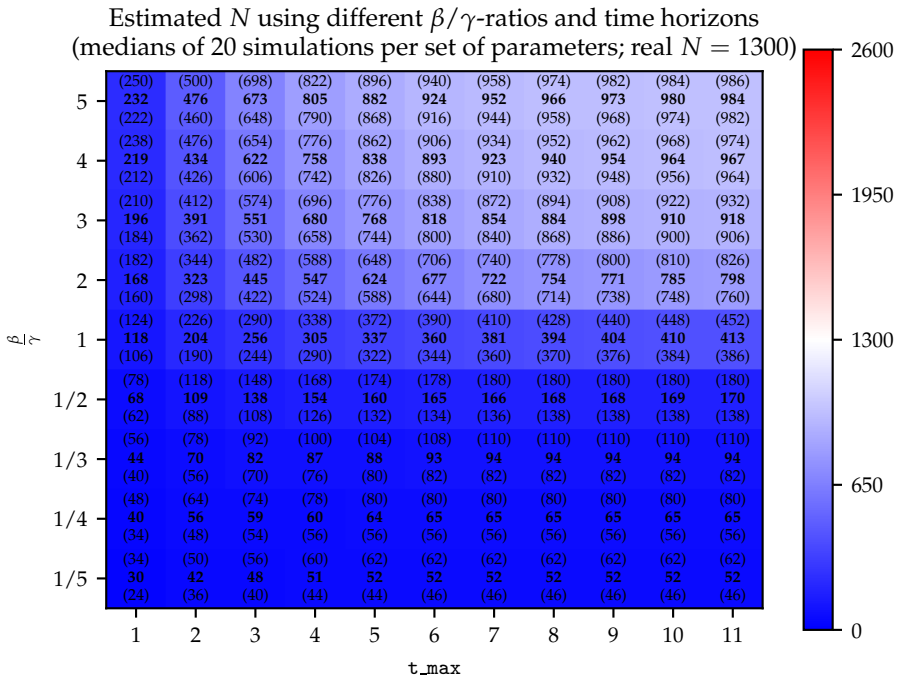


Figure 4.8.: Heat map with estimations of N when maximizing the HawkesN likelihood in all four parameters simultaneously.

5. Conclusion

Rizoiu et al. (2018b) present a link between the SIR model and a variant of the Hawkes process which they call HawkesN process. This link is based on the intensities of the infection processes in the SIR model and that of the HawkesN process. We have extended this link to the SEIR model and show that the corresponding kernel in the intensity of the HawkesN process is a generalization to the one found by Rizoiu et al. (2018b).

For the SEIR related version of the HawkesN we have calculated the likelihood formula as well as its partial derivatives with respect to all parameters, which can be used for likelihood maximization.We have shown that estimating the HawkesN parameters based on an SEIR model rather than an SIR model with the same parameters β and γ leads to worse results. We assume that this is a consequence of the slower spread of the epidemic in the SEIR model and the additional parameter which has to be estimated.

If the data permits HawkesN estimates that are close to the true values, one can use these parameters in the SEIR model. We have shown how this model can generate predictions of the final size of the epidemic. Rizoiu et al. (2018b) argue that the link between the SIR model and the HawkesN process can be utilized to predict information diffusion in social networks like Twitter. With the additional latent period in the SEIR model we expect the corresponding HawkesN process to be more suitable for analyzing moderated communities.

Appendix A.

Code Examples in this Thesis

Rizoiu et al. (2018a) offers R code that can be viewed as the implementation of several of the procedures Rizoiu et al. (2018b) describe in their work. In Karakaš (2019) we provide our own implementation written in Python 3. In the following appendices we have a closer look at the two programs. Hence, we often refer to code snippets in both the R and Python 3 language.

