Universiteit Utrecht

A BACHELOR THESIS SUBMITTED FOR THE DEGREE OF

BSc Physics & Astronomy

BSc Mathematics

Mathematical Institute

Parametric Hawkes Processes:

Theory, Simulation and Application

Author:
E.J. van de Winkel

Supervisor:
Dr. C. Spitoni

June 10, 2019

Introduction

When modelling a real life situation, we often want to use information from the past to better predict future values. In some situations past events raise the likelihood for future events to occur, so information about the past is necessary to set up a good model. Processes that follow this pattern, are called self-exciting processes. Self-exciting processes are widespread: an earthquake is modelled as a self-exciting process, such are several option related models.

In this Thesis we will be looking at the Hawkes process, which is a self-exciting process. The Thesis is divided into four parts.

Part I is about stochastic processes. A Hawkes process is an example of a stochastic process, so before we are able to describe the Hawkes process, we should know what stochastic processes are and what important type of stochastic processes we have. Special attention will be given to the homogeneous and nonhomogeneous Poisson processes, since Hawkes processes are related to these.

Part II is about Hawkes processes. We define what Hawkes processes are in both the one-dimensional and multidimensional case. We will look at the stationarity conditions and will see how the expected number of events for a one-dimensional Hawkes process can be calculated. Besides, we will look at how the parameters in a Hawkes process can be calculated using the maximum-likelihood estimates.

Part III is about the simulation of Poisson and Hawkes processes. If a sample of events have already occured, simulation studies can help us to predict the continuation of the process based on the events already occured. Besides, simulations can help us to verify theoretical findings in a more visual way. This Part forms the core of the Thesis.

In Part IV of this Thesis two real life applications of Hawkes processes will be discussed. We will look at the use of Hawkes processes in finance and in the description of social media. The two Chapters in this Part are based on existing papers, so no own models have been set up. These Chapters are mainly there to give a quick look at how Hawkes processes can be used in real life situations. Besides, in these two Chapters we will use the knowledge acquired in the previous Chapters, so it can also be seen as a refresher of the theory dealt with in mostly Part I and Part II of this Thesis.

Contents

Part I

Stochastic Processes

Chapter 1

Measure Theory of Stochastic Processes

In this Thesis we are going to look at Hawkes processes. The Hawkes process is an example of a stochastic process, so in order to be able to describe Hawkes process, we should first have a understanding of what a stochastic process is.

In this Chapter we will define a stochastic process from a measure theoretical point of view. Besides given the definition we will look at two concepts of stochastic processes that are important in the following of this Thesis.

Readers who are not familiar with the basics of measure and probability theory are encouraged to first go to Appendix A and Appendix B to learn more about these topics.

1.1 Defining Stochastic Processes

The core of this Chapter lies in defining what a stochastic process exactly is. Before giving a precise definition it is good to first denote that a stochastic process is a set of random variables, where each random variable represents the value of the process at a specific time. To know at which times we are looking at, we always first denote the time index set, which we will define first.

Definition 1 ([8], Definition 3.1.1.): The time index set T is the collection of times at which we observe random outcomes.

Although we have defined T here as a time index set, we could have defined T more generally as an index set, since a stochastic processes is not necessarily about time processes. We could have also looked at the size of a certain population, for different generations, so in this case we could have set $T = \{1, 2, 3, 4, ...\}$, where each $t \in T$ represents the t-th generation of the population. However, since we are looking at time processes in this Thesis, T will from now on always represent a time index set.

For the time index set T there are two options. We could be looking at a continuous time interval T (in this case we are looking at a continuous-time process), or T could be a set of integers (in this case we are looking at a discrete-time process). ([8], Definition 3.1)

Now that we have defined what a time index set is, we are able to define a stochastic process.

Definition 2 ([1], Definition 6.3.1.): A (real valued) stochastic process with [time] index set T is a family $\{X_t : t \in T\}$ of random variables defined on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$.

A stochastic process can thus be thought of as a family of random variables that describes the evolution through time of some (physical) process, thus all random variables are defined on the same probability space.

If X_t is given, than we say that X_t is the state of the process at time t. ([26], page 77) For example, X_t can be the number of cars driving on a road at time t.

A definition linked to the state of a process, is that of a state space.

Definition 3 ([26], in text, page 78): The state space of a stochastic process is defined as the set of all possible values that the random variables X_t can assume.

1.2 Filtrations

In the definition of Hawkes processes, which are the main interest of this Thesis, the history of the process up to a certain time is important. Without going into details about the specific definition of a Hawkes process yet, we will now define a filtration, a concept that is the same as the history in the definition of Hawkes processes.

Definition 4 ([9], in text, page 345): Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space. A filtration is a sequence $\{\mathcal{F}_n\}_{n=0}^{\infty}$ of sub-σ-algebras of $\mathcal{A}$ that is increasing, in the sence that $\mathcal{F}_n \subseteq \mathcal{F}_{n+1}$ holds for each n.

A filtration can be seen as the history of a process. The elements in the σ-algebra $\mathcal{F}_n$ can be seen as all the events up to time n. The elements in $\mathcal{F}_n$ can be seen as all the events that are determined by the random variables X_0, X_1, ..., X_n. The next example shows a more concrete example of the fact that a filtration can be seen as the history of a process.

Example 1: As in the Examples in Appendix B, we will look at the example of tossing two fair contains. As can be seen in Example B1, the sample space of the experiment is $\Omega = \{HH, HT, TH, TT\}$. As σ-algebra on Ω we choose $\mathcal{A} = \{\emptyset, \{HH\}, \{HT\}, \{TH\}, \{TT\}, \{HH, HT\}, \{HH, TH\}, \{HH, TT\}, \{HT, TH\}, \{HT, TT\}, \{TH, TT\}, \{HH, HT, TH\}, \{HH, HT, TT\}, \{HH, TH, TT\}, \{HT, TH, TT\}, \{HH, HT, TH, TT\}\}$, as we did in Example B2.

Right now there are three times of interest: $t = 0$ (have not tossed a coin yet), $t = 1$ (after having tossed one coin) and $t = 2$ (after having tossed both coins).

At time $t = 0$ we have no information about which event will occur. The only two things we know for certain now yet is that $\emptyset$ would not occur and that Ω will occur for sure, so $\mathcal{F}_0 = \{\emptyset, \Omega\}$.

At time $t = 1$ we have information about the first toss. Say we throw heads at $t = 1$, so only HT and HH can occur, so $\mathcal{F}_1 = \{\emptyset, \{HH\}, \{HT\}, \{HH, HT\}, \{HH, TH\}, \{HH, TT\}, \{HT, TH\}, \{HT, TT\}, \{HH, HT, TH\}, \{HH, HT, TT\}, \{HH, TH, TT\}, \{HT, TH, TT\}, \{HH, HT, TH, TT\}\}$.

At time $t = 2$, we know for certain which event has occured, so $\mathcal{F}_2 = \mathcal{A}$.

We see $\mathcal{F}_0 \subseteq \mathcal{F}_1 \subseteq \mathcal{F}_2$, and $\mathcal{F}_0$, $\mathcal{F}_1$, and $\mathcal{F}_2$ are all σ-algebras, thus $\{\mathcal{F}_0, \mathcal{F}_1, \mathcal{F}_2\}$ is a filtration.

1.3 Stopping Time

Another definition that is closely connected to that of the filtration is the definition of the stopping time.

Definition 5 ([9], in text, page 345): Let $\{F_n\}$ be a filtration on the probability space $(\Omega, \mathcal{A}, \mathbb{P})$. A stopping time (or an optional time) is a function $\mathcal{T} : \Omega \to \mathbb{N}_0 \cup \{\infty\}$ such that $\{\mathcal{T} \leq n\} \in \mathcal{F}_n$.

The formal definition of the stopping time is more difficult than the interpretation.

Say we are having a stochastic process with time index set T defined on the probability space $(\Omega, \mathcal{A}, \mathbb{P})$, and a filtration on $(\Omega, \mathcal{A}, \mathbb{P})$. We are observing random variables X_0, X_1, ..., where $X_i \in \mathcal{F}_i$ (for $i = \{1, 2, ...\}$). At time $\mathcal{T}$ we want to stop observing the random variables. It is important to say that $\mathcal{T}$ does not really have to be a numerical value, but can also be a rule like “two times heads has been thrown” in the experiment of throwing a fair dice. Say event n occurs at time $\mathcal{T}$. The decision to stop observing at the n-th event, depends only on information from the previous events X_0, ..., X_{n-1} and not on future events X_{n+1}, ..., thus $\{\mathcal{T} \leq n\} \in \mathcal{F}_n$. ([9], page 345)

Example 2: To give a more concrete example of what a stopping time is and also what is not a stopping time, we look at playing roulette. Say we always play on red. Playing exactly two games is a stopping time, since no information of the future is needed. Playing until all money is lost is also a stopping time, since no future information is relevant as soon as you are broke. However, playing until you have maximized your profit is not a stopping time, since future information information is needed to determine whether the profit is maximized.

Chapter 2

Toolbox for Stochastic Processes

Now we know what stochastic processes are, we are going to look at some of the tools needed to work with stochastic processes. These tools form the basis for our analysis of Hawkes processes. We will also define the point process, the counting process and the regular point process in this Chapter.

2.1 Point Processes

Before we can look at some of the tools needed in the description of stochastic and Hawkes processes, we have to define what a point process is, since point processes are used in some of the Definitions, Propositions and Theorems following.

> **Definition 6 ([21], Definition 2):** If a sequence of random variables $T = \{T_1, T_2, ...\}$, taking values in $[0, \infty)$, has $\mathbb{P}(0 \leq T_1 \leq T_2 \leq ...) = 1$, and the number of points in a bounded region is almost surely [(see Appendix B2)] finite, then T is a point process.

Based on Definition 6, we see that a point process is just a sequence of sorted random variables. We call a point process simple if and only if $T_i < T_{i+1}$ for all values of i. ([3], Definition 1)

The set T is mostly a set with elements that corresponds to the time a certain event has taken place. The element T_i is the event time of the i-th event. The event time is the time of occurence of the i-th event. The event time is sometimes called the arrival/waiting time.

2.2 Counting Process

Besides the point process, we also need to define a counting process.

Definition 7 ([21], Definition 1): A counting process is a stochastic process $\{N(t) : t \geq 0\}$ taking values in $\{0, 1, 2, 3, ...\}$ that satisfies $N(0) = 0$, is almost surely finite, and is a right-continuous step function with increments of size $+1$.

The point process we have just defined, can be rearranged very easily into a counting process.

Whereas a point process is a collection of times at which certain event occurs, a counting process represents the total number of events that have occured at a certain time.

The number of events occured at time t for the counting process $\{N(t), t \geq 0\}$ can be calculated as

$$N(t) = \sum_{i \geq 1} \mathbb{I}_{t \geq T_i},$$

where $\mathbb{I}$ is the indicator function defined as

$$\mathbb{I}_{t \geq T_i} = \begin{cases} 0 & \text{if } T_i > t, \\ 1 & \text{if } T_i \leq t, \end{cases}$$

so a value of one is assigned to all the events that have occured before t and a value of zero is assigned to all the other events. ([25], page 3) In this way exactly all the events that have occured before time t will be counted, thus $N(t)$ indeed represents the number of events occured at time t.

The connection between the point process and the counting process is shown in Figure 2.1.

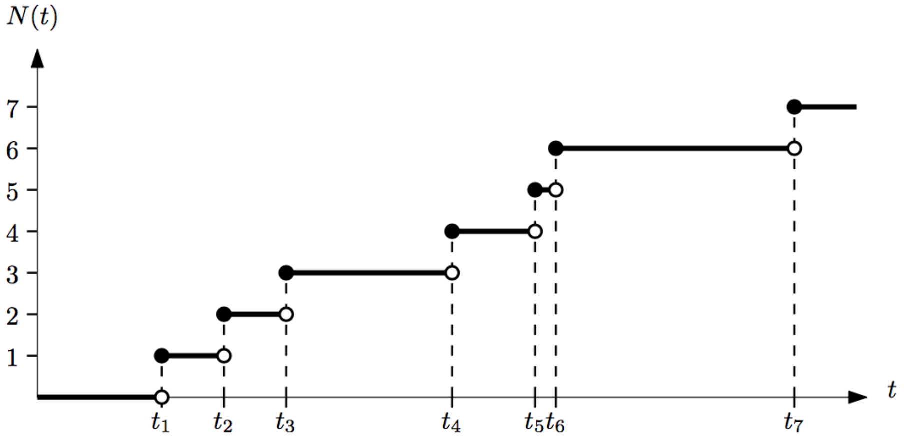


Figure 2.1: In this graph a graphical representation of both a point and counting process can be seen. Is this process seven events have occured, and their event times are denoted on the horizontal axis by t_i. A set containing $\{t_1, ..., t_7\}$ can be seen as a point process. On the vertical axis we see the number of events counted, so this is the corresponding counting process $N(t)$. (Figure taken from [21], Figure 1)

Based on the information of the counting process, we see a counting process must satistfy the following four rules: ([26], page 297)

- $N(t) \geq 0$,
- $N(t)$ is integer valued,
- if $s < t$, then $N(s) \leq N(t)$,
- for $s < t$, $N(t) - N(s)$ equals the number of events that occur in the interval $(s, t]$.

Besides the conditions listed above, a counting process is said to possess independent increments. A process is said to possess independent increments if the number of events that occur in disjoint time interval are independent. ([26], page 297)

Independent increments prove to be very useful in the definition of the Poisson process in Chapter 3.

2.3 Conditional Intensity Function: Part I

In Section 2.1 we have defined what point processes are, but for some of the Definitions, Propositions and Theorems following, we want to define regular point processes. However, in order to define regular point processes we first have to define what a conditional intensity function is. As the word conditional suggests, the conditional intensity function depends explicitely on the past. By $\mathcal{H}_{t-}$ we denote the σ-algebra of all the events that have occured at times up to but not including t. ([10], page 232) We call $\mathcal{H}_{t-}$ the history of the process, and this history is a filtration (Definition 4).

For the conditional intensity function, which is denoted by $\lambda^*(t|\mathcal{H}_{t-})$, we have $\lambda^*(t|\mathcal{H}_{t-})dt \approx E[N(dt)|\mathcal{H}_{t-}]$, so the conditional intensity function can be thought of as the expected rate of arrivals at t, given the realization of the process before t. ([10], page 232)

In most literature the history of the process is not denoted explicitely. Instead, an superscript asterisk is used. (This is for example done in [10].) So we will write $\lambda^*(t)$ instead of $\lambda^*(t|\mathcal{H}_{t-})$.

Definition 8 ([21], Definition 3): Let $N(.)$ be a counting process with associated histories $\mathcal{H}_{.}$. If a (non-negative) function $\lambda^*(t)$ exists such that

$$\lambda^*(t) = \lim_{h \downarrow 0} \frac{E[N(t+h) - N(t)|\mathcal{H}_{t-}]}{h}$$

which only relies on information of $N(.)$ in the past (that is, $\lambda^*(t)$ is $\mathcal{H}_{t-}$-measurable), then it is called the conditional intensity function of $N(.)$.

We call a process nonstationary if the conditional intensity function of the process depends explicitely on time, otherwise we call a process stationary. ([17], in text, page 4) We will come back to the concept of (non)stationarity in more detail in Section 4.3 and Subsection 4.4.3.

2.4 The Little o-Notation

The last step before we can define the regular point processes, is defining the little o-notation.

Definition 9 ([26], Definition 5.1): A function $f(.)$ is said to be $o(h)$ if

$$\lim_{h\to 0} \frac{f(h)}{h} = 0.$$

It may be clear that, for example, $f(h) = h^3$ is $o(h)$ since

$$\lim_{h\to 0} \frac{f(h)}{h} = \lim_{h\to 0} \frac{h^3}{h} = \lim_{h\to 0} h^2 = 0.$$

The reason to use the $o(h)$ notation is mostly for making statements more precise. Say X is a random variable with probability density function f_X. In approximation we can say $\mathbb{P}(t < X < t+h) \approx f_x(t)\cdot h$. Instead, we say $\mathbb{P}(t < X < t+h) = f_x(t)\cdot h + o(h)$ to be more precise. ([26], page 299)

2.5 Regular Point Process

We can now define regular point processes.

Definition 10 ([17], adapted from pages 3-4): A regular point process is defined so that the probability of an event occuring in the time interval $[t, t+\Delta t)$ is given by

$$\mathbb{P}(N(t+\Delta t) - N(t) = m|\mathcal{H}_{t-}) = \begin{cases} \lambda^*(t)\Delta t + o(h) & \text{if } m = 1, \\ o(h) & \text{if } m > 1, \end{cases} \tag{2.1}$$

where $\mathcal{H}_{t-}$ is the history of the process.

We see a regular point process is a point process in which the probability of having one event in a small time-interval Δt is proportional to the length of the time-interval, and in which the probability of observing two or more events in a small time-interval is almost equal to zero. The history of the process contains information about both the number of events and the event times that have previously occured. The probabilities we are looking at are conditional probabilities: they are conditional on the history of the point process. ([17], pages 3-4)

2.6 Conditional Arrival Function

The (regular) point processes we have just defined can be characterized in a number of ways. One way is to look at the distribution function of the next arrival time conditional on the past.

Definition 11 ([21], in text, page 3): Given the history up until the last arrival u, $\mathcal{H}_{u-}$, define the conditional cumulative distribution function $F^*(t|\mathcal{H}_{u-})$ of the next arrival time T_{k+1} as

$$F^*(t|\mathcal{H}_{u-}) = \int_u^t \mathbb{P}(T_{k+1} \in [s, s+ds]|\mathcal{H}_{u-})ds = \int_u^t f^*(s|\mathcal{H}_{u-})ds,$$

where $f^*(t|\mathcal{H}_{u-})$ is the conditional probability density function of the next arrival time, also called the conditional arrival distribution.

As was the case with the conditional intensity function, also here we do not explicitely write down the history, but instead use a superscript asterisk: $F^*(t|\mathcal{H}(u)) = F^*(t)$ and $f^*(t|\mathcal{H}(u)) = f^*(t)$.

2.7 Conditional Intensity Function: Part II

In Section 2.3 we have already defined the conditional intensity function, but we have not yet made clear we this function is used. In Section 2.6 we defined the conditional arrival distribution. This distribution can be used to characterize a point process. However, the conditional arrival distribution is difficult to work with, so another characterization is preferred: the conditional intensity function. ([21], page 3)

To see why the conditional intensity function is so useful, we will look at the following Proposition, which talks about a regular point process rather than a point process.

Propostion 1 ([10], Proposition 7.2.IV): Let N be a regular point process on $\mathbb{R}_+$. Then, the conditional intensity function determines the probability structure of the point process uniquely.

So making use of the conditional intensity function, we uniquely define a (regular) point process which makes is a very useful way to characterize a process.

Besides this definition of the conditional intensity function, the conditional intensity function can also be expressed in terms of the conditional arrival function.

Definition 12 ([24], in text, page 148; [21], in text, page 3): Given the conditional cumulative distribution function $F^*(t)$ and the conditional probability density function $f^*(t)$. The conditional intensity function $\lambda^*(t)$, originally called the conditional hazard function, is defined as

$$\lambda^*(t) = \frac{f^*(t)}{1 - F^*(t)}.$$

2.8 Cumulative Conditional Intensity

Not only the conditional intensity function is used, also the integrated conditional intensity function is used. This function, which we call the cumulative conditional intensity function, will be important in the calculation of the maximum-likelihood estimator of the Hawkes process.

Definition 13 ([21], Definition 4): For a counting process $N(.)$ the non-decreasing function

$$\Lambda(t) = \int_0^t \lambda^*(s)ds$$

is called the compensator of the counting process.

Note that in [21] this function is called the compensator. However, compensator is a term that is mostly only used when talking about martingales, and since we are not talking about those in this Thesis, we will not use the term compensator.

Chapter 3

Homogeneous and Nonhomogeneous Poisson Process

Now we have defined what stochastic processes are, and what kind of tools we need to describe these processes, we are going to look at the homogeneous and nonhomogeneous Poisson process. We spent quite some time describing the (nonhomogeneous) Poisson processes, since these processes form the basis of the Hawkes processes. Without going into any detail yet (this will be done in Chapter 4), the reason we describe the (nonhomogeneous) Poisson processes so extensive, is because the Hawkes process is the non-Markovian counterpart of the (nonhomogeneous) Poisson processes.

3.1 The (homogeneous) Poisson process

The homogeneous Poisson process (from now on simply Poisson process), is one of the most import classes of stochastic processes. As the name suggests, these processes are based on the Poisson distribution.

Definition 14 ([12], Definition, page 170): A discrete random variable X has a Poisson distribution with parameter μ, with $\mu > 0$ if its probability mass function p is given by

$$p(k) = \mathbb{P}(X = k) = \frac{\mu^k}{k!}e^{-\mu}$$

for $k = 0, 1, 2, ...$ We denote this distribution by $Pois(\mu)$.

To see how the Poisson process is related to the Poisson distribution we first have to look at the definition of a Poisson process.

Definition 15 ([26], Definition 5.2): The counting process $\{N(t), t \geq 0\}$ is said to be a Poisson process with rate $\lambda > 0$ if the following axioms hold:

1. $N(0) = 0$,
2. $\{N(t), t \geq 0\}$ has independent increments,
3. $\mathbb{P}(N(t+h) - N(t) = 1) = \lambda h + o(h)$,
4. $\mathbb{P}(N(t+h) - N(t) \geq 2) = o(h)$.

Comparing Definition 15 with Definition 10 of a regular point process, we see the Poisson process is not dependent on the history, so the conditional intensity function is here just a fixed constant. This constant is called the rate of the process, and the rate of the process determines the rate at which event occurs. The greater the rate, the faster a new event occurs. A Poisson process is stationary since the conditional intensity function here does not explicitely depend on the time.

The link between the Poisson process and the Poisson distribution is made clear from the following theorem.

Theorem 1 ([26], Theorem 5.1): If $\{N(t), t \geq 0\}$ is a Poisson process with rate $\lambda > 0$, then for all $s > 0$, $t > 0$, $N(s+t) - N(s)$ is a Poisson random variable with mean λt.

Proof: See for example [26] Theorem 5.1.

Q.E.D.

So we see that in any interval of length t, the number of events is Poisson distributed with mean λt.

In terms of probability, the result of Theorem 1 can be written as

$$\mathbb{P}(N(s+t) - N(s) = n) = \frac{e^{-\lambda t}(\lambda t)^n}{n!}. \tag{3.1}$$

Defining the Poisson process in another way, gives us a Definition that results in another look at the Poisson process.

Definition 16 ([25], Definition 1): Let $\{\tau_i\}_{i \geq 1}$ be a sequence of independent and identically distributed exponential random variables with parameter λ[, thus the probability density function of a random variable τ if given by

$$f_\tau(t) = \begin{cases} \lambda e^{-\lambda t} & \text{if } t \geq 0, \\ 0 & \text{if } t < 0. \end{cases}$$

] The event times are $T_n = \sum_{i=1}^{n} \tau_i$. The process $\{N_t, t \geq 0\}$ defined by $N_t := \sum_{i \geq 1} \mathbb{I}_{t \geq T_i}$ is called a Poisson process with intensity λ.

In comparison to Definition 15, Definition 16 makes the link between the Poisson distribution and the Poisson process more clear. The Poisson process can be seen as a counting process, in which all the event times are made up of elements that are exponentionally distributed.

Definition 15 gives rise to a sequence $\{\tau_i\}_{i\geq 1}$ of random variables that are called interarrival times: the first event occurs at time τ_1, the second event occurs τ_2 after the first event, etc.

The connection between Definition 15 and Definition 16 can be made clear by calculating the expected value of τ. The expected value of τ can be calculated using integration by parts:

$$\begin{aligned} E[\tau] &= \int_{-\infty}^{\infty} t f_\tau(t) dt = \lambda \int_0^\infty t e^{-\lambda t} dt \\ &= \lambda \left[-\frac{t}{\lambda} e^{-\lambda t} \right]_{t=0}^{t=\infty} + \int_0^\infty e^{-\lambda t} dt \\ &= 0 + \left[-\frac{1}{\lambda} e^{-\lambda t} \right]_{t=0}^{t=\infty} = \frac{1}{\lambda}. \end{aligned}$$

Since $E[\tau] = \frac{1}{\lambda}$, events are arriving at an average rate of λ per time unit, which is exactly what Definition 15 told us.

The event times, which we have already seen in Section 2.1, are the times at which the events itself occur, so they can be calculated if the interarrival times are known.

The event time T_n is the time of the n-th arrival, thus $T_n = \sum_{i=1}^n \tau_i$. Since the τ_i are random variables, the sequence of event times $\{T_1, T_2, ...\}$ form a random configuration of points on the real line $[0, \infty)$. We know N_t is a counting process, so N_t increments by one for each T_i; explicitely ([25], page 4)

$$N_t = \begin{cases} 0 & \text{if } 0 \leq t < T_1, \\ 1 & \text{if } T_1 \leq t < T_2 \\ . & \\ n & \text{if } T_n \leq t < T_{n+1} \\ . & \end{cases}$$

Maybe the most important property of the Poisson process, is the memoryless property.

Memoryless Property ([25], in text, page 4): Being memoryless in a point process means that the distribution of future interarrival times depends only on relevent information about the current times, but not on information from further in the past. So a point process is memoryless if $\mathbb{P}(\tau > t + m | \tau > m) = \mathbb{P}(\tau > t)$.

To prove that the Poisson process is memoryless, we first calculate the cumulative distribution function of τ (for $t \geq 0$):

$$F_\tau(t) = \int_{-\infty}^t f_\tau(x) dx = \int_0^t \lambda e^{-\lambda x} dx = \left[-e^{-\lambda x} \right]_{x=0}^{x=t} = 1 - e^{-\lambda t}.$$

Since $F_\tau(t) = \mathbb{P}(\tau \leq t)$, we now know $\mathbb{P}(\tau > t) = 1 - \mathbb{P}(\tau \leq t) = e^{-\lambda t}$, thus

$$\begin{aligned}\mathbb{P}(\tau > t + m | \tau > m) &= \frac{\mathbb{P}(\tau > t + m, \tau > m)}{\mathbb{P}(\tau > m)} = \frac{\mathbb{P}(\tau > t + m)}{\mathbb{P}(\tau > m)} \\ &= \frac{e^{-\lambda(t+m)}}{e^{-\lambda t}} = e^{-\lambda m} = \mathbb{P}(\tau > t),\end{aligned}$$

so the Poisson process is memoryless. The fact that the Poisson process is memoryless, shows that the probability of waiting an additional time t after having already waited a time m, is equal to the probability of waiting a total time of t.

3.2 The nonhomogeneous Poisson process

In the (homogeneous) Poisson process we have discussed so far, all events arrive independently of each other at a fixed rate λ. For a Poisson process we assume that the rate always stays the same. Whereas this model is good for simulating simple processes, such as the number of customers in a store over a short period of time, this model is insufficient to model complex situations, like modelling the number of customers in a store just before diner time. For these situations the rate depends explicitely on the time, so $\lambda = \lambda(t)$. In this case $\lambda(t)$ is an intensity function. The nonhomogeneous Poisson process can now be defined.

Definition 17 ([26], Definition 5.3): The counting process $\{N(t), t \geq 0\}$ is said to be a nonhomogeneous Poisson process with intensity function $\lambda(t)$, $t \geq 0$, if the following axioms hold:

1. $N(0) = 0$,
2. $\{N(t), t \geq 0\}$ has independent increments,
3. $\mathbb{P}(N(t + h) - N(t) = 1) = \lambda(t)h + o(h)$,
4. $\mathbb{P}(N(t + h) - N(t) \geq 2) = o(h)$.

Comparing Definition 15 with Definition 17 we see that the only thing that has changed is that the fixed rate λ has been replaced by the intensity function $\lambda(t)$.

In the light of Definition 10 of a regular point process, we see the conditional intensity function here does not depend on the history. The intensity function can be interpreted in the same way as the rate: to each time it assesses a certain intensity that tells us how fast we expect the next event to occur. The difference with the Poisson process is now that the intensity is not the same for all values of t. However, since the intensity function depends on time, the nonhomogeneous Poisson pricess is nonstationary.

As was the case with the Poisson process, the link between the Poisson distribution and the nonhomogeneous Poisson process is very clear. The link is made clear in the following Theorem.

Theorem 2 ([6], Proposition 4.1): Let the point process N be a nonhomogeneous Poisson process with intensity function $\lambda(t)$, then $N(t)$ follows a Poisson distribution with parameter $\int_0^t \lambda(s)ds$, i.e.

$$\mathbb{P}(N(t) = n) = \frac{e^{-\int_0^t \lambda(s)ds} \left(\int_0^t \lambda(s)ds\right)^n}{n!}.$$

Proof: See for example [26] Proposition 4.1.

Q.E.D.

So for the nonhomogeneous Poisson process, the probability of the number of events that have happened at time t is Poisson distributed with parameter $\int_0^t \lambda(s)ds$. The parameter looks very similar to the cumulative conditional intensity defined in Definition 13, but here we have just the cumulative intensity. In the light of nonhomogeneous Poisson processes, this cumulative intensity is sometimes also called the mean value function, since the expectation of a Poisson process is equal to the value of the parameter. ([26], in text, page 322)

The result of Theorem 2 could also be written in another way.

Theorem 3 ([26], Theorem 5.3): If $\{N(t), t \geq 0\}$ is a nonstationary Poisson process with intensity function $\lambda(t)$, $t \geq 0$, then $N(t+s) - N(s)$ is a Poisson random variable with mean $\int_s^{t+s} \lambda(y)dy$, [i.e.

$$\mathbb{P}(N(t+s) - N(s) = n) = \frac{e^{-\int_s^{t+s} \lambda(y)dy} \left(\int_s^{t+s} \lambda(y)dy\right)^n}{n!}.]$$

Proof: See for example [26] Theorem 5.3.

Q.E.D.

The last Theorem tells us that if $n = 0$, the probability of having no events in $[s, t+s]$ is equal to

$$\mathbb{P}(N(t+s) - N(s) = 0) = e^{-\int_s^{t+s} \lambda(y)dy}. \tag{3.2}$$

The last formula determines the law of occurrence for the next point. For simple point processes, the points arrive one by one, thus checking whether the condition listed above is fulfilled, tells us whether the point process is a nonhomogeneous Poisson process. ([6], page 7)

Part II

Hawkes Processes

Chapter 4

The One-Dimensional Hawkes Process

In the first Part of this Thesis we have looked at the theory of stochastic processes. We have also seen some classes of stochastic processes, of which the (nonhomogeneous) Poisson processes were our main interest. We were interested in the (nonhomogeneous) Poisson processes, since they are somehow similar to the Hawkes processes, which air the main interest in this Thesis. These Hawkes processes will be defined in this Chapter. Besides defining these processes we will also look at some of their properties like stationarity and stability. Since parameter estimation is of great importance in applications, we will look at the maximum-likelihood estimator of Hawkes processes.

4.1 A Self-Exciting Process

In Chapter 3 we have seen the (homogeneous) Poisson process and the nonhomogeneous Poisson process. In the Poisson process, the events arrive independently at a constant rate. In the nonhomogeneous Poisson process, the events arrive independently due to an intensity function.

However, in reality it is not always the case that events arrive independently of each other. We know, for example, that seismic activity of the past can help predicting future earthquakes. ([23]) So in the case of earthquakes we see that the occurence of an event affects the occurence of an upcoming effect. Also in other fields of study we find that events arrive dependently of each other. In Chapter 9 we will look at retweet cascades in social media. There we will find that as soon a retweet took place the intensity function increases, which is the same as saying that the likelihood of future events increase. ([25]) It may be clear that there are processes in which events arrive dependently of each other, such processes are called self-exciting processes.

A self-exciting process is a point process in which the arrival of an event causes the conditional intensity function ot increase. ([25], in text, page 6)

For a self-exciting process, the conditional intensity function increases if a new event arrives. Since the conditional intensity function increases, the occurence of a new event is more probable. So a self-exciting process can be seen as a process in which the occurence of past points makes the occurence of future points more probable. See Figure 4.1 for an example realisation a conditional intensity function of a self-exciting process.

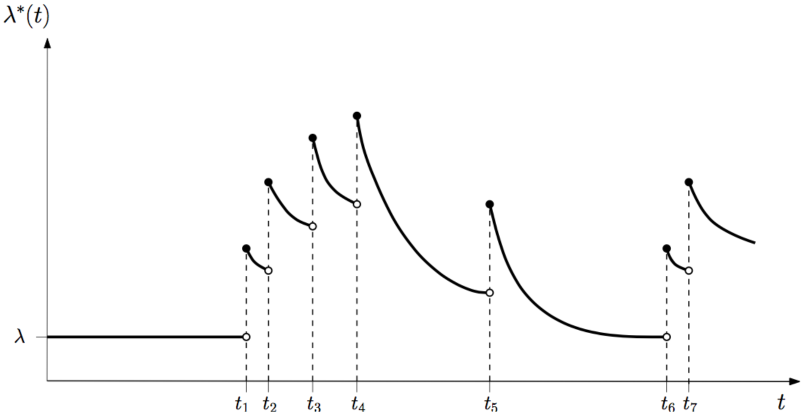


Figure 4.1: A conditional intensity function for a self-exciting process is shown. We see that as soon as an event occurs, the conditional intensity function increases. This increase, decays over time until a new event occurs. (Figure taken from [21], Figure 2)

4.2 Defining the One-Dimensional Hawkes Process

In 1971 Hawkes proposed a new self-exciting model, that turned out be very practical and is still used. The process is now referred to as Hawkes process. As is stated in [14] about the process: “one may think of this as a self-exiciting shot process in which the current intensity of events is determined by events in the past”. So the Hawkes process is a self-exciting process that depends explicitely on all previously occured events.

Definition 18 ([21], Definition 5; [25], Definition 3): Let $\{N(t) : t \geq 0\}$ be a couting process, that satisfies Equation 2.1 of Definition 10. The point process $N(.)$ is said to be a Hawkes process if the conditional intensity function $\lambda^*(t)$ is of the form

$$\lambda^*(t) = \lambda_0 + \sum_{i:t>T_i} \mu(t - T_i)$$

for some $\lambda_0 > 0$ and $\mu : (0, \infty) \to [0, \infty)$, and where T_i are the event times.