Most of the times when we demonstrate Python 3 code, our presentation of the code mimics a notebook environment as the one offered by the Jupyter project. So we show code inputs and the results produced by them. Code inputs are shown in input cells which are labeled by `In [x]:` where `x` is a number that is incremented by one from one input cell to the next. Results of the code are shown in output cells which are labeled by `Out[x]:` where `x` is the same number as in the corresponding input cell. Example input and output are shown in `[1]`.

```
In [1]: 1 + 1
```

```
Out[1]: 2
```

We assume the code examples are input in the order presented in this thesis. This makes sure that all variables used in one input cell are defined either in the input cell or in one that precedes it.

As mentioned above, we also refer to R code written by Rizoiu et al. (2018a) and we present its usage in interactive mode. To this end the input will be prefixed by a prompt (>) whereas the output lacks a prompt. The following is an example of R used interactively.

```
> 1 + 1
[1] 2
```

Here, `1 + 1` is the input and `2` is the result. Whenever an R input statement spreads over multiple lines, the first line starts with the prompt `>` and the following lines have a `+` at the beginning. The next example shall clarify this.

```
> 5 *
+ 5
[1] 25
```

Here, we multiply 5 by 5 and get 25 as result.

Appendix B.

Working with Epidemic Models Using Software

B.1. Estimation Results with Shorter Time Horizons or Lower Infection Rates

In subsection 3.3.3 we claimed that the R code's results heavily depend on the number of observed infections and that shorter observation periods as well as lower infection rates will thus lead to worse estimation results. To prove this statement we run the code in the Jupyter notebook in Rizoiu et al. (2018a) in this section and compare the results for different values of `t_max` and `beta`. But first, we set a seed to make our results reproducible.

```
> set.seed(0)
```

Next, we load a required library and the scripts of Rizoiu et al. (2018a).

```
> library(parallel)
> source('scripts/functions-SIR-HawkesN.R')
> source('scripts/functions-size-distribution.R')
```

Now we generate 20 simulations.

```
> params.S <- c(N = 1300, I.0 = 300, gamma = 0.2, beta = 1)
> nsim <- 20
> simdat <- replicate(
+     n = nsim,
+     generate.stochastic.sir(params = params.S, Tmax = 11,
+                             hide.output = T)
+ )
```

Next, we estimate the parameters N, β, and γ.

```
> # initial fitting point for each execution
> params.fit.start <- c(N = 0.1, I.0 = 0.1, gamma = 0.1,
+                       beta = 0.1)
>
> .cl <- makeCluster(spec = min(nsim, detectCores()),
+                    type = 'FORK')
> results <- parSapply(cl = .cl, X = 1:nsim,
+     FUN = function(i) {
+         mysim <- as.data.frame(simdat[, i])
+         return(fit.stochastic.sir(mysim, params.fit.start))
+ })
> stopCluster(.cl)
>
> # reconstruct result data format
> res <- as.data.frame(results[1,])
> names(res) <- 1:nsim
> res <- as.data.frame(t(res))
>
> complete_res <- res
```

The summary of the estimation results can be shown also follows.

```
# let's see how well parameters were retreived
> prnt <- rbind(params.S[c('N', 'I.0', 'gamma', 'beta')],
+               apply(X = complete_res[,
+                   c('N', 'I.0', 'gamma', 'beta')],
+                   MARGIN = 2, FUN = median),
+               apply(X = complete_res[,
+                   c('N', 'I.0', 'gamma', 'beta')],
+                   MARGIN = 2, FUN = sd))
```

```
> rownames(prnt) <- c('theoretical', 'median', 'sd')
> print(prnt[, c('N', 'I.0', 'gamma', 'beta')], digits = 2)
                 N I.0  gamma  beta
theoretical 1300.0 300 0.2000 1.000
median      1282.5 300 0.1986 0.993
sd             5.7   0 0.0063 0.033
```

With an estimate of 1282.5 for N we are near the true value of 1300. It is only slightly worse than the results presented by Rizoiu et al. (2018a) which we also show in table 3.1. The only difference between this table and the results below the last code input is the seed that we set but which was absent in Rizoiu et al. (2018a).

If we run all the R inputs again (starting from `set.seed(0)`) but change `Tmax = 11` to `Tmax = 5` as the argument to the function `generate.stochastic.sir`, then we get the results in table B.1.

	N	β	γ
theoretical	1300	1.000	0.200
median	1197	0.928	0.2003
standard deviation	13	0.037	0.0091

Table B.1.: Summary of estimation results of the R code with a shorter time horizon.

With `Tmax = 11` again but $\beta = 0.5$ instead of $\beta = 1$ the R code yields the results shown in table B.2.

	N	β	γ
theoretical	1300	0.500	0.200
median	1096	0.420	0.203
standard deviation	25	0.025	0.007

Table B.2.: Summary of estimation results of the R code with a reduced infection rate.

In a nutshell, the estimation of N deteriorated remarkably when either the simulations' time horizon was shortened or when the infection rate was decreased. This is the behavior that we projected in subsection 3.3.3.

B.2. Likelihood with R when Disease Dies out

In subsection 3.3.3 we maintained that the R code written by Rizoiu et al. (2018a) would not be capable of producing the heat map in figure 3.9. This is because the code cannot compute the likelihood correctly if the disease in the provided observation has died out. This can easily be seen by trying to calculate the log-likelihood of such a simulation. To this end we save a simulation where the disease dies out in [2] using the `feather` serialization library. The last line in input cell [2] only serves demonstration purposes. It produces the output which shows that there are no infected individuals left at the end of the observation.

In [2]:
```
from py_hawkesn_sir.py_hawkesn_sir.sir_stochastic \
    import StochasticSIR
s_0 = 80
i_0 = 20
r_0 = 0
t_max = 11
n_simulations = 20
random_state = 0
model_tmp = StochasticSIR(s_0=s_0,
                          i_0=i_0,
                          r_0=r_0,
                          beta=1,
                          gamma=.2)
model_tmp.simulate(t_max=t_max,
                   n_simulations=n_simulations,
                   random_state=random_state)
model_tmp.data_[0].to_feather("obs_disease_dies_out")
model_tmp.data_[0].tail()
```

Out[2]:

	i	r	s	t
416	4	356	940	5.403488
417	3	357	940	5.526455
418	2	358	940	5.976188
419	1	359	940	6.064920
420	0	360	940	6.973503

We can then load the data frame in R. The following interactive use of R shows that the calculation of the log-likelihood fails, making it impossible to fit parameters for such an observation.

```
> library(feather)
> observation <- read_feather('../obs_disease_dies_out')
> colnames(observation) <- c("I", "R", "S", "time")
> stochastic.sir.complete.neg.log.likelihood(
+     c(1000, 300, 0.2, 0.1),
+     observation
+ )
neg.ll
Inf
```

However, using only one part of the observation such that at least one individual remains infected at the end leads to a finite result. This can be seen in the following lines of code.

```
> observation.short <- observation[1:400, ]
> stochastic.sir.complete.neg.log.likelihood(
+     c(1000, 300, 0.2, 0.1),
+     observation.short
+ )
neg.ll
-1134.18
```

If the log-likelihood is $-\infty$ for an observation irrespective of the parameters, then it is impossible to fit the parameters. Only if there are parameter combinations for which the log-likelihood function produces finite results, an optimization is possible.

Appendix C.

Working with HawkesN Processes Using Software

C.1. The Intensity of a HawkesN Process Using Software

C.1.1. Calculation of the Intensity

In this section we demonstrate how our Python program can be used to calculate the intensity of a HawkesN process in the form of (4.6). We also verify the correctness of the calculations using a small example and contrast the results with those produced by the R program of Rizoiu et al. (2018a).