In the Hawkes process λ_0 is called the background intensity and μ is an excitation function, concepts we will define more precisely in the following subsections.

Looking at the definition of a Hawkes process, we observe that a Hawkes process is a nonhomogeneous Poisson process as long as $\mu(.) \neq 0$. If $\mu(.) = 0$, then we are having a homogeneous Poisson process, thus this trivial case will not be considered. Besides, we notice the fact that the conditional intensity function depends explicitely on previous events, which makes clear why a Hawkes process is self-exciting.

A Hawkes process is in general non-Markovian. ([30]) Very simply said, a stochastic process is a Markov process, if it has the Markovian property that, conditional on the present, the future is independent of the past. ([25], in text, page 11) It may be clear that the Hawkes process (mostly) is non-Markovian, since the past is of great importance to the future as can be observed in the conditional intensity function. The (nonhomogeneous) Poisson process is memoryless, and thus Markovian. As said in the introduction of Chapter 4, the Hawkes process can thus be seen as the non-Markovian counterpart of the (nonhomomgeneous) Poisson process.

4.2.1 The background intensity

The first term in the conditional intensity function of the self-exciting process if the background intensity term λ_0. To find out what λ_0 does, set $\mu(.) = 0$, such that $\lambda^*(t) = \lambda_0$. So if $\mu(.) = 0$ we are simply having a homogeneous Poisson process. This explanation makes clear that λ_0 is simply the rate of the process. The background intensity does not contribute to the fact that the Hawkes process is self-exciting.

In this Thesis, we will treat λ_0 as a fixed constant, but the background intensity may also be a function $\lambda_0(t)$. ([25], page 6)

As we will see in Section 4.4, a Hawkes process can be seen as a process in which there are "immigrant" and "offspring" events. The background intensity term determines the arrival of the immigrant events, and their arrival is independent of previous events occured. ([25], page 6)

In Definition 18 we have seen that λ_0 should be bigger than 0. It may be clear that the background intensity can not be smaller than zero. To see why λ_0 can not be equal to zero look at the case were there have not occured any events yet (in Figure 4.1 this is the part before t_1). In this case $\sum_{i:t>T_i} \mu(t - T_i) = 0$, such that $\lambda^*(t) = \lambda_0$. If we would have $\lambda_0 = 0$, then there will not occur any events in the future, thus the process will not really do anything. To prevent this case, we should have $\lambda_0 > 0$.

4.2.2 The excitation function

The second term in the conditional intensity function of the Hawkes process contains the excitation fucntion $\mu(.)$. The excitation function is also called the memory kernel. As is clear from $\sum_{i:t>T_i} \mu(t-T_i)$, the kernel $\mu(t-T_i)$ modulates the change than an event at time T_i has on the intensity function at time t. ([25], page 6) It is common to choose the excitation function $\mu(.)$ to be monotonically decreasing. This is common, since in this way it can be regulated that more recent events have a higher influence on the conditional intensity at this time, than events that have occured more early in time.

The excitation function can have many forms. In this Thesis we will define both the exponential and power law kernel, but we will focus on the exponential kernel.

The exponential kernel

For the excitation function $\mu(.)$ a function of exponential decay is mostly used. The form $\mu(t) = \sum_{j=1}^{k} \alpha_j e^{-\beta_j t}$, for $t > 0$, was introduced by Hawkes in 1971. ([14]) In this case the conditional intensity function becomes

$$\lambda^*(t) = \lambda_0 + \sum_{i:t>T_i} \sum_{j=1}^{k} \alpha_j e^{-\beta_j (t-T_i)},$$

so the conditional intensity function is now parameterised by the two constants α_j and β_j. As we will see in Section 4.3, stationarity will imply $\alpha_j, \beta_j > 0$ and $\sum_{j=1}^{k} \frac{\alpha_j}{\beta_j} < 1$.

In this Thesis we will mostly look at the case where $k = 1$, so in this case the conditional intensity function becomes

$$\lambda^*(t) = \lambda_0 + \sum_{i:t>T_i} \alpha e^{-\beta(t-T_i)}.$$

The last equation makes it possible for us to interpret the two coefficients. If a new event arrives, the conditional intensity function increases by α. However, this increase will decrease over time at an exponentional rate of β.

If k would not be equal to 1, then both parameters take different values for all events. In Section 4.5, we are going to look at a marked Hawkes process with exponential kernel, which is basically the Hawkes process with exponentional kernel, with β fixed and α_j event dependent. The mark should be seen as the increase in the conditional intensity function due to the occurence of the event.

The power law kernel

Besides a function of exponential decay, also a power law function $\mu(t) = \frac{K}{(c+t)^p}$ is often used. In this case the conditional intensity function becomes

$$\lambda^*(t) = \lambda_0 + \sum_{i:t>T_i} \frac{K}{(c + (t - T_i))^p},$$

so the conditional intensity function is now parameterised by the three constants k, c and p. In Chapter 9 we will be looking at an application in which this power law kernel is used. The intepretation of the parameters will there be explained in light of the application.

4.2.3 Visual Representation of the One-Dimensional Hawkes Process

To finish this section about what a Hawkes process precisly is, we will give a visual representation of such a process. See Figure 4.2.

4.3 Stationarity Conditions

As we have seen a stochastic process is a set of random variables, all coming from the same distribution, that are indexed to a certain set. Mostly we consider time index sets, such that we get a list of random variables that somehow vary over the time. The outcoming set of this stochastic process, can be seen as a time series. A complete area of econometrics is focussed on studying and modelling these time series. When running a regression on the time series, often stationarity is assumed.

Definition 19 ([26], in text, page 633): A stochastic process $\{X(t), t \geq 0\}$ is said to be a stationary process if for all $n, s, t_1, ..., t_n$ the random vectors $X(t_1), ..., X(t_n)$ and $X(t_1 + s), ..., X(t_n + s)$ have the same joint distribution.

So a stochastic process is stationary if the sample mean, the variance and the autocorrelation are constant over time. ([13], in text, page 77) Very heuristically said, the autocorrelation is the correlation between two elements of a time series.

We want time series to be stationary, since a violation of this assumption makes the outcomes of the regression we have runned not interpretable in the right way. ([29], page 395)

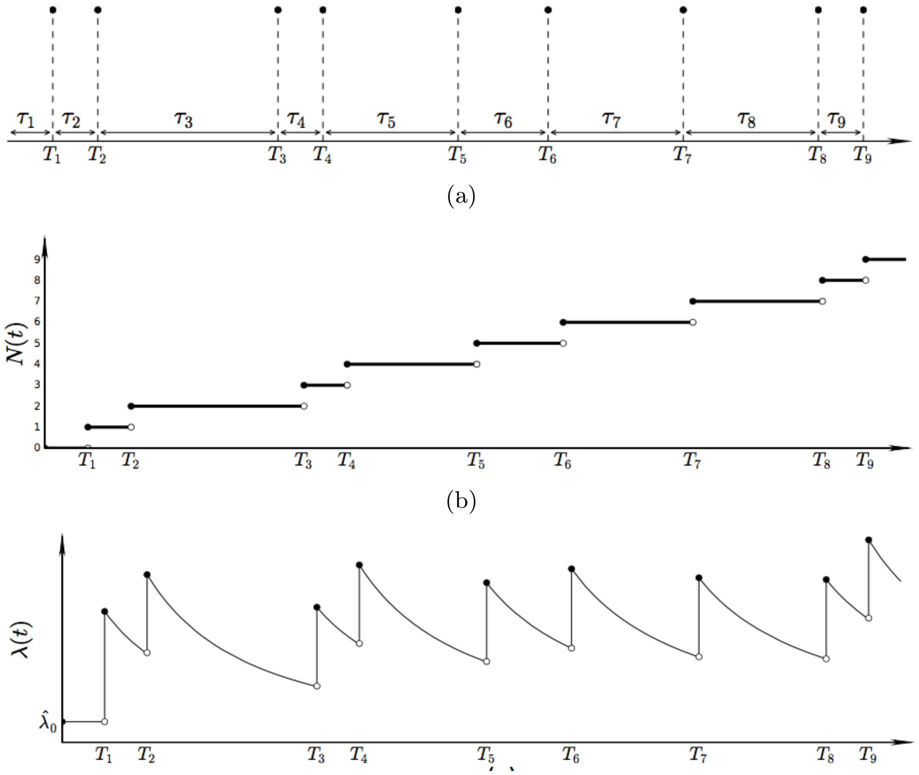
Readers who are interested in the analysis of time series are encouraged to read [13] (Chapter 4 focusses on stationarity) and [29] (Chapter 12 is about time series models).

However, for practical purposes, the stationary condition is often too strict. Often the condition of weak stationarity is enough to work with.

Definition 20 ([26], in text, page 634): A stochastic process $\{X(t), t \geq 0\}$ is said to be a second-order stationary or a weakly stationary process if $E[X(t)] = c$ for some constant c and $Cov(X(t), X(t + s))$ does not depend on t.

So put differently, a process is weakly stationary if the first two moments of $X(t)$ are the same for all t and the covariance between $X(s)$ and $X(t)$ depends only on $|t - s|$. ([26], in text, page 634)

We want to see under which condition the Hawkes process is weakly stationary. We are only going to look at the constant mean part, since this is the most important property of weak stationarity.



(a)

(b)

(c)

Figure 4.2: Figure 4.2a shows an example realization of a Hawkes process: nine events are observed, at times T_1, T_2, ..., T_9, and their corresponding inter-arrival times τ_1, τ_2, ..., τ_9. Figure 4.2b shows the corresponding counting process N_t over time, which increases by one unit for each T_i. Figure 4.2c shows the intensity function $\lambda^*(t)$ over time. Visibly, the value of the intensity function increases immediately after the occurence of an event T_i, and diminishes as time passes and the effect of the given event T_i decays. (Figure taken from [25], Figure 1.2a-1.2c)

Let us determine the expected value of the conditional intensity function of Hawkes process:

$$E[\lambda^*(t)] = E\left[\lambda_0 + \sum_{i:t>T_i} \mu(t - T_i)\right] = \lambda_0 + E\left[\sum_{i:t>T_i} \mu(t - T_i)\right].$$

In [14] (in text, Equation 8) the conditional intensity function of Hawkes process is defined with a stochastic integral instead of a sum of all the event times occured before the time we are looking at. Since talking about stochastic integrals is outside the scope of this Thesis, in Definition 18 we have only given the definition of a Hawkes process with a sum. However, using the stochastic integral definition, it is shown in [14] (page 84) that $E\left[\sum_{i:t>T_i} \mu(t - T_i)\right] = \int_{-\infty}^{t} \mu(t-u)E[\lambda^*(t)]du$. The expected value of the conditional intensity function now becomes

$$E[\lambda^*(t)] = \lambda_0 + \int_{-\infty}^{t} \mu(t - u)E[\lambda^*(t)]du.$$

Assuming stationarity, we can take $E[\lambda^*(t)]$ out of the integral and rewrite the equation above to

$$E[\lambda^*(t)] = \frac{\lambda_0}{1 - \int_{-\infty}^{t} \mu(t - u)du} = \frac{\lambda_0}{1 - \int_{0}^{\infty} \mu(u)du}. \tag{4.1}$$

Now Equation 4.1 again shows us that λ_0 should be greater than zero, since a negative expected value for a positive function does not make sense. Besides, Equation 4.1 tells us that in order to have stationarity, it should be the case that

$$\int_{0}^{\infty} \mu(u)du < 1.$$

In the case of the exponentional kernel $\mu(u) = \alpha e^{-\beta u}$, we observe that stationarity is fulfilled if

$$\int_{0}^{\infty} \mu(u)du = \int_{0}^{\infty} \alpha e^{-\beta u}du = \alpha\left[\frac{-e^{-\beta u}}{\beta}\right]_{0}^{\infty} = \frac{\alpha}{\beta} < 1.$$

In Section 4.4.3 we will have a more complete look at the stationarity of a Hawkes process.

4.4 Branching Structure

The Hawkes process can also be viewed as a Poisson cluster process representation. ([15]) In this view, the events in the Hawkes process are seen as either “immigrants” or “offspring”. As the names suggests, an immigrant event is an event that arrives independently of other events, while an offspring event is triggered by a previous event in the process. In a cluster all the offspring associated with a certain immigrant are structured. This view of a Hawkes process is called the branching structure. ([25], page 8)

In the scope of this Thesis there are three justifications for explaining this alternative view. Firstly, this new view makes it more easy to prove certain properties of the Hawkes process: we will again talk about the stationarity of the process, but then based on this branching structure. Secondly, based on the branching structure an Algorithm for simulating a Hawkes process can be set up, as we will see in Section 7.3. Thirdly, the branching structure can make the modelling of certain real life situations possible, as we will see in Chapter 9.

4.4.1 An Example Branching Structure

Imagine that we our counting the population in a country. The country consists of immigrants and their offspring. The immigrant events follow a homogeneous Poisson process with rate λ_0. The offspring events are then a result of the immigrant events: an immigrant produces zero or more children all independent of the other immigrants. So the offspring events follow a self-exciting process.

In Figure 4.3 the branching structure of the example Hawkes process in Figure 4.2a is shown. However, it is good to note that based on the points in a point process, the branching strucure can not be observed: an event alone does not tell us if it is an immigrant or an offspring event. All the events are denoted by circles. The circle where the arrow points at is the offspring of the circle from where the arrow starts. The random variables Z_{ij} are introduced, where $Z_{i0} = 1$ if event i is an immigrant and $Z_{ij} = 1$ if event j is an offspring of i. The generation of the event, which is denoted in the circle, denotes the generation to which an event belongs. The immigrants are labeled as Gen_0, where the offspring are denoted by Gen_k, with $k > 0$. ([25], pages 8-9)

In the cluster representation all the offspring associated with a certain immigrant are structured. In this cluster representation, the events that are directly or indirectly connected to a specific immigrant form the cluster of offspring with that immigrant. For example T_2, T_3, T_4, T_5 and T_6 form the cluster of offspring of T_1. The offspring events that are associated with a certain immigrant event, are arriving according to a nonhomogeneous Poisson process with an intensity function $\lambda(.)$. Looking at Figure 4.3, we observe that the third event, with corresponding event time T_3, is coming from a nonhomogeneous Poisson process endowed with intensity $\lambda(t - T_2)$ for $t > T_2$. ([25], page 9)

4.4.2 The Branching Factor

A quantity that can be used to describe the Hawkes process is the branching factor/ratio. The branching factor is defined as the expected number of direct offspring spawned by a single event. ([25], in text, page 9)

The branching factor n^* can be computed by integrating the conditional intensity function over all time values ([21], page 7):

$$n^* = \int_0^\infty \mu(s)ds. \tag{4.2}$$

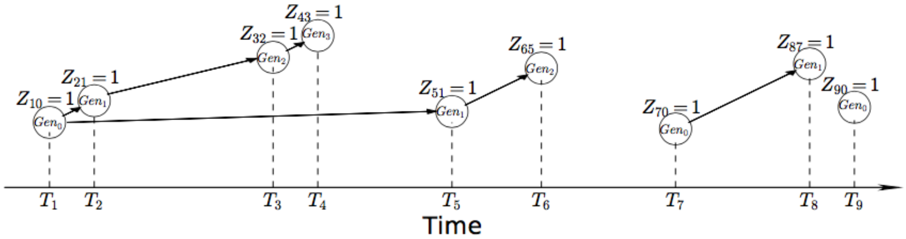


Figure 4.3: An example branching strucute of the Hawkes process in Figure 4.2a is shown. All circles represent an event that has occured at time T_i. The arrows between the circles tell us the relation between the different events, so connected circles are in the same cluster. The generation of a certain event is denoted by Gen_i, with $i = 0$ for immigrant events and $i > 0$ for the offspring. The random variables Z_{ij} tell us which events are related: $Z_{i0} = 1$ if event i is an immigrant and $Z_{ij} = 1$ if event j is an offspring of i. (Figure taken from [25], Figure 1.2d)

So the branching factor is the value of the cumulative conditional intensity (Definition 13) at infinity.

In the case of an exponentional kernel, $\mu(s) = \alpha e^{-\beta s}$, the branching factor becomes

$$n^* = \int_0^\infty \alpha e^{-\beta s} ds = \alpha \left[-\frac{e^{-\beta s}}{\beta} \right]_{s=0}^{s=\infty} = \frac{\alpha}{\beta}.$$

4.4.3 Stationarity Revised

In Section 4.3 we have defined stationarity and we have seen that the Hawkes process is weakly stationary if $\frac{\alpha}{\beta} < 1$. Since $n^* = \frac{\alpha}{\beta}$, we can express the stationarity requirement as $n^* < 1$.

However, depending on the values of α and β, n^* can also take values bigger than or equal to 1 (in Section 4.2.2 we have seen that $\alpha, \beta > 0$, so n^* can not become negative or zero).