We start by the calculation of the intensity using our Python program. To this end we first instantiate the `HawkesN` class and assign the created object to the variable `hn`.

```
In [3]: import numpy as np
        from py_hawkesn_sir.py_hawkesn_sir import hawkesn
        hn = hawkesn.HawkesN()
```

Next, we define our example data. We will assume that the scale and decay parameters equal 0.5 and that the population size is 100. To make our example results easy to verify we choose a short history containing only the three event times 0, 1, and 2.

```
In [4]: his = np.array([0, 1, 2])
        scale = decay = 0.5
        n = 100
```

To obtain a function for calculating the exponential intensity with our parameters from [4], we can pass these parameters to the `exp_intensity` method. After obtaining the function in [5] we evaluate it at time 2. This roughly results in 0.48.

```
In [5]: int_fun = hn.exp_intensity(scale=scale,
                                   decay=decay,
                                   n=n,
                                   history=his)
        int_fun(2)
```

```
Out[5]: array([ 0.47879445])
```

Now let us verify that this result is correct. Following equation (4.6) and inserting the parameters from [4] we obtain

$$\begin{aligned}\lambda_2^H &= \left(1 - \frac{3}{N}\right) \kappa\theta \left(\mathrm{e}^{-\theta\cdot 2} + \mathrm{e}^{-\theta\cdot 1}\right) \\ &= 0.97 \cdot 0.5 \cdot 0.5 \cdot \left(\mathrm{e}^{-1} + \mathrm{e}^{-0.5}\right) \approx 0.236. \end{aligned} \tag{C.1}$$

This result differs from the 0.48 our program calculated. That is, because the Python code mimics the R program of Rizoiu et al. (2018a) by default. As a consequence, `exp_intensity` by default calculates the intensity with a $t_j^C \leq t$ as sum index instead of $t_j^C < t$ as in equation (4.6). To get the latter behavior we need to pass an additional argument to `exp_intensity` as shown in [6].

```
In [6]: int_fun_less = hn.exp_intensity(scale=scale,
                                        decay=decay,
                                        n=n,
                                        history=his,
                                        sum_less_equal=False)
        int_fun_less(2)
```

```
Out[6]: array([ 0.23629445])
```

Now we get the same result as in our calculation by hand in equation (C.1).

To calculate the intensity using the R program written by Rizoiu et al. (2018a), we can do the following.

```
> source('scripts/functions-SIR-HawkesN.R')
> make.history.compatible.to.HawkesN <- function(his){
+   df <- as.data.frame(list(rep("dummy", length(his)), his))
+   colnames(df) <- c("dummy_col", "time")
+   return(df)
+ }
>
> his <- make.history.compatible.to.HawkesN(c(0,1,2))
> t <- 2
> lambda(t, his, c(K=0.5, c=0.001, theta=0.5, N=100))
[1] 0.4787944
```

As mentioned above, this is the same result as the one produced by our Python program without the `sum_less_equal` argument set to `False`. Note that for the R function to work we needed to supply it with a `data.frame` consisting of two columns with the event times being in the second column. We defined the function `make.history.compatible.to.HawkesN` to encapsulate the construction of this `data.frame`.

C.1.2. Plotting the Intensity

Both the Python and the R program can be used to plot a process' intensity. The Python input in [7] creates the plot shown in figure C.1.

```
In [7]: t_max = 10
        hn.plot_exp_intensity(t_max, scale=scale, decay=decay,
                              n=n, history=his)
```

If we want to suppress the default behavior which mimics the R program and we are rather interested in the intensities according to equation (4.6), we can again supply `sum_less_equal=False` as argument. This is shown in [8].

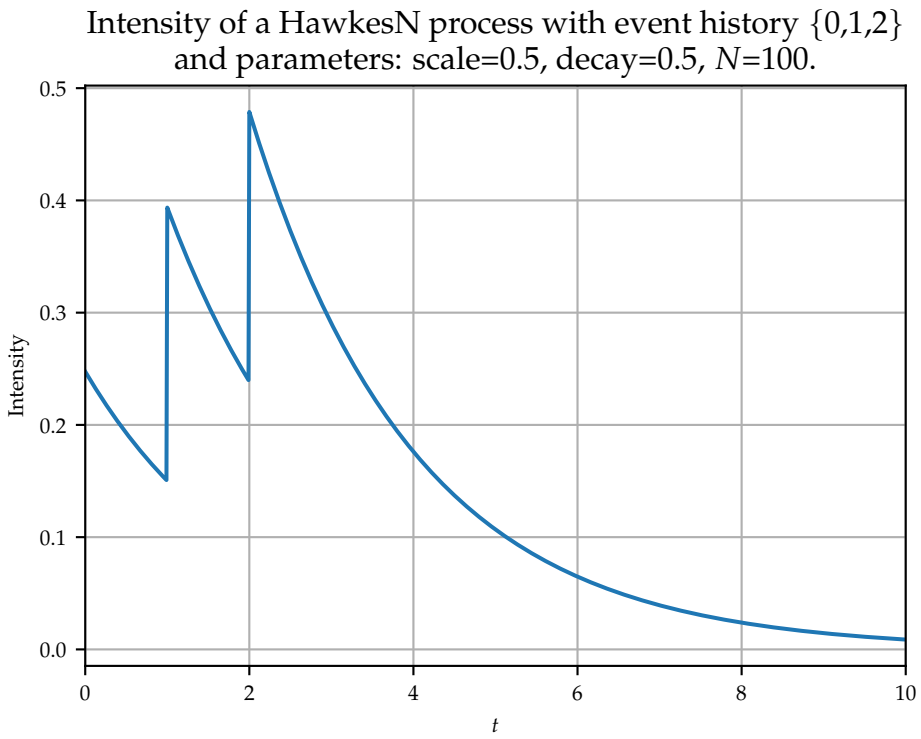


Figure C.1.: Intensity of an example HawkesN Process calculated with our Python code mimicking the R code.

```
In [8]: hn.plot_exp_intensity(t_max, scale=scale, decay=decay,
                              n=n, history=his,
                              sum_less_equal=False)
```

This creates figure 4.2 which differs from figure C.1 only in the event times.

In the following we also offer the R code for generating an intensity plot. Figure C.2 shows the result.

```
> library(ggplot2)
> intensity <- function(t){
+   return(lambda(t, his, c(K=0.5, theta=0.5, N=100)))
+ }
> ggplot(data.frame(x=c(0, 10)), aes(x=x)) +
+   theme(text=element_text(family="Palatino")) +
+   stat_function(fun=intensity) +
+   xlab('t') + ylab("Intensity") +
+   scale_x_continuous(breaks=seq(0, 10, 2)) +
+   ggtitle("Intensity using the R code")
```

C.2. Calculating the Log-Likelihood of a HawkesN Process using Software

C.2.1. Calculation of the Log-Likelihood

In this section we continue our Python and R sessions from section C.1. Now we calculate the log-likelihood of our example HawkesN process with the parameters $\kappa = \theta = 0.5$, $N = 100$ and the event history $\{0, 1, 2\}$.

The R code by Rizoiu et al. (2018a) calculates the log-likelihood times -1. The following code shows the use of the corresponding function. Note that the parameters `K`, `theta`, and `N` correspond to κ, θ, and N, respectively. The parameter `c` which we set to zero is not used by the function.