Before discussing the stationarity, we are first going to calculate the expected number of offspring for one immigrant. Let us first define A_i as the expected number of events in Gen_i. Based on how the branching factor is defined, we notice that each event in the previous generation has on average n^* offspring events, thus $A_i = n^* A_{i-1} = ... = (n^*)^i$, for $i \geq 1$. ([25], Equation 1.16) The expected number of offspring for one individual now becomes

$$E\left[\sum_{i=1}^{\infty} A_i\right] = \sum_{i=1}^{\infty} E[A_i] = \sum_{i=1}^{\infty} (n^*)^i = \begin{cases} \frac{n^*}{1-n^*} & \text{if } n^* < 1, \\ \infty & \text{if } n^* \geq 1. \end{cases}$$

If $n^* \in (0,1)$ we have that

$$\frac{E[\sum_{i=1}^{\infty} A_i]}{1 + E[\sum_{i=1}^{\infty} A_i]} = \frac{\frac{n^*}{1-n^*}}{1 + \frac{n^*}{1-n^*}} = n^*,$$

so the branching ratio can be seen as the ratio between the number of events in a cluster to the total number of the entire family (the cluster plus the immigrant). ([21], page 8)

Let us now get back to looking at the stationarity. Corresponding to the three possible "states" of n^*, the process can be in three different phases ([2]: both the phases, and there explanation are taken from here):

1. If $n^* < 1$ we are in the sub-critical phase. As seen before, $n^* < 1$ corresponds to a stationary process. In this sub-critial phase each immigrant event generates on average less than one offspring event. In this phase the total progeny of each event is almost surely (see Appendix B2) finite, as is the average number of generations before extinction. This is also something that can be seen at looking at Equation 4.3: if $n^* < 1$ the expected number of events in a cluster is bounded.

2. If $n^* = 1$ we are in the critical phase. In this phase, the process is quasi-stationary. The fact that the process is quasi-stationary means that although the average conditional intensity of the process grows exponentionally (for the exponential kernel) in time, the process may still possess a finite expected number of events. ([2]). In this phase the total progeny is almost surely finite, but there are large fluctuations in the size of the progeny. These large fluctuations result in a diverging average number of generations before extinction.

3. If $n^* > 1$ we are in the super-critical phase. In this phase, the process is nonstationary. In the super-critical phase each immigrant event generates on average more than one offspring event. This results in a progeny of an immigrant event that might be infinite. This is also something that can be seen at looking at Equation 4.3: if $n^* > 1$ the expected number of events in a cluster is unbounded.

4.5 Expected Number of Events

In Chapter 3, we have seen the distribution of the number of events for both the homogeneous and nonhomogeneous Poisson process at a certain point in time.

For Hawkes processes we will not look at a certain distribution for the number of events occured, but we will look at the expected number of events at a certain point in time. We will only look at the case with the exponential kernel.

However, we slightly change our definition of the exponential kernel. In Section 4.2.2 we have already mentioned the marked Hawkes process. The marked Hawkes process is simply a Hawkes process with exponential kernel in which α is not a constant, but is different for each event: $\mu(t) = \sum_{j=1}^{k} \alpha_j e^{-\beta t}$. So the increase in the conditional intensity is not the same for every event occured. In this way, we can make the occurence of one event more important than the occurence of another event. The value of each mark is nonnegative and its value is independent of all events previously occured. ([26], page 335)

Theorem 4 now gives us an expression for the expected number of events.

Theorem 4 ([26], Proposition 5.5): If α is the expected value of a mark in a Hawkes process, then for this process

$$E[N(t)] = \lambda_0 t + \frac{\lambda_0 \alpha}{(\alpha - \beta)^2} \left(e^{(\alpha-\beta)t} - 1 - (\alpha - \beta)t \right). \tag{4.3}$$

Proof: See for example [26] Proposition 5.5.

Q.E.D.

In Theorem 4 we have said the expected value of the mark to be equal to α: this α corresponds to the α in Definition 19.

We can now look more carefully at Equation 4.3 by looking at all the variables separately. The code for making the graphs in the following subsections can be found back in Appendix C.

4.5.1 α as a Variable

In Figure 4.4 the expected number of events has been drawn as a function of α. We observe that the number of events increases slowly, but explodes rapidly after a certain value. This value is the value after which the process is not stationary anymore. We know the process is stationary if $\alpha/\beta < 1$. So if α is smaller than β then the process is stationary and the expected number of events only increases slowly. However, when α is bigger than β than the process is not stationary anymore and the expected number of events rapidly increases.

4.5.2 β as a Variable

In Figure 4.5 the expected number of events has been drawn as a function of β. We observe exactly the converse of what we have seen in Figure 4.4. For values of β smaller than α the process is not stationary, thus the expected number of events, as β gets smaller, increases rapidly. If β is bigger than α, than the process is stationary, thus the expected number of events does not change much anymore.

4.5.3 λ_0 and t as a Variable

In Figure 4.6 we have drawn the expected number of events: in the graph on the left as a function of λ_0 and in the graph on the right as a function of t. As was to be expected from Equation 4.3, the expected number of events is a linearly dependent on λ_0. The expected number of events is also almost linearly dependent on t; there is just a very light curve.

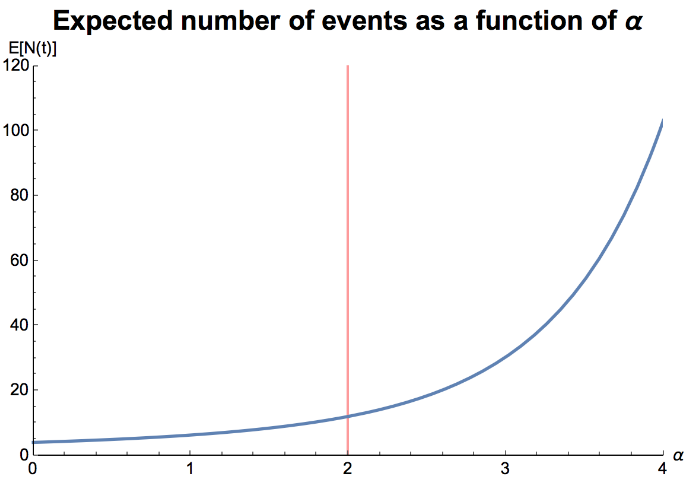


Figure 4.4: The expected number of events has been drawn as a function of α. The other variables are all equal to 2, thus $\beta = \lambda_0 = t = 2$. The red line denotes the value after which the Hawkes process is not stationary anymore.

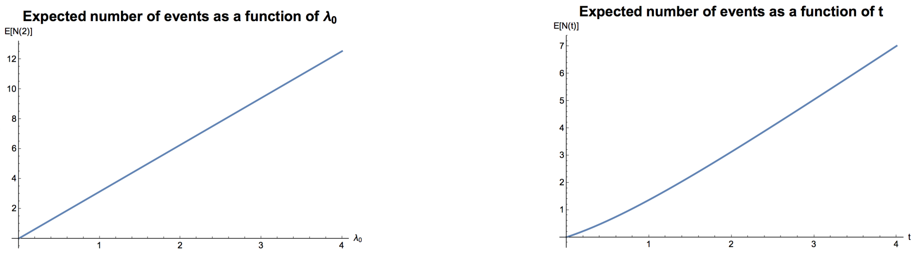


Figure 4.6: In the graphs the expected number of events has been drawn: left as a function of λ_0 and right as a function of t. We have set $\alpha = 1$ and $\beta = 2$ to assure a stationary process. The other variable has been set equal to 2.

4.6 Maximum-Likelihood Estimation

Now that we have seen what a Hawkes process is, we would like to apply the Hawkes process to real life data. The difficult step in finding a good Hawkes process for a certain data set is the estimation of the parameters in the model. One way to find values for these parameters is by finding the maximum-likelihood estimators.

4.6.1 Likelihood Function of a Hawkes Process

In order to find the maximum-likelihood estimators for our data, we first have to find the likelihood function that is associated by the process. In the following,

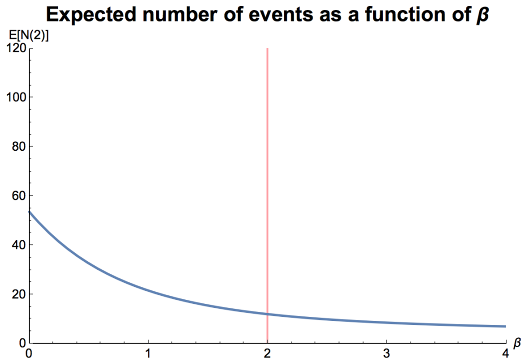


Figure 4.5: The expected number of events has been drawn as a function of β. The other variables are all equal to 2, thus $\alpha = \lambda_0 = t = 2$. The red line denotes the value after which the Hawkes process is not stationary anymore.

let θ be the set of parameters of the Hawkes process for which values want to be found.

Theorem 5 ([10], Proposition 7.2.III.): Let N be a regular point process on $[0, T]$ for some finite positive T and let $\{T_1, T_2, ..., T_n\}$ [be the corresponding list of event times]. Then, the likelihood $L(\theta)$ of such N is expressible in the form

$$L(\theta) = \left[\prod_{i=1}^{n} \lambda^*(T_i)\right] \cdot \exp\left(-\int_0^T \lambda^*(u)du\right). \tag{4.4}$$

Let us prove Theorem 5. (This proof follows [21], Theorem 3, and [25], in text, page 13).

Proof We start with looking at the realization $\{T_1, T_2, ..., T_n\}$ of the point process. The conditional arrival distribution at event time T_i is $f^*(T_i)$. Since all event times are independent, the joint probability function, and thus the likelihood function, becomes

$$L(\theta) = f^*(T_1, T_2, ..., T_n) = \prod_{i=1}^{n} f^*(T_i).$$

Using Definition 12 we can write

$$\lambda^*(t) = \frac{f^*(t)}{1 - F^*(t)} = \frac{\frac{\mathrm{d}}{\mathrm{d}t}[F^*(t)]}{1 - F^*(t)} = -\frac{\mathrm{d}}{\mathrm{d}t}[\ln(1 - F^*(t))].$$

Integrating both sides from T_n to t we get

$$\int_{T_n}^{t} \lambda^*(t)dt = \int_{T_n}^{t} \left(-\frac{\mathrm{d}}{\mathrm{d}t}[\ln(1 - F^*(t))]\right) dt = \ln(1 - F^*(T_n)) - \ln(1 - F^*(t)).$$

Since $T_{n+1} > T_n$, we have $F^*(T_n) = 0$, so

$$\int_{T_n}^{t} \lambda^*(t)dt = -\ln(1 - F^*(t)),$$

and thus

$$1 - F^*(t) = \exp\left(-\int_{T_n}^{t} \lambda^*(t)dt\right).$$

Rewritting Definition 12 and plugging in the result found, we get

$$f^*(t) = \lambda^*(t) \cdot \exp\left(-\int_{T_n}^{t} \lambda^*(t)dt\right).$$

Plugging the last result into the likelihood function we find

$$\begin{aligned} L(\theta) = \prod_{i=1}^{n} f^*(T_i) &= \prod_{i=1}^{n} \lambda^*(T_i) \cdot \exp\left(-\int_{T_n}^{T_i} \lambda^*(T_i)\right) \\ &= \left[\prod_{i=1}^{n} \lambda^*(T_i)\right] \cdot \exp\left(-\int_{0}^{T} \lambda^*(u)du\right). \end{aligned}$$

Q.E.D.

4.6.2 Log-Likelihood Function of a Hawkes Process with Exponential Kernel

Using Theorem 4 and the Definition of the cumulative conditional intensity (Definition 13), we see that the log-likelihood function of a Hawkes process over the interval $[0, T_n]$ is

$$\begin{aligned} l(\theta) = \ln(L(\theta)) &= \ln\left(\left[\prod_{i=1}^{n} \lambda^*(T_i)\right] \cdot \exp\left(-\int_{0}^{T_n} \lambda^*(u)du\right)\right) \\ &= \sum_{i=1}^{n} \ln(\lambda^*(T_i)) - \int_{0}^{T_n} \lambda^*(u)du = \sum_{i=1}^{n} \ln(\lambda^*(T_i)) - \Lambda(T_n). \end{aligned}$$

Let's look at $\Lambda(T_n)$. Using the definition of cumulative conditional intensity we can write

$$\begin{aligned} \Lambda(T_n) &= \int_{0}^{T_n} \lambda^*(u)du \\ &= \int_{0}^{T_1} \lambda^*(u)du + \int_{T_1}^{T_2} \lambda^*(u)du + ... + \int_{T_{n-1}}^{T_n} \lambda^*(u)du \\ &= \int_{0}^{T_1} \lambda^*(u)du + \sum_{i=1}^{n-1} \int_{T_i}^{T_{i+1}} \lambda^*(u)du. \end{aligned}$$

In the case of an exponential kernel with $k = 1$, so $\lambda^*(t) = \lambda_0 + \sum_{i:t>T_i} \alpha e^{-\beta(t-T_i)}$, we get

$$\begin{aligned}
\Lambda(T_n) &= \int_0^{T_1} \left(\lambda_0 + \sum_{j:u>T_j} \alpha e^{-\beta(u-T_j)} \right) du + \sum_{i=1}^{n-1} \int_{T_i}^{T_{i+1}} \left(\lambda_0 + \sum_{j:u>T_j} \alpha e^{-\beta(u-T_j)} \right) du \\
&= \int_0^{T_1} \lambda_0 du + \int_{T_1}^{T_2} \lambda_0 du + ... + \int_{T_{n-1}}^{T_n} \lambda_0 du + \int_0^{T_1} \left(\sum_{j:u>T_j} \alpha e^{-\beta(u-T_j)} \right) du \\
&+ \int_{T_1}^{T_2} \left(\sum_{j:u>T_j} \alpha e^{-\beta(u-T_j)} \right) du + ... + \int_{T_{n-1}}^{T_n} \left(\sum_{j:u>T_j} \alpha e^{-\beta(u-T_j)} \right) du \\
&= \int_0^{T_n} \lambda_0 du + \alpha \sum_{i=1}^{n-1} \int_{T_i}^{T_{i+1}} \left(\sum_{j=1}^{i} e^{-\beta(u-T_j)} \right) du \\
&= \lambda_0 T_n + \alpha \sum_{i=1}^{n-1} \sum_{j=1}^{i} \int_{T_i}^{T_{i+1}} e^{-\beta(u-T_j)} du \\
&= \lambda_0 T_n + \alpha \sum_{i=1}^{n-1} \sum_{j=1}^{i} \left[-\frac{1}{\beta} e^{-\beta(u-T_j)} \right]_{u=T_i}^{u=T_{i+1}} \\
&= \lambda_0 T_n - \frac{\alpha}{\beta} \sum_{i=1}^{n-1} \sum_{j=1}^{i} \left[e^{-\beta(T_{i+1}-T_j)} - e^{-\beta(T_i-T_j)} \right].
\end{aligned}$$

In [21] (page 16) it is said $\sum_{i=1}^{n-1} \sum_{j=1}^{i} \left[e^{-\beta(T_{i+1}-T_j)} - e^{-\beta(T_i-T_j)} \right] = \sum_{i=1}^{n-1} \left[e^{-\beta(T_n-T_i) - e^{-\beta(T_i-T_i)}} \right]$, thus we finally find

$$\begin{aligned}
\Lambda(T_n) &= \lambda_0 T_n - \frac{\alpha}{\beta} \sum_{i=1}^{n-1} \left[e^{-\beta(T_n-T_i) - e^{-\beta(T_i-T_i)}} \right] \\
&= \lambda_0 T_n - \frac{\alpha}{\beta} \sum_{i=1}^{n} \left[e^{-\beta(T_n-T_i)-1} \right].
\end{aligned}$$

So the log-likelihood function now becomes

$$\begin{aligned}
l(\theta) &= \sum_{i=1}^{n} \ln(\lambda^*(T_i)) - \Lambda(T_n) \\
&= \sum_{i=1}^{n} \ln(\lambda_0 + \alpha \sum_{j:T_i>T_j} e^{-\beta(T_i-T_j)}) - \lambda_0 T_n + \frac{\alpha}{\beta} \sum_{i=1}^{n} \left[e^{-\beta(T_n-T_i)-1} \right] \\
&= \sum_{i=1}^{n} \ln(\lambda_0 + \alpha \sum_{j=1}^{i-1} e^{-\beta(T_i-T_j)}) - \lambda_0 T_n + \frac{\alpha}{\beta} \sum_{i=1}^{n} \left[e^{-\beta(T_n-T_i)-1} \right].
\end{aligned}$$

We now get a log-likelihood function which contains a double summation. Computationally wise, this double sum is very unattractive, since it results in $\mathcal{O}(n^2)$ complexity. However, it is possible to lower the complexity to $\mathcal{O}(n)$. ([21], page 17) To do this, we define a recursive formula, which has been first suggested in [23]:

$$B(i) = \sum_{j=1}^{i-1} e^{-\beta(T_i - T_j)}.$$

The recursive formula can be rewritten as

$$\begin{aligned} B(i) = \sum_{j=1}^{i-1} e^{-\beta(T_i - T_j)} &= \sum_{j=1}^{i-1} e^{-\beta(T_i - T_{i-1} + T_{i-1} - T_j)} \\ &= e^{-\beta(T_i - T_{i-1})} \sum_{j=1}^{i-1} e^{-\beta(T_{i-1} - T_j)} \\ &= e^{-\beta(T_i - T_{i-1})} \left(\sum_{j=1}^{i-2} e^{-\beta(T_{i-1} - T_j)} + e^{-\beta(T_{i-1} - T_{i-1})} \right) \\ &= e^{-\beta(T_i - T_{i-1})} \left(B(i-1) + 1 \right). \end{aligned}$$

In terms of the recursive formula, the log-likelihood function of a Hawkes process with exponential kernel now becomes

$$l(\theta) = \sum_{i=1}^{n} \ln(\lambda_0 + \alpha B(i)) - \lambda_0 T_n + \frac{\alpha}{\beta} \sum_{i=1}^{n} \left[e^{-\beta(T_n - T_i) - 1} \right],$$

where $\theta = \{\lambda_0, \alpha, \beta\}$. The computation of the maximum-likelihood estimator now entails a complexity of $\mathcal{O}(n)$. ([21], page 17)

Chapter 5

The Multidimensional Hawkes Process

In Chapter 4 we have looked in depth to the one-dimensional Hawkes process. The Hawkes process can also be defined in a multidimensional way. We will not discuss the multidimensional Hawkes processes as extensively as the one-dimensional Hawkes process, since the theory, computation and simulation is way more complex for the multidimensional case, and this is beyond the scope of this Thesis. However, to give a complete overview of the Hawkes processes we will give the definition and the maximum-likelihood estimator of the multi-dimensional Hawkes process in this Chapter, but all calculations and details are omitted. Besides, we define multivariate point processes and natural fitrations, since these concepts are used to define the multidimensional Hawkes process. We will only consider the case with the exponential kernel. Readers who are interested in these processes are encouraged to look at [6] or [31].

5.1 Basics of Multivariate Point Processes

In Section 4.2.2 we have already looked at the mark of a Hawkes process, which we used in Section 4.5 to determine the expected number of events for a Hawkes process. The mark was defined as the immediate increase in the conditional intensity function due to the occurence of a new event.

We can define the mark also in a more general way. Say we are having a point process consisting of the points $\{T_1, T_2, ...\}$. Besides these points T_i, we now also observed another bit of information for each event i, denoted by m_i. We call this additional bit of information a mark. So we are now observing the process $\{(T_1, m_1), (T_2, m_2), ...\}$. Such a process is called a marked point process. ([27], in text, page 9)

This marked point process, can be seen as an example of a multivariate point process.

Definition 21 ([3], Definition 2): Let $\{T_i\}_{i\in\{1,2,...\}}$ be a simple point process on $[0,\infty)$, defined on $(\Omega,\mathcal{F},\mathbb{P})$, and let $\{Z_i\}_{i\in\{1,2,...\}}$ be a sequence of $\{1,2,...,M\}$- values random variables (also defined on $(\Omega,\mathcal{F},\mathbb{P})$, with $1 \leq M \leq \infty$). Then the double sequence $\{T_i,Z_i\}_{i\in\{1,2,...\}}$ is called a M-variate point process on $[0,\infty)$.

Associated with this M-variate, or more generally multivariate, point process, we can define the M-variate counting process.

Definition 22 ([3], Definition 2): Let $\{T_i,Z_i\}_{i\in\{1,2,...\}}$ be a M-variate point process on $[0,\infty)$. Define for all m, $1 \leq m \leq M$, and all $t \geq 0$

$$N_m(t) = \sum_{i\geq 1} \mathbb{I}_{T_i\leq t}\mathbb{I}_{Z_i=m}.$$

Then the M-vector process $\boldsymbol{N}(t) = (N_1(t),...,N_M(t))$ is the M-variate counting process associated with $\{T_i,Z_i\}$.

5.2 Natural Filtrations

In Section 1.2 we have defined a filtration (Definition 4). Let $\{\mathcal{H}_{t-}\}_{t\geq 0}$ be a filtration. To recall: with t we denote the time, and the σ-algebra $\mathcal{H}_{t-}$ consists of all the events that have occured up to, but not including, time t. The filtration $\{\mathcal{H}_{t-}\}$ can thus be seen as the evolution of the information of the process over time.

Definition 22 ([7], Definition 2): A stochastic process $\{X_t\}_{t\geq 0}$ on $(\Omega,\mathcal{H},\mathbb{P})$ is adapted to the filtration $\{\mathcal{H}_{t-}\}_{t\geq 0}$ if, for each $t \geq 0$, X_t is $\{\mathcal{H}_{t-}\}$-measurable.

Based on Definition 22, we see that if $\{X_t\}_{t\geq 0}$ is an adapted process, then the value of X at time t, denoted by X_t only depends on the evolution of the process up to, but not including time t. ([7], page 1)

Definition 23 ([7], in text, page 1): Let $\{X_t\}_{t\geq 0}$ be a stochastic process. Denote the natural filtration by $\{\mathcal{H}^X_{t-}\}_{t\geq 0}$. The natural filtration is the smallest σ-algebra with respect to which all the variables X_s $(s \leq t)$ are measurable.

Based on Definition 23 we can see that $\{\mathcal{H}^X_{t-}\}_{t\geq 0}$ is the smallest filtration to which X is adapted. A stochastic process $\{X_t\}_{t\geq 0}$ is always adapted to its natural filtration. ([7], in text, page 1)

5.3 Defining the Multidimensional Hawkes Process

As we have seen in Section 4.2, a Hawkes process is self-exciting in the sense that events in the past increase the occurence of an event in the future. Multidimensional Hawkes processes are also self-exciting in this way, but besides, they also have cross-exciting between different dimensions. ([5])

Definition 24 ([5], Definition 1.1): Let $\boldsymbol{N}(t) = (N_1(t), N_2(t), ..., N_M(t))$ be a simple multivariate point process, with associated history $\{\mathcal{H}^X_{t-}\}_{t\geq 0}$, that satisfies

$$\mathbb{P}(N^m(t+h) - N^m(t) = k|\mathcal{H}^X_{t-}) = \begin{cases} \lambda^m(t)h + o(h) & \text{if } k = 1, \\ o(h) & \text{if } k > 1, \\ 1 - \lambda^m(t)h + o(h) & \text{if } k = 0. \end{cases}$$

The point process $\boldsymbol{N}(.)$ is said to be a M-variate Hawkes process [with exponential kernel] if the conditional intensity function $\lambda^{m*}(t)$ is of the form

$$\lambda^{m*}(t) = \lambda^m_0 + \sum_{n=1}^{M} \sum_{k:t>T^n_k} \alpha_{mn} e^{-\beta_{mn}(t-T^n_k)}, \tag{5.1}$$

where $\lambda^m_0 > 0$, $\alpha_{mn} > 0$ and $\beta_{mn} > 0$ for $m, n = 1, 2, ..., M$.

5.4 Stationarity Conditions

As we did for the one-dimensional Hawkes process with exponential kernel, we will determine the condition for which the multivariate Hawkes process with exponential kernel is stationary.

The thing is in [5] and [31] the multidimensional Hawkes process with exponential kernel is defined using stochastic integrals. As said in Section 4.3 we will not explain these integrals in this Thesis. Instead, we just give the condition under which the multidimensional Hawkes process with exponential kernel is stationary.

The condition for the multidimensional Hawkes process with exponential to be stationary is that the spectral radius of the matrix

$$\boldsymbol{\Gamma} = \int_0^\infty \boldsymbol{G}(u)du = \left(\frac{\alpha_{mn}}{\beta_{mn}}\right)_{m,n=1,\dots,M},$$

where

$$\boldsymbol{G}(u) = \left(\alpha_{mn}e^{-\beta_{mn}u}\right)_{m,n=1,2,\dots,M},$$

is strictly smaller than 1. ([5], Remark 1.1)

The spectral radius of the matrix $\boldsymbol{\Gamma}$ is defined as

$$\rho(\boldsymbol{\Gamma}) = \max_{a\in\mathcal{S}(\boldsymbol{\Gamma})} |a|,$$

where $\mathcal{S}(\mathbf{\Gamma})$ denotes the set of all eigenvalues of $\mathbf{\Gamma}$. ([31])

5.5 Maximum-Likelihood Estimation

We just give the final result from [31]. The final expression of the log-likelihood for a multidimensional Hawkes process is

$$\begin{aligned}\ln\left(l^m(\{T_i\})\right) = T &- \sum_{i=1}^{N}\sum_{n=1}^{M}\frac{\alpha_{mn}}{\beta_{mn}}\left(1-e^{-\beta_{mn}(T-T_i)}\right)\\ &+ \sum_{T_l^m}\ln\left[\lambda_0^m(T_l^m)+\sum_{n=1}^{M}\alpha_{mn}R_{mn}(l)\right],\end{aligned}$$

where $R_{mn}(l) = \sum_{T_k^n < T_l^m} e^{-\beta_{mn}(T_l^m - T_k^n)}$ and $R_{mn}(0) = 0$.

Part III

Simulating (Nonhomogeous) Poisson and Hawkes Processes

Chapter 6

Simulation of (Nonhomogeneous) Poisson Processes

In Chapter 3 we have looked at the (nonhomogeneous) Poisson processes. These processes formed the basis for defining the Hawkes processes we investigate in this Thesis. The theoretical properties we have examined in Section 3.3 can be visualised by making use of a simulation study. Besides, simulation studies can help us to give more insight in how a process works. So in this Chapter we will take a look at the simulation of both homogeneous and nonhomogeneous Poisson processes. By running simulations in Matlab, some of the findings of Section 3.3 will be checked.

6.1 The Homogeneous Poisson Process

6.1.1 Recap of Poisson Processes

The first process we will be simulating is the (homogeneous) Poisson process. Suppose we want to simulate a Poisson process with rate λ over the interval $[0, T]$. The simulation should then give us a list of event times $\{T_1, ..., T_n\}$ of all the n events that took place between 0 and T.

Instead of simulating the event times $\{T_1, ..., T_n\}$, we will simulate the interarrival times $\{\tau_1, ..., \tau_n\}$, since the distribution of the interarrival times is known. The n-th event time of the process is then simply the sum of the first n interarrival times.

As we have seen in Section 3.3, the interarrival times τ_i of a Poisson process with rate λ are independent and identically distributed according to an exponential distribution with parameter λ, so

$$f_\tau(t) = \begin{cases} \lambda e^{-\lambda t} & \text{if } t \geq 0, \\ 0 & \text{if } t < 0. \end{cases}$$

The cumulative distribution function corresponding to the probability density function of the interarrival times of the Poisson process, then is

$$F_\tau = 1 - e^{-\lambda t},$$

if $t > 0$,

6.1.2 The Theory Behind the Simulation

To simulate a certain variable, many methods can be used. The method used in this Thesis is called the inverse transformation method. This method in based on Proposition 2.

Proposition 2 ([26], proposition 11.1) Let U be a uniform $(0, 1)$ random variable. For any continuous distribution function F if we define the random variable X by $X = F^{-1}(U)$, then the random variable X has distribution function F.

Let us prove Proposition 2. (This proof follows [26], Proposition 11.1)

Proof: F is the continuous distribution function corresponding to X, thus $F_X(a) = \mathbb{P}(X \leq a)$. Since $X = F^{-1}(U)$, we can write $F_X(a) = \mathbb{P}(F^{-1}(U) \leq a)$. In [26] (page 650) it is stated that $F(a)$ is a monotone function, from which it follows that $F^{-1}(U) \leq a$ if and only if $U \leq F(a)$. Combining $F_X(a) = \mathbb{P}(F^{-1}(U) \leq a)$ and $F^{-1}(U) \leq a$ if and only if $U \leq F(a)$, gives us $F_X(a) = \mathbb{P}(F^{-1}(U) \leq a) = \mathbb{P}(U \leq F(a)) = F_U(a)$.

Q.E.D.

As is clear from Proposition 2, this method is very simple and that is also the reason why this method is chosen in this Thesis. A disadvantage of the inverse transformation method is that a closed form of the cumulative distribution function is needed, but this is not a concern here, since we have already found the cumulative distribution function corresponding to the interarrival times.

6.1.3 The Algorithm for the Simulation

To simulate the interarrival times τ we make use of the Proposition 2. We can sample τ by sampling $\tau^* = {F_\tau}^{-1}(U)$, where $U \sim \text{Uni}(0, 1)$.

However, before we can sample an interarrival time of the Poisson process, we first need to find ${F_\tau}^{-1}$. To find a closed form expression for ${F_\tau}^{-1}$ we first replace F_τ by t and t by ${F_\tau}^{-1}$ in the expression for F_τ: so $F_\tau = 1 - e^{-\lambda t}$ becomes $t = 1 - e^{-\lambda {F_\tau}^{-1}}$. Rewritting the last expression, we find ${F_\tau}^{-1} = \frac{-\ln(1-t)}{\lambda}$.

Now we have found the expression to sample the interarrival times, we can set up an algorithm that can be used to simulate the Poisson process. The algorithm to simulate the Poisson process can be found below as Algorithm 1. The algorithm, although slightly changed in notation, is taken from [6] (Algorithm 1). The set up of the algorithm will now be explained.

When making an algorithm, it is always important to first clearly state what result is wanted, such that it is clear what the outcome of the algorithm will be. In this case the result will be a set of event times corresponding to a Poisson process.

After we have denoted what outcome we expect, we denote what input should be given. In this case the rate λ of the process and the end time T of the interval over which we simulate should be listed.

To have a good working algorithm, we need a variable n that keeps track of how many elements there are in the Poisson process and we need to state that no events occured at $t = 0$, thus $T_0 = 0$ (line 1 of Algorithm 1).

Now we are able to generate the event times of the process. Based on Proposition 2, we first generate $u \sim \text{Uni}(0, 1)$ (line 3), after which we can simulate the new interarrival time τ as $\tau = -\frac{\ln(u)}{\lambda_0}$ (line 4). Note that in the above we found $F_\tau{}^{-1} = \frac{-\ln(1-t)}{\lambda}$, however since $u \sim \text{Uni}(0, 1)$, it does not matter if we replace $1 - u$ by u, since both are uniform random variables between zero and one. The new event time is now equal to the old event time plus the new interarrival time (line 5). We will repeat these steps of generating a new interarrival time and then adding it to the old event time, as long as $T_{n+1} \leq T$, since then the event time is not bigger than the interval over which we simulate our process. So if $T_{n+1} > T$ we stop the simulation and we need to get back the values of the event times (steps 6 and 7). If $T_{n+1} \leq T$, we raise our counter by one and repeat the simulation (steps 8 and 9).

Algorithm 1: Simulation of a homogeneous Poisson process with rate λ on $[0, T]$.

```
   Result: Set of event times ${T_1, ..., T_n}$.
   Input  : $\lambda, T$.
1  Initialize $n = 0$, $T_0 = 0$;
2  while True do
3      Generate $u \sim \text{Uni}(0, 1)$;
4      Let $\tau = -\frac{\ln(u)}{\lambda}$;
5      Set $T_{n+1} = T_n + \tau$ ;
6      if $T_{n+1} > T$ then
7          return ${T_1, ..., T_n}$ ;
8      else
9          Set $n = n + 1$;
10     end
11 end
```

6.1.4 Running the Simulation

Based on Algorith 1 a Matlab code has been written to simulate a Poisson process. The code that has been used to produce the graphs in this Subsection can be found in Appendix D1.

In Figure 6.1 a simulation of a homogeneous Poisson process with rate $\lambda = 2$ can be seen. From the graph we observe that there occured 29 events in the simulation interval.

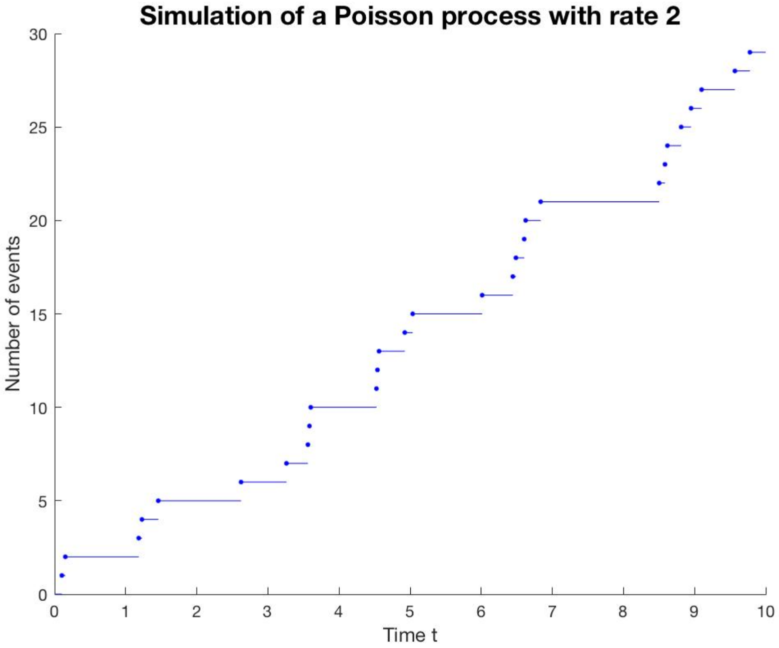


Figure 6.1: A simulation of a homogeneous Poisson process with rate $\lambda = 2$ on the interval $[0, 10]$.

In Figure 6.2 a simulation of a homogeneous Poisson process with rate $\lambda = 4$ can be seen. From the graph we observe that there occured 39 events in the simulation interval.