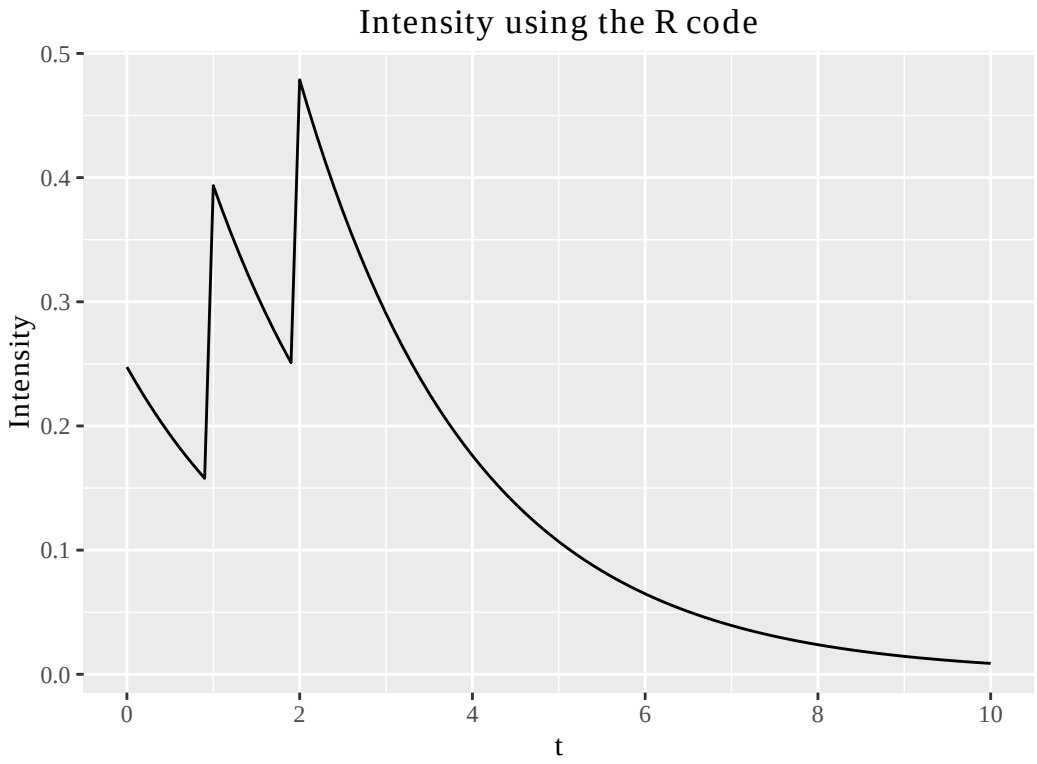


Figure C.2.: Intensity of an example HawkesN Process calculated with the R code of Rizoiu et al. (2018a).

```
> neg.log.likelihood(c(K=0.5, c=0, theta=0.5, N=100),
+                    history = his)
[1] 2.17341
```

We can replicate the result of 2.17341 using our Python program. Since it calculates the log-likelihood without the additional factor of -1 we would expect a result of -2.17341. Output cell [9] shows that the two programs indeed produce equivalent results.

```
In [9]: hn.llf(scale=scale, decay=decay, n=n , history=his)

Out[9]: -2.1734103794884079
```

We recall from section C.1 that the R code and thus our Python code use a slight variation of the intensity formula in their calculations. This adaption causes differences to the true intensity only when the function argument is an event time. Since in the calculation of the log-likelihood the intensity is evaluated in the event times, this leads to a miscalculation. To get the true log-likelihood, we pass `sum_less_equal=False` as keyword argument to our Python method as shown in [10].

```
In [10]: hn.llf(scale=scale, decay=decay, n=n , history=his,
                sum_less_equal=False)

Out[10]: -3.8536800501301425
```

Note that the difference in results stems from only one of the two parts of the likelihood formula. We referred to this part as the sum part in section 4.7. The integral part is not affected because the integrand is only changed on a finite set of points in time.

C.2.2. Calculation of the Gradient

Both the Python and the R program use the L-BFGS algorithm in their fitting procedures and both provide the algorithm with the gradient of the objective function (i.e. the log-likelihood times -1). In doing so both

programs avoid the numerical calculation of the gradient. In this section we review the results for the gradient. Again, we start with the R program by Rizoiu et al. (2018a).

```
> closedGradient(c(K=0.5, c=1, theta=0.5, N=100),
+                history = his)
             K              c          theta              N
-2.9909872055  0.0000000000 -0.9617048023 -0.0002222062
```

Calculating the derivative of the log-likelihood function with respect to κ in our Python program gives around 2.99 as [11] shows. Note that the R program calculates the gradient of the log-likelihood function times -1, so the two results are equivalent.

```
In [11]: hn.dllf_dscale(scale=scale, decay=decay, n=n,
                        history=his)
```

```
Out[11]: 2.9909872054635205
```

Next, let us compute the derivate of the log-likelihood function with respect to θ. In the R program the result was roughly -0.96. [12] shows our implementation.

```
In [12]: hn.dllf_ddecay(scale=scale, decay=decay, n=n,
                        history=his)
```

```
Out[12]: 2.2818614701258939
```