Comparing Figure 6.1 with Fig 6.2 shows us graphically some things we noticed theoretically in Section 3.3. It can be seen that the interarrival times of the process with the rate of $\lambda = 4$ are shorter than the interarrival times of the process with rate $\lambda = 2$. Since the interarrival times of the process with rate $\lambda = 4$ are shorter, there occur more events in the same time for the process with the higher rate. These were two things to be expected, since a larger rate implies more events, thus smaller interarrival times.

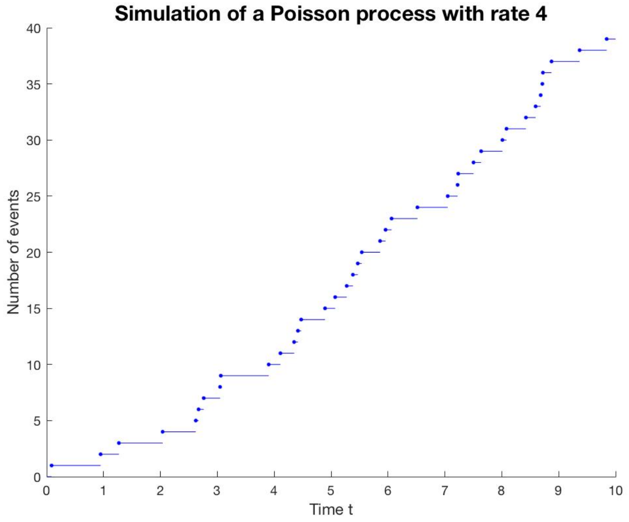


Figure 6.2: A simulation of a homogeneous Poisson process with rate $\lambda = 4$ on the interval $[0, 10]$.

6.1.5 Checking Properties

Properties of a Poisson process can be checked by making use of a simulation study.

The most interesting property to check, is Theorem 1. This Theorem states that $N(s+t) - N(s)$ follows a Poisson random variable with mean λt. Although the Theorem states $s > 0$, we look at the case where $s = 0$, since then we can look of the distribution of the number of events taken place at time t starting from 0. The Theorem says

$$\mathbb{P}(N(t) = n) = \frac{e^{-\lambda t}(\lambda t)^n}{n!}.$$

To test the Theorem, we simulate a Poisson process with rate 4 10000 times (Matlab code listed in Appendix C). For every simulation we save the number of events that have occured at the end of the simulation. The histogram of this simulation is seen in Figure 6.3.

According to the Theorem, we should fine that $N(T)$ follows a Poisson distribution with parameter 40.

If we fit a Poisson distribution to our data, the parameter found by Matlab is 40.1253. The 95% confidence interval of the parameter is $[40.0011, 40.2495]$. Although the value of the parameter and the confidence interval are in very good approximation equal to 40, the found parameter is slightly above the true value. This can be due to some simulation error.

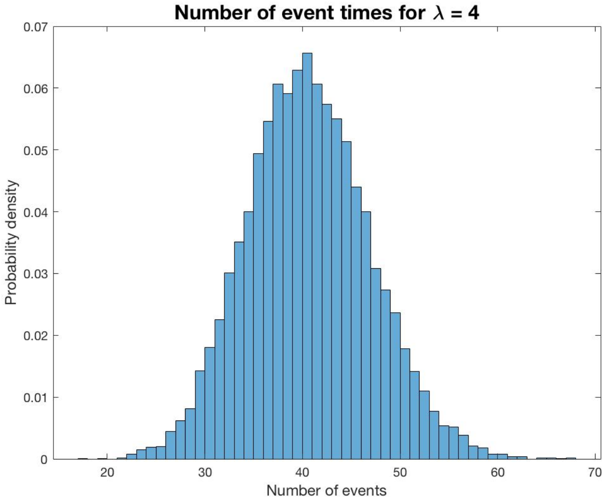


Figure 6.3: A Poisson process with rate $\lambda = 4$ has been simulated 10000 times over the interval $[0, 10]$. A histogram in which the number of events that have occured at the end of each simulation is shown.

In Figure 6.4 the fitted and actual distribution are shown in the histogram. Also here we observe that the fitted distribution is in very good approximation equal to the actual distribution: the red line of the fitted distribution is almost completely below the orange line from the actual distribution.

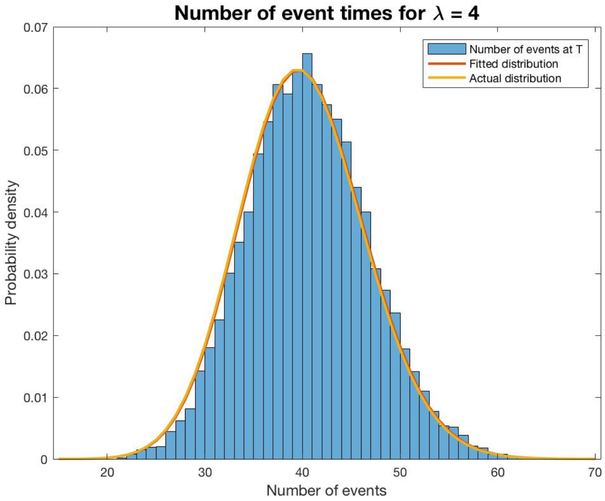


Figure 6.4: The fitted and actual Poisson distributions are plotted together with the histogram.

6.2 The Nonhomogeneous Poisson Process

6.2.1 Recap of Nonhomogeneous Poisson Processes

Now we want to simulate a nonhomogeneous Poisson process. Suppose we want to simulate a nonhomogeneous Poisson process with intensity function $\lambda(t)$ over the interval $[0, T]$. The simulation should then give us a list of event times $\{T_1, ..., T_n\}$ of all the n events that took place between 0 and T.

As was the case with the homogeneous Poisson process, we can not simulate the event times directly: we will simulate the interarrival times, after which we calculate the event times.

The difficulty with the nonhomogeneous Poisson process, compared to the homogeneous Poisson process, is that the interarrival times of a nonhomogeneous Poisson process do not follow a specific distribution, whereas the interarrival times of a homogeneous Poisson process do.

In order to simulate a nonhomogeneous Poisson process, we make use of the fact that we can simulate a nonhomogeneous Poisson process by first simulating a homogeneous Poisson process, which is something we can do.

To simulate a nonhomogeneous Poisson process based on a homogeneous Poisson process, we make use of the thinning property of Poisson processes.

Thinning Property ([25], in text, page 10): Consider a Poisson process with rate λ. This single process can now be split into two independent Poisson processes with rate λ_1 and λ_2 respectively, as long as $\lambda = \lambda_1 + \lambda_2$.

The thinning property shows us, we can sample a nonhomogeneous Poisson process by first simulating a homogeneous Poisson process with rate $\bar{\lambda} \geq \lambda(t)$, for all the values of t in the interval over which we make the simulation. After we have simulated the homogeneous Poisson process, we can thin the process to a nonhomogeneous Poisson process, by selecting certain points based on Theorem 6. ([25], in text, page 10)

6.2.2 The Theory Behind the Simulation

In the previous subsection we have already seen that we simulate the nonhomogeneous Poisson process by thinning a homogeneous Poisson process. The Theorem on which this method is based, is given below.

Theorem 6 ([6], slightly adapted, Theorem 4.2) Consider a homogeneous Poisson process $\bar{N}(t)$ with intensity function $\bar{\lambda}$. Let $\bar{T}_1, \bar{T}_2, ..., \bar{T}_n$ be the [event times of the homogeneous Poisson process, which are] in the interval $(0, T]$. Suppose that for $0 \leq t \leq T$, $0 \leq \lambda(t) \leq \bar{\lambda}$. For $k = 1, 2, ..., n$, delete the point $\bar{T}_k$ with probability $1 - \lambda(\bar{T}_k)/\bar{\lambda}$; then the remaining points form a point process $N(t)$ satisfying

$$\mathbb{P}(N(t+s) - N(s) = 0) = e^{-\int_s^{t+s} \lambda(y)dy}.$$

Proof: See for example [6] Theorem 4.2.

Q.E.D.

As we have seen in Section 3.3, point processes satisfying $\mathbb{P}(N(t+s)-N(s) = 0) = e^{-\int_s^{t+s}\lambda(y)dy}$ form a nonhomogeneous Poisson process, so applying a method based on Theorem 6 indeed gives us a nonhomogeneous Poisson process.

6.2.3 The Algorithm for the Simulation

As seen in the previous subsections we can simulate a nonhomogeneous Poisson process by first simulating a homogeneous Poisson process and then deleting points according to a certain probability. The algorithm to simulate the nonhomogeneous Poisson process can be found below as Algorithm 2. The algorithm, although slightly changed, is taken from [6] (Algorithm 2). The set up of the algorithm will now be explained.

As was the case with the simulation of a homogeneous Poisson process, the result of the simulation should be a list of event times corresponding to a nonhomogeneous Poisson process.

To simulate a nonhomogeneous Poisson process, we need to give as input the intensity function $\lambda(t)$ and the end time T of the interval over which we simulate the process.

For the initialization, we now have to look at three different things: counters for the number of events, the event times, and the values of $\bar{\lambda}$, which is the rate of the homogeneous Poisson process. For the number of events and the event time we need specific variables for both the homogeneous (n and S_0) and nonhomogeneous (m and T_0) Poisson process. Both counters start at 0. Also $S_0 = T_0 = 0$, since no events occured at time $t = 0$. The rate of the homogeneous Poisson process, depends on the intensity function. According to Theorem 6 we should have $\bar{\lambda} \geq \lambda(t)$ over all values t over which we simulate, so we choose $\bar{\lambda} = \sup_{0 \leq t \leq T} \lambda(t)$. These steps are all in line 1.

We are now able to simulate the nonhomogeneous Poisson process. We keep on generating new points, as long as $S_m < T$, so as long as the m-th event time of the homogeneous Poisson process is not bigger than the end point of our simulation interval (line 2). Based on the Proposition 2, we know we can simulate a new interarrival time σ of the homogeneous Poisson process, by first generating a random variable $u \sim \text{Uni}(0,1)$ (lines 3 and 4). The new event time S_{m+1} of the homogeneous Poisson process is then equal to the old event time S_m plus the generated interarrival time σ (line 5). We can now use Theorem 6 to check whether the event time of the homogeneous Poisson process also corresponds to an event time in the nonhomogeneous Poisson process. To do this, we first generate a random variable $D \sim \text{Uni}(0,1)$ (line 6), after which we check if $D \leq \lambda(S_{m+1})/\bar{\lambda}$ is true (line 7). If the last condition is true, the new generated event also belongs to the nonhomogeneous Poisson process, thus $T_{n+1} = S_{m+1}$ (line 8), so we also have to update our counter (line 9). If $D > \lambda(S_{m+1})/\bar{\lambda}$, we just simply raise our counter of the homogeneous Poisson

Algorithm 2: Simulation of a nonhomogeneous Poisson process with bounded intensity function $\lambda(t)$, on $[0, T]$.

```
   Result: Set of event times $\{T_1, ..., T_n\}$.
   Input  : $\lambda(t), T$.
1  Initialize $n = m = 0$, $T_0 = S_0 = 0$, $\bar{\lambda} = \sup_{0 \leq t \leq T} \lambda(t)$;
2  while $S_m < T$ do
3      Generate $u \sim \text{Uni}(0, 1)$;
4      Let $\sigma = -\ln(u)/\bar{\lambda}$;
5      Set $S_{m+1} = S_m + \sigma$ ;
6      Generate $D \sim \text{Uni}(0, 1)$ ;
7      if $D \leq \lambda(S_{m+1})/\bar{\lambda}$ then
8          $T_{n+1} = S_{m+1}$ ;
9          $n = n + 1$ ;
10     end
11     $m = m + 1$;
12 end
13 if $T_n \leq T$ then
14     return $\{T_1, ..., T_n\}$;
15 else
16     return $\{T_1, ..., T_{n-1}\}$;
17 end
```

process (line 11) and start the steps over again. However, if now $S_m \geq T$ the simulation is finished. The only thing we have to do is to check whether the last found event time of the nonhomogeneous Poisson process T_n is bigger than T (line 13). If $T_n \leq T$, we just return $\{T_1, ..., T_n\}$ (line 14). If $T_n > T$, we return $\{T_1, ..., T_{n-1}\}$ (line 16).

6.2.4 Running the Simulation

Based on Algorithm 2 a Matlab code has been written to simulate a nonhomogeneous Poisson process. The code that has been used to produce the graphs in this subsection can be found in Appendix D2.

In Figure 6.5 a simulation of a nonhomogeneous Poisson process over the interval $[0, 10]$ is shown. The solid blue line is the intensity function of the nonhomogeneous Poisson process. The intensity function is $\lambda(t) = 1 + \sin(t)$. The rate of the homogeneous Poisson process equals $\bar{\lambda} = 2$.

The dashed line is the supremum of the intensity function over the simulation interval, so $\bar{\lambda} = 2$, which is also the rate of the homogeneous Poisson process. All the points on the x-axis, thus the green and red points together are the points that are in the homogeneous Poisson process. The green points are the points that end up in the nonhomogeneous Poisson process and the red points are the points that are in the homogeneous Poisson process but that are declined based on Theorem 6. The higher points show the value of $\bar{\lambda}$ times the value of a random variable $D \sim \text{Uni}(0, 1)$. The points that have a value higher than the value of the intensity function are not added to the nonhomogeneous Poisson process.

Looking at Figure 6.5, we count that there have occured 15 events at the end of the simulation interval for the nonhomogeneous Poisson process.

The number of events that have occured at the end of the simulation interval, depends mainly on the value of $\bar{\lambda}$. Theorem 6 states that we should use a $\bar{\lambda}$ such that $\bar{\lambda} \geq \lambda(t)$ for all $t \in [0, T]$. However, in our algorithm we choose $\bar{\lambda} = \sup_{0 \leq t \leq T} \lambda(t)$. Theorem 6 is not really restrictive on the value of $\bar{\lambda}$: as long as $\bar{\lambda}$ is bigger than $\lambda(t)$ on all values of $t \in [0, T]$, it is fine. However, a very large value of $\bar{\lambda}$ is not preferred. We know $\bar{\lambda}$ to be the rate of the homogeneous Poisson process, so if its rate is larger there are more points simulated. At the same time $\bar{\lambda}$ determines which points in the homogeneous Poisson process will be added to the nonhomogeneous Poisson process. A bigger value of $\bar{\lambda}$ will result in more points declined, so less points will be added to the nonhomogeneous Poisson process. A bigger value of $\bar{\lambda}$ will result in more points declined, so less points will end up in the nonhomogeneous Poisson process. These two effects show us a "small" value of $\bar{\lambda}$ is preferred, so in the algorithm we choose $\bar{\lambda} = \sup_{0 \leq t \leq T} \lambda(t)$. To check this fact, we runned our simulation again, but now with a value of $\bar{\lambda} = 3$. The result of this simulation can be seen in Figure 6.6. Although also in this case 15 events end up in the nonhomogeneous Poisson process, it is clear that more points of the homogeneous Poisson process are declined.

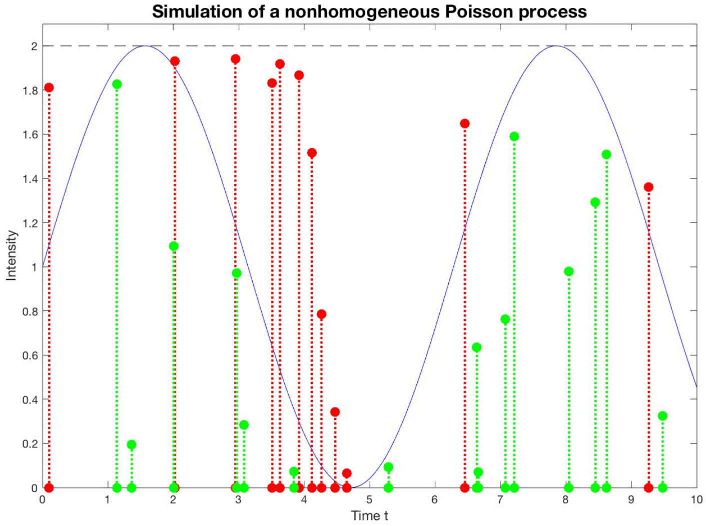


Figure 6.5: A simulation of a nonhomogeneous Poisson process. The value of the green points on the x-axis, represent the event times of the points that end up in the nonhomogeneous Poisson process. The rate of the homogeneous Poisson process is $\bar{\lambda} = 2$. The intensity function of the nonhomogeneous Poisson process is $\lambda(t) = 1 + \sin(t)$.

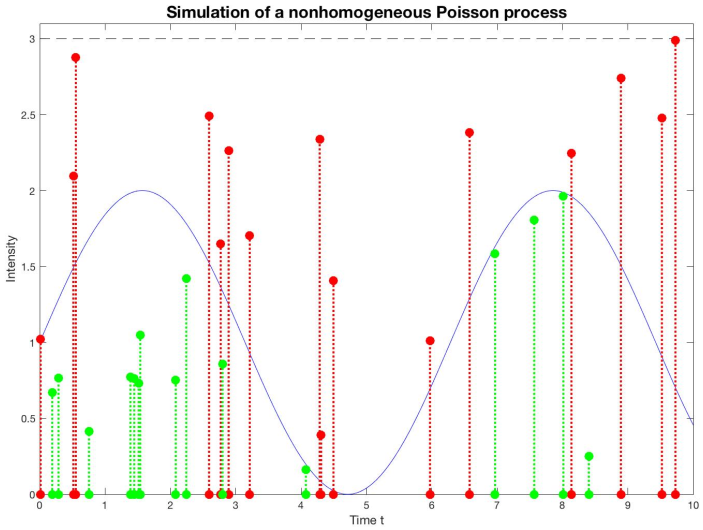


Figure 6.6: A simulation of a nonhomogeneous Poisson process. The value of the green points on the x-axis, represent the event times of the points that end up in the nonhomogeneous Poisson process. Now $\bar{\lambda}$ is not equal to the supremum of the intensity function. The rate of the homogeneous Poisson process is $\bar{\lambda} = 3$. The intensity function of the nonhomogeneous Poisson process is $\lambda(t) = 1 + \sin(t)$.

6.2.5 Checking Properties

Like we did in the case of the homogeneous Poisson process, we now check a property of the nonhomogeneous Poisson process using a simulation.

Theorem 3 states that $N(t+s) - N(s)$ follows a Poisson random variable with mean $\int_s^{t+s} \lambda(y)dy$, thus

$$\mathbb{P}(N(t+s) - N(s) = n) = \frac{e^{-\int_s^{t+s} \lambda(y)dy} \left(\int_s^{t+s} \lambda(y)dy\right)^n}{n!}.$$

Theorem 3 shows us that we expect the distribution of the number of events n occured at time T starting from 0, is equal to

$$\mathbb{P}(N(T) = n) = \frac{e^{-\int_0^T \lambda(y)dy} \left(\int_0^T \lambda(y)dy\right)^n}{n!}.$$

To check Theorem 3, we simulate a nonhomogeneous Poisson process with intensity function $\lambda(t) = 1 + \sin(t)$ over the interval $[0, 10]$ 10000 times. For every simulation we save the number of events that have occured at the end of the simulation. The histogram of this simulation is seen in Figure 6.7.

According to the Theorem, we should find that $N(T)$ follows a Poisson distribution with parameter $\int_0^T \lambda(y)dy$. So in this case we expect a parameter of

$$\int_0^{10} (1+\sin(y))\,dy = [y-\cos(y)]_{y=0}^{y=10} = 11-\cos(10) \approx 11.8391.$$

If we fit a Poisson distribution to our data, the parameter found by Matlab is 11.8266. The 95% confidence interval of the parameter is $[11.7592, 11.894]$. We see the "true" value of the parameter is in the 95% confidence interval of the found parameter, thus the simulation is in good approximation equal to the true parameter. So Theorem 3 and the simulation study are supportive of each other.

In Figure 6.7 the fitted and actual distribution are shown in the histogram. Also here we observe that the fitted distribution is in very good approximation equal to the actual distribution: the orange line of the actual distribution is almost completely above the red line of the fitted distribution.

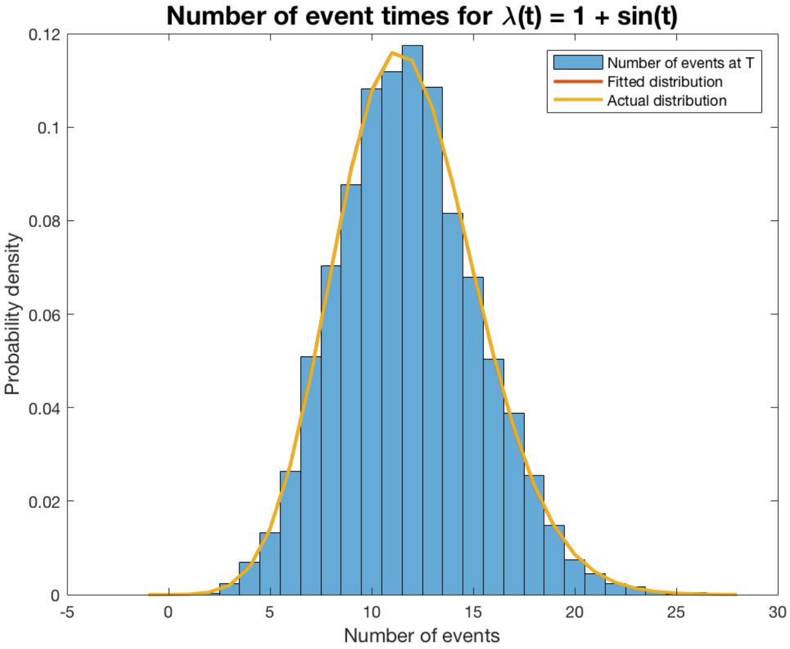


Figure 6.7: A Poisson process with intensity $\lambda(t) = 1+\sin(t)$ has been simulated 10000 times over the interval $[0, 10]$. A histogram in which the number of events that have occured at the end of each simulation is shown. Also the fitted and actual distribution are plotted.

One thing which is also good to check, is the fact that the distribution of the number of events n occured at time T starting from 0, does not depend on $\bar{\lambda}$. We have already seen that a value of $\bar{\lambda}$ which is higher than $\sup_{0\leq t\leq T}\lambda(t)$, results in more points of the homogeneous Poisson process being rejected. However, this does not affect the distribution of the nonhomogeneous Poisson process, as can be seen in Theorem 3: $\mathbb{P}(N(T))$ does not depend on $\bar{\lambda}$.

In Figure 6.8 the histogram with the actual and fitted distribution can be seen. The parameter found is 11.7903, with a 95% confidence interval of [11.723, 11.8576]. The true value of the parameter lies within the 95% confidence interval and the graphs are almost completely overlapping, thus we see that $\bar{\lambda}$ indeed does not affect the distribution of $\mathbb{P}(N(T))$.

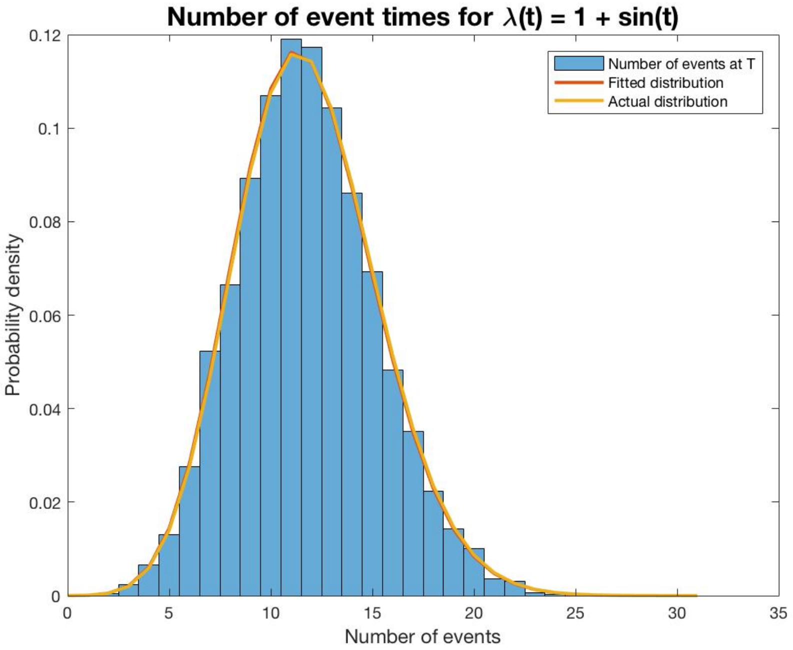


Figure 6.8: A Poisson process with intensity $\lambda(t) = 1+\sin(t)$ has been simulated 10000 times over the interval $[0, 10]$. A histogram in which the number of events that have occured at the end of each simulation is shown. Compared to Figure 6.7, now $\bar{\lambda}$ is not equal to $\sup_{0\leq t\leq T}\lambda(t)$: now $\bar{\lambda} = 3 > \sup_{0\leq t\leq T}\lambda(t)$. Also the fitted and actual distribution are plotted.

Chapter 7

Simulation of Hawkes Processes

In Chapter 6 we have simulated (nonhomogeneous) Poisson processes. Now we will take a look at simulating Hawkes processes. There are a lot of different ways a Hawkes process can be simulated. In this Chapter we will look at three different ways to simulate such a process. First we will look at Ogata's Algorithm: this is a algorithm that is based on the thinning property, but then applied to a Hawkes process. Then we will look at Dassios and Zhao's Algorithm, which is an algorithm that is more efficient than Ogata's Algorithm when a Hawkes process with exponential kernel is simulated. Finally we will look at an algorithm that is based on the branching factor.

7.1 Ogata's Algorithm

7.1.1 Recap of Hawkes Processes

As we have seen in Chapter 4 a Hawkes process is a specific kind of nonhomogeneous Poisson process. Compared to the nonhomogeneous Poisson process, the Hawkes process has a conditional intensity function instead of an intensity function. The conditional intensity function is explicitly dependent on the history of the process:

$$\lambda^*(t) = \lambda_0 + \sum_{i:t>T_i} \mu(t - T_i). \tag{7.1}$$

The conditional intensity function consists of a background term λ_0 and a kernel function $\mu(.)$. For the kernel lots of functions can be chosen. The choice for a specific kernel depends on what has to be modelled.

Now we want to simulate such a Hawkes process. Suppose we want to simulate a Hawkes process with conditional intensity function $\lambda^*(t)$ over the interval $[0, T]$. The simulation should then give us a list of event times $\{T_1, ..., T_n\}$ of all the n events that took place between 0 and T.

As we have seen when we simulated the (nonhomogeneous) Poisson process, the event times can not be simulated directly. Instead, we first simulate the interarrival times and then calculate the event times.

7.1.2 The Theory Behind the Simulation

Given Equation 7.1 the conditional intensity function of the Hawkes process can be written in the following way:

$$\lambda^*(t) = \begin{cases} \lambda_0 & \text{if } 0 \leq t < T_1, \\ \lambda_0 + \mu(t - T_1) & \text{if } T_1 \leq t < T_2, \\ \lambda_0 + \mu(t - T_1) + \mu(t - T_2) & \text{if } T_2 \leq t < T_3, \\ \vdots \end{cases} \tag{7.2}$$

Let us imagine our Hawkes process already consists of k points: $\{T_1, ..., T_k\}$. Given Equation 7.2 we see that on the interval $[T_k, T_{k+1})$ the conditional intensity function is completely known, given we know all the previous event times:

$$\lambda^*(t) = \lambda_0 + \sum_{i=1}^{k} \mu(t - T_i), \tag{7.3}$$

where $t \in [T_k, T_{k+1})$. As a result, we see that if our Hawkes process already consists of the event times $\{T_1, ..., T_k\}$, the generation of the next point in the Hawkes process can be seen as the generation of a point in a nonhomogeneous Poisson process with the known intensity function given by Equation 7.3. ([6])

As we have seen in Section 6.2, a nonhomogeneous Poisson process can be simulated by thinning. A Hawkes process can also be simulated by thinning. However, the value $\bar{\lambda}$ should be chosen in a different way, since this value is now also dependent on the history. Since we choose our kernel function to be monotonically decreasing, the value of $\lambda^*(t)$ on $[T_k, T_{k+1})$ is maximum for T_k, thus we chose $\bar{\lambda} = \lambda^*(T_k)$. However, now we also have to update the value of $\bar{\lambda}$, since after a new point is added to the Hawkes process, the value of $\bar{\lambda}$ should be calculated for the new set of event times.

7.1.3 The Algorithm for the Simulation

The Hawkes process can be simulated in a way very similar to simulating a nonhomogeneous Poisson process. The algorithm to simulate the Hawkes process can be found below as Algorithm 3. This Algorithm is known as Ogata's Algorithm. The algorithm, although slightly changed in notation and steps, is taken from [6] (Algorithm 3). The set up of the algorithm will now be explained.

In the explanation of the algorithm we will be talking about the nonhomogeneous Poisson process, but it is good to note that we are not simulating a real nonhomogeneous Poisson process over here. What we are doing is only generating the first point of a nonhomogeneous Poisson process corresponding to a specific value of $\bar{\lambda}$. If a point is added to the Hawkes process, we choose a new value of $\bar{\lambda}$, so the new nonhomogeneous Poisson point, does not come from the same nonhomogeneous Poisson process as the first.

As a result of our simulation we want to have a list of event times coore-sponding to a Hawkes process.

To simulate a Hawkes process, we need to give as input the background parameter α, the kernel $\mu(.)$ and the end time T of the interval over which we simulate the process.

For the initialization, we now have to look at three different things: counters for the number of events, the event time of a nonhomogeneous Poisson process, and a set with the event times of our Hawkes process. For the number of events we set a counter for the Hawkes process (n) and a counter for the nonhomogeneous Poisson process (m). Both counters start at 0. The event time of the nonhomogeneous Poisson process S_0 is initialy 0, since no events occured at time $t = 0$. The set in which the events of the Hawkes process are stored ($\mathcal{T}$) is empty at the beginning, since there have not occured any events at the start of the simulation. These steps are all in line 1.

We are now able to simulate the Hawkes process. We keep on generating new points, as long as $S_m < T$, so as long as the m-th event time of the nonhomogeneous Poisson process is not bigger than the end point of our simulation interval (line 2). First we need to choose the current value of $\bar{\lambda}$. We have seen that we choose $\bar{\lambda} = \lambda^*(S_m)$, so the intensity for the possible new Hawkes point (line 3). Based on Proposition 2, we know we can simulate a new interarrival

Algorithm 3: Simulation of a univariate Hawkes process, on $[0, T]$.

```
   Result: Set of event times ${T_1, ..., T_n}$.
   Input  : $\lambda_0, \mu(.), T$.
1  Initialize $n = m = 0$, $S_0 = 0$, $\mathcal{T} = \emptyset$;
2  while $S_m < T$ do
3      Set $\bar{\lambda} = \lambda_0 + \sum_{\alpha \in \mathcal{T}} \mu(S_m - \alpha)$;
4      Generate $u \sim \text{Uni}(0, 1)$;
5      Let $w = -\ln(u)/\bar{\lambda}$;
6      Set $S_{m+1} = S_m + w$ ;
7      Generate $D \sim \text{Uni}(0, 1)$ ;
8      if $D \leq \lambda(S_{m+1})/\bar{\lambda}$ then
9          $T_{n+1} = S_{m+1}$ ;
10         $\mathcal{T} = \mathcal{T} \cup \{T_{n+1}\}$;
11         $n = n + 1$ ;
12     end
13     $m = m + 1$;
14 end
15 if $T_n \leq T$ then
16     return $\{T_1, ..., T_n\}$
17 else
18     return $\{T_1, ..., T_{n-1}\}$
19 end
```

time of the homogeneous Poisson process, that forms the basis of the nonhomogeneous Poisson process, by first generating a random variable $u \sim \mathrm{Uni}(0,1)$ and then calculating $-\ln(u)/\bar{\lambda}$ (line 4 and 5). The new event time of the homogeneous Poisson process is then equal to the old value plus the generated interarrival time (line 6). We can now use Theorem 6 to check whether the event time of the homogeneous Poisson process also corresponds to an event time in the nonhomogeneous Poisson process. To do this, we first generate a random variable $D \sim \mathrm{Uni}(0,1)$ (line 7), after which we check if $D \leq \lambda(S_{m+1})/\bar{\lambda}$ is true (line 8). If the last condition is true, the new generated event belongs to the nonhomogeneous Poisson process, so also to the Hawkes process, thus $T_{n+1} = S_{m+1}$. We now also have to add T_{n+1} to the set of event times of the Hawkes process (line 10) and we have to raise the counter for the Hawkes process (line 11). If $D < \lambda(S_{m+1})/\bar{\lambda}$ we just simply raise the counter of the nonhomogeneous Poisson process (line 13) and start the steps over again. However, if now $S_m \geq T$ the simulation is finished. The only thing we have to do is to check whether the last found event time of the Hawkes process T_n is bigger than T (line 15). If $T_n \leq T$, we just return $\{T_1, ..., T_n\}$ (line 16). If $T_n > T$, we return $\{T_1, ..., T_{n-1}\}$ (line 18).

7.1.4 Running the Simulation

We are going to run the simulation for both a Hawkes process with exponential kernel and a Hawkes process with power law kernel.

Exponential Kernel

To recall, the exponential kernel looks like

$$\mu(t - T_i) = \alpha e^{-\beta(t-T_i)},$$

where $\alpha, \beta > 0$.

Based on Algorithm 3 a Matlab code has been written to simulate a Hawkes process with exponential kernel. The code that has been used to produce the graphs in this Subsubsection can be found in Appendix D3.

In Figure 7.1 a simulation of a Hawkes process with exponential kernel with parameters $\alpha = 0.6$ and $\beta = 0.8$ can be seen. The background intensity parameters λ_0 equals 1.2. For these parameter values, the Hawkes process is stationary, since $\alpha/\beta = 0.6/0.8 = 0.75 < 1$.

The effect of the background intensity term can be seen clearly: until the first event takes place, the intensity equals 1.2, so the value of the background intensity parameter.

Although not completely clear, we can observe that as soon as an event occurs, the intensity raises by 0.6, which is the value of α, so α indeed corresponds to the increase the intensity immediately makes as soon as an event occurs.