Here we have a clear difference in results. This is also the case for the derivative with respect to N. As we see in [13] our program yields approximately 0.00053, whereas the R program computes a result of around 0.00022.

```
In [13]: hn.dllf_dn(scale=scale, decay=decay, n=n,
                    history=his)
```

```
Out[13]: 0.00043047446027130102
```

To determine which of the two programs is right, we developed a second implementation of the `HawkesN` class which uses `sympy`. Starting from a

log-likelihood method that produces the same results as the R and the first Python implementation, we can use `sympy` to calculate the derivative. [14] demonstrates that this second implementation produces the same results for the log-likelihood.

```
In [14]: from py_hawkesn_sir.py_hawkesn_sir import hawkesn_sympy
         from sympy import Array
         params = [("scale", scale), ("decay", decay), ("n", n)]
         hn_sym = hawkesn_sympy.HawkesN(Array(his))
         hn_sym.llf().subs(params)
```

```
Out[14]: -2.17341037948841
```

In [15] we see that the gradient calculated symbolically by `sympy` equals the one produced by our first implementation which does not make use of `sympy` for performance reasons. This underlines the correctness of the derivatives computed by our first implementation.

```
In [15]: hn_sym.llf_gradient().subs(params)
```

```
Out[15]: [2.99098720546352, 2.28186147012589,
             0.0004304744602713 01]
```

So far we have dealt with the gradient of the log-likelihood as calculated by R program. In C.1 and C.2.1 we claimed that the intensity function used in these calculation is not accurate. In order to use the correct intensity function, we will again pass `sum_less_equal=False` as keyword argument.

```
In [16]: hn.dllf_dscale(scale=scale, decay=decay, n=n,
                        history=his, sum_less_equal=False)
```

```
Out[16]: 2.99098720546
```

```
In [17]: hn.dllf_ddecay(scale=scale, decay=decay, n=n,
                        history=his, sum_less_equal=False)
```

```
Out[17]: 0.961704802296
```

```
In [18]: hn.dllf_dn(scale=scale, decay=decay, n=n,
                    history=his, sum_less_equal=False)
```

```
Out[18]: 0.000430474460271
```

Comparing [16] with [11] and [18] with [13] we see that the derivatives with respect to κ and to N were not affected by changing the calculation of the intensity. That is because the sum contained in the formula for the intensity was canceled out in equations 4.19 and 4.22.

In equation 4.20, however, this sum still exists which means that the result will be affected by the choice of whether to sum over $t_j^C < t$ or $t_j^C \leq t$. That is why [17] holds a different result than [12]. We can verify that the results in [16]–[18] are correct by using our `sympy`-powered implementation. [19] serves as confirmation.

```
In [19]: hn_sym.llf_gradient(sum_less_equal=False).subs(params)
```

```
Out[19]: [2.99098720546352 0.961704802296088
          0.0004304744602713 01]
```

As a final remark we want to point out that the value of approximately 0.96 obtained with `sum_less_equal` set to `False` equals the result in R. That shows that the R code is calculating a derivative which does not belong to the intended function.

Bibliography

Aalen, Odd, Ornulf Borgan, and Hakon Gjessing (2008). *Survival and event history analysis: a process point of view*. Springer Science & Business Media. ISBN: 9780387202877 (cit. on pp. 6, 7, 9).

Allen, Linda JS (2008). "An introduction to stochastic epidemic models." In: *Mathematical Epidemiology*. Springer-Verlag Berlin Heidelberg, pp. 81–130. ISBN: 9783540789109 (cit. on pp. 13, 17, 20).

Andersen, Per Kragh et al. (1996). *Statistical Models Based on Counting Processes*. Springer Series in Statistics. Springer New York. ISBN: 9780387945194 (cit. on pp. 5, 8).

Bacry, Emmanuel, Iacopo Mastromatteo, and J.F. Muzy (2015). "Hawkes Processes in Finance." In: *arXiv preprint arXiv:1502.04592* (cit. on p. 9).

Brauer, Fred (2008). "Compartmental Models in Epidemiology." In: *Mathematical Epidemiology*. Springer-Verlag Berlin Heidelberg, pp. 19–79. ISBN: 978-3-540-78911-6 (cit. on p. 13).

Daley, Daryl J. and David Vere-Jones (2003). *An introduction to the theory of point processes. Volume I: Elementary Theory and Methods*. Springer-Verlag New York. ISBN: 978-1-4757-8109-0 (cit. on pp. 7, 52, 54, 69).

De la Sen, M, S Alonso-Quesada, and A Ibeas (2011). "A SEIR Epidemic Model with Infectious Population Measurement." In: *Proceedings of the World Congress on Engineering*. Vol. 3 (cit. on p. 42).

Feng, Zhilan, Dashun Xu, and Haiyun Zhao (Aug. 2007). "Epidemiological Models with Non-Exponentially Distributed Disease Stages and Applications to Disease Control." In: *Bulletin of mathematical biology* 69, pp. 1511–36 (cit. on p. 42).

Karakaš, Aleksandar (2019). *Github repository accompanying "Linking Epidemic Models and Hawkes Processes"*. URL: https://github.com/yogabonito/seir_hawkes (visited on 02/21/2019) (cit. on pp. 17, 37, 43, 47, 91).

Kingman, John Frank Charles (1992). *Poisson processes*. Vol. 3. Oxford Studies in Probability. Clarendon Press Oxford (cit. on p. 21).

Klenke, A. (2013). *Probability Theory: A Comprehensive Course*. 2nd ed. Springer London. ISBN: 9781447153603 (cit. on pp. 3, 5, 51).

Laub, Patrick J, Thomas Taimre, and Philip K Pollett (2015). "Hawkes processes." In: *arXiv preprint arXiv:1507.02822* (cit. on pp. 9, 54).

Li, Michael Y et al. (1999). "Global dynamics of a SEIR model with varying total population size." In: *Mathematical biosciences* 160.2, pp. 191–213 (cit. on pp. 42, 64).

Martcheva, M. (2015). *An Introduction to Mathematical Epidemiology*. Springer US. ISBN: 9781489976123 (cit. on pp. 12, 16).

Resnick, S.I. (1992). *Adventures in Stochastic Processes*. Birkhäuser Boston. ISBN: 9780817635916 (cit. on p. 7).

Rizoiu, Marian-Andrei et al. (2018a). *Github repository accompanying "SIR-Hawkes: Linking Epidemic Models and Hawkes Processes to Model Diffusions in Finite Populations"*. URL: https://github.com/computationalmedia/sir-hawkes (visited on 10/09/2018) (cit. on pp. 2, 21, 27, 29, 31, 33, 34, 71, 74, 91, 93, 95, 96, 99–101, 103, 104, 106).

Rizoiu, Marian-Andrei et al. (2018b). "SIR-Hawkes: Linking Epidemic Models and Hawkes Processes to Model Diffusions in Finite Populations." In: *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, pp. 419–428 (cit. on pp. 1, 21, 27, 31, 51, 55–57, 61, 71, 73, 74, 89, 91).

Rüschendorf, L. (2014). *Mathematische Statistik*. Springer Spektrum Berlin Heidelberg. ISBN: 9783642419966 (cit. on p. 6).

Yan, Ping (2008). "Distribution Theory, Stochastic Processes and Infectious Disease Modelling." In: *Mathematical Epidemiology*. Springer-Verlag Berlin Heidelberg, pp. 229–293. ISBN: 9783540789109 (cit. on pp. 17, 18).

Zhu, Ciyou et al. (Dec. 1997). "Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-scale Bound-constrained Optimization." In: *ACM Trans. Math. Softw.* 23.4, pp. 550–560. ISSN: 0098-3500. DOI: 10.1145/279232.279236. URL: http://doi.acm.org/10.1145/279232.279236 (cit. on p. 27).