The effect of β can be better seen by comparing Figure 7.1 with Figure 7.2 which shows a Hawkes process with parameters $\lambda_0 = 1.2$, $\alpha = 0.6$ and $\beta = 1.6$.

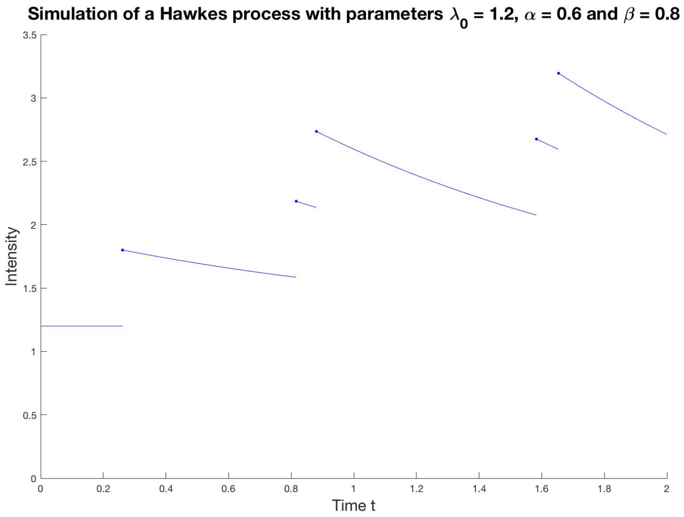


Figure 7.1: A Hawkes process with exponential kernel with parameters $\lambda_0 = 1.2$, $\alpha = 0.6$ and $\beta = 0.8$ has been simulated over the interval $[0, 2]$.

In Figure 7.2 the β parameter is twice as big as the β parameter in Figure 7.1; the rest is all the same, so also the Hawkes process in Figure 7.2 is stationary. We observe that the intensity for the higher value of β declines much faster than the intensity for the lower value of β, which makes sense since β determines the rate at which the intensity declines as soon as an event took place.

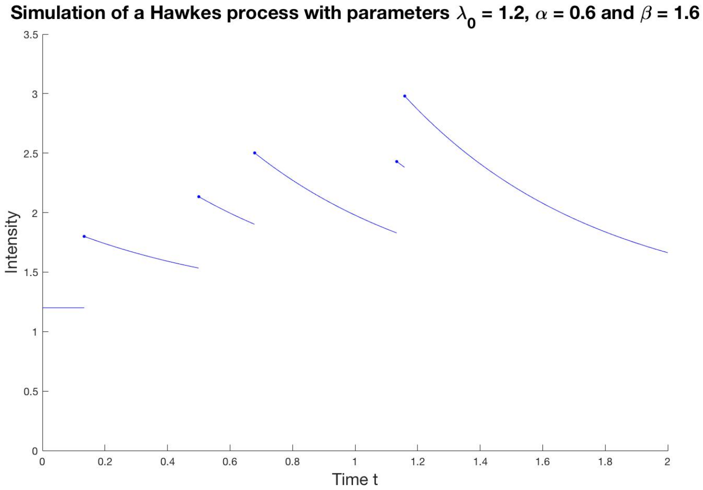


Figure 7.2: A Hawkes process with exponential kernel with parameters $\lambda_0 = 1.2$, $\alpha = 0.6$ and $\beta = 1.2$ has been simulated over the interval $[0, 2]$.

Comparing Figure 7.1 with Figure 7.2 we notice that both show Hawkes processes consisting of five elements. This is not we would have expected based on Equation 4.3; the expected number of events of a Hawkes process with exponential kernel can be calculated as

$$E[N(t)] = \lambda_0 t + \frac{\lambda_0 \alpha}{(\alpha - \beta)^2} \left(e^{(\alpha-\beta)t} - 1 - (\alpha - \beta)t \right).$$

So for the Hawkes process in Figure 7.1 we expect

$$E[N(2)] = 1.2 \cdot 2 + \frac{1.2 \cdot 0.6}{(0.6 - 0.8)^2} \left(e^{(0.6-0.8)\cdot 2} - 1 - (0.6 - 0.8) \cdot 2 \right) \approx 4 \quad (7.4)$$

events on average.

For the Hawkes process in Figure 7.2 we expect

$$E[N(2)] = 1.2 \cdot 2 + \frac{1.2 \cdot 0.6}{(0.6 - 1.6)^2} \left(e^{(0.6-1.6)\cdot 2} - 1 - (0.6 - 1.8) \cdot 2 \right) \approx 3 \quad (7.5)$$

events on average.

The reason the Hawkes processes in both Figure 7.1 and Figure 7.2 show five events, is because we have chosen to show this number of events. We have simulated both processes as long as one with five events showed up. This generalizations has than been graphed. The reason processes with five event times have been drawn is just basically because in this way more of the properties of the Hawkes process can be seen.

We have runned both simulations 10000 times and made histogram to check the distribution of the number of events occured at the end of the simulation interval. Both histograms are shown in Figure 7.3. The histogram of the Hawkes process with $\beta = 0.8$ shows that on average simulations with 3 and 4 events occur, which is in line with the result of Equation 7.4. The histogram of the Hawkes process with $\beta = 1.6$ shows that on average simulations with 3 events occur, which is in line with the result of Equation 7.5.

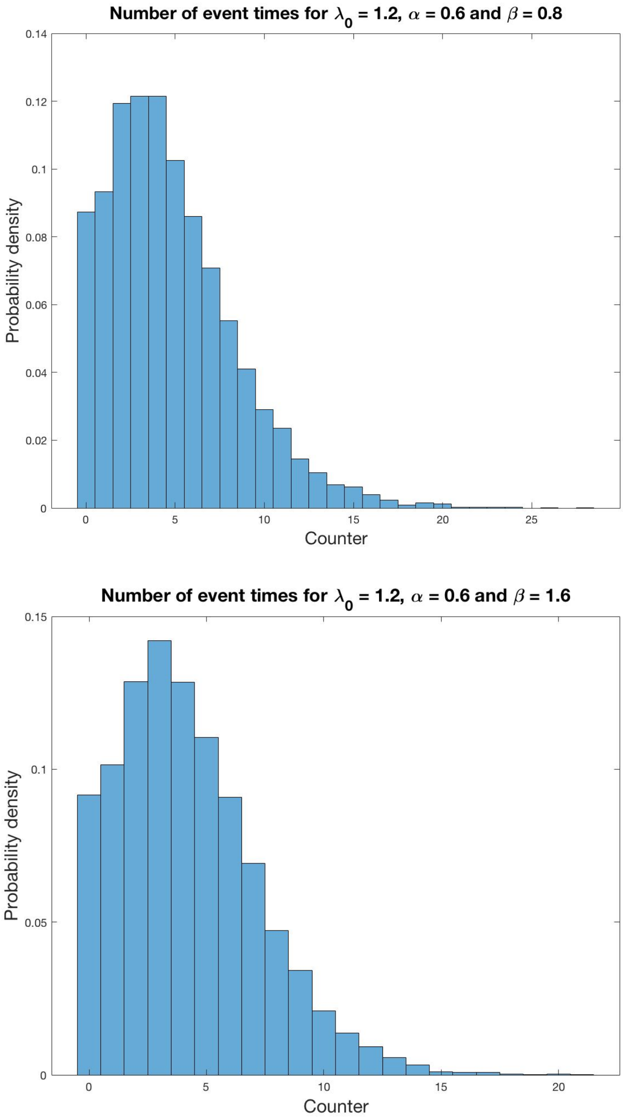


Figure 7.3: A Hawkes process with exponential kernel with parameters $\lambda_0 = 1.2$, $\alpha = 0.6$ and $\beta = 0.8$ (above) and $\lambda_0 = 1.2$, $\alpha = 0.6$ and $\beta = 1.6$ has been simulated 10000 times over the interval $[0, 2]$. Histograms are shown in which the number of evens that have occured at the end of each simulation is shown.

Power law kernel

To recall, the power law kernel looks like

$$\mu(t - T_i) = \frac{K}{(c + (t - T_i))^p},$$

where K, c and p are three constants.

Based on Algorithm 3 a Matlab code has been written to simulate a Hawkes process with power law kernel. The code that has been used to produce the graphs in this Subsubsection can be found in Appendix D4.

In Figure 7.4 a simulation of a Hawkes process with power law kernel with parameters $\lambda_0 = K = c = 1$ and $p = 2$ can be seen.

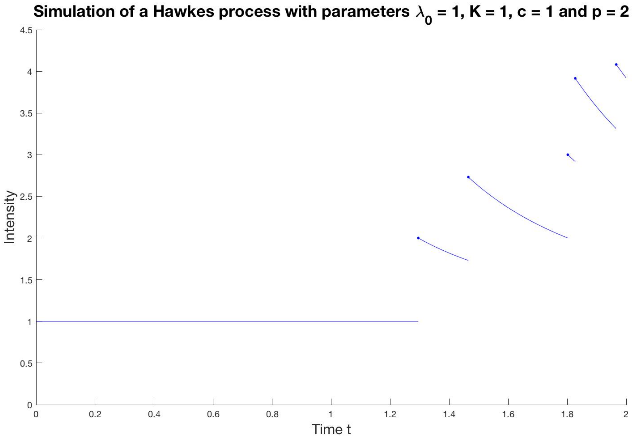


Figure 7.4: A Hawkes process with power law kernel with parameters $\lambda_0 = K = c = 1$ and $p = 2$ has been simulated over the interval $[0, 2]$.

We have not looked very extensive to the power law kernel yet, since the power law kernel is not our main research interest. In Chapter 9, we will look at this power law kernel in more detail when we are talking about a specific application of the Hawkes process.

However, something which is interesting to look at is the histogram we get of the number of events that have occured in the simulation interval when we run the simulation 10000 times. The histogram can be seen in Figure 7.5. We notice that at the end of the simulation, in most cases no events have taken place in our Hawkes process. Simulating a Hawkes process with power law kernel over the interval $[0, 2]$ is thus not a good idea, when we are using the parameters $\lambda_0 = K = c = 1$ and $p = 2$, since then it is very likely that no events have occured at the end of the simulation.

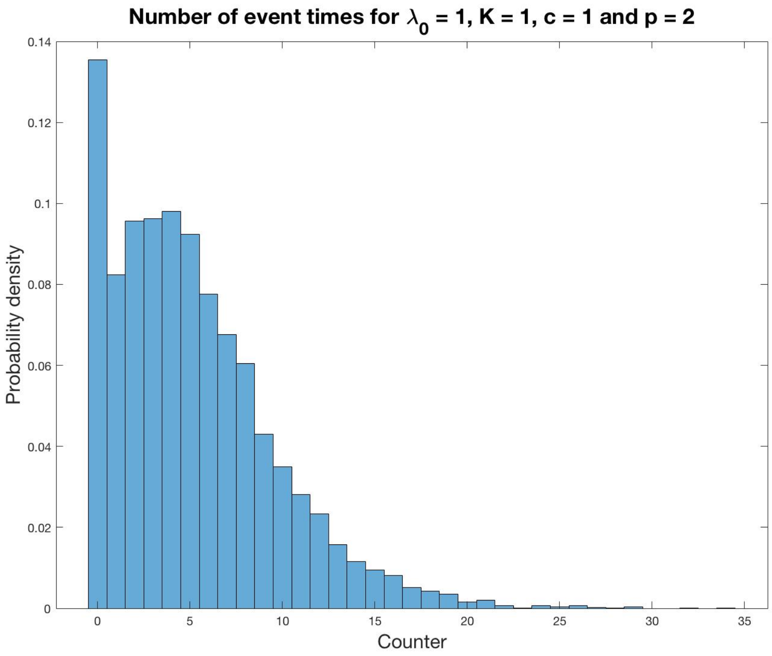


Figure 7.5: A Hawkes process with power law kernel with parameters $\lambda_0 = K = c = 1$ and $p = 2$ has been simulated 10000 times over the interval $[0, 2]$. A histogram is shown in which the number of evens that have occured at the end of each simulation is shown.

7.1.5 Complexitity

Although this algorithm is fairly simply, it is computationally not really fast. If we want to simulate a sample with N events, the complexity to compute just the event intensity according to

$$\lambda^*(t) = \lambda_0 + \sum_{t>T_i} \mu(t - T_i)$$

is $\mathcal{O}(N)$. So to sample N events, the complexity becomes $\mathcal{O}(N^2)$. ([25], page 11)

7.2 Dassios and Zhao's Algorithm

7.2.1 New Look at the Conditional Intensity Function

In 2013 Dassios and Zhao proposed a new algorithm for simulating Hawkes processes with exponential kernels. ([11]) In Subsection 7.2.4 we will explain why this new algorithm has been proposed and is in some cases more appropriate for simulating a Hawkes process than Ogata's algorithm.

To explain this algorithm, we have to define the conditional intensity function in a more general way than we did before in Section 4.2. The conditional intensity function of our Hawkes process with exponential kernel is now defined as

$$\lambda^*(t) = a + (\lambda_0 - a)e^{-\delta t} + \sum_{T_k < t} Y e^{-\delta(t - T_k)},$$

where $a \geq 0$ is the constant reversion level (the average value of the intensity over time), $\lambda_0 > 0$ is the background intensity, $\delta > 0$ is the constant rate of the exponential decay and Y are the sizes of the self-excited jumps, previously referred to as marks. ([11], Definition 2.1) More generally, we could have defined Y to be event dependent, so Y_k, and assign a distribution to Y_k but this is beyond the scope of this Thesis.

The conditional intensity function just described belongs to a Hawkes process with exponential immigrant rate and exponentional memory kernel. The exponential immigrant rate is described by a nonhomogeneous Poisson process following the exponential function $a + (\lambda_0 - a)e^{-\delta t}$. The exponential memory kernel is $\sum_{T_k < t} Y e^{-\delta(t - T_k)}$. ([25], page 11)

7.2.2 The Theory Behind the Simulation

This algorithm is based on the Markov property. Very simply said, a stochastic process is a Markov process, if it has the property that, conditional on the present, the future is independent of the past. ([25], in text, page 11)

In [22] it is showed that a process with an intensity function, is a Markov process if the kernel is exponential. So the Markov property can be applied to the conditional intensity function defined in Section 7.2.1.

7.2.3 The Algorithm for the Simulation

There is a small difference between the three algorithms we described before, and the new algorithm we will describe now. Algorithms 1 till 3 describe how we can simulate a certain process over the interval $[0, T]$. However, in the algorithm we will describe below, we will give an algorithm of how we can simulate N event times of a Hawkes process with exponential kernel. So we do not want to simulate our process over a certain time interval, instead we want to simulate N points in our process.

The algorithm to simulate the Hawkes process with exponential kernel can be found below as Algorithm 4. The algorithm, although slightly changed in notation, is taken from [11] (Algorithm 3.1) and [25] (Algorithm 2). The set up of the algorithm will now be explained.

To simulate the Hawkes process with exponential kernel, we need to give as input the parameters of our process (the constant reversion level a, the background parameter λ_0, the constant rate of the exponential decay δ and the size of the self-excited jumps Y) and the number of events we would like to have at the end of our simulation N (line 1).

For the initialization, we need to initialize the event time T_0 and the initial event rate $\lambda^*(T_0)$. The event time is initialy 0, since there have not occured any events at the start of the simulation: $T_0 = 0$. At the start of the simulation, the conditional intensity equals the background intensity, since there have not occured any events yet: $\lambda^*(T_0) = \lambda_0$. This is all listed in line 2.

To simulate the interarrival times we make use of the Markov property. Using this Markov property we can decompose the interarrival times into two independent random variables. ([25], page 11) We will have two random variables: s_0 and s_1. The random variable s_0 represents the interarrival time of the next event, if it were to come from a homogeneous Poisson process with rate a. Based on Proposition 2, we know we can simulate such a random variable by first generating a random variable $u \sim \text{Uni}(0,1)$ and then calculating $s_0 = \frac{-\ln(u_0)}{a}$ (line 3a). The random variable s_1 represents the interarrival time of the next event if it were to come from either the exponentional immigrant rate $a + (\lambda_0 - a)e^{-\delta t}$ or the exponential memory kernel $\sum_{T_k<t} Ye^{-\delta(t-T_k)}$. In [11] it is shown that s_1 can be generated by first calculating a certain value d (line 3b). If $d > 0$, then we can calculate the second interarrival time s_1 according to Proposition 2. The "overall" interarrival time τ_i now becomes the minimum of s_0 and s_1. If $d < 0$, then the interarrival time τ_i equals s_0, so in this case the interarrival time results from the constant rate. (line 3c) The new event time can now be calculated as the sum of the old event time T_{i-1} plus the simulated interarrival time τ_i (line 3d). Afterwards we need to update our conditional intensity (line 3e), and repeat the algorithm until we have simulated N events.

Algorithm 4: Simulation of a Hawkes process with exponential kernel according to Dassios and Zhao.

```
1. Input: $a$, $\lambda_0$, $\delta$, $Y$ and $N$.
2. Set $T_0 = 0$, initial event rate $\lambda^*(T_0) = \mu_0$.
3. For $i = 1, 2, ..., N$.
    (a) Draw $u_0 \sim \text{Uni}(0,1)$ and set $s_0 = \frac{-\ln(u_0)}{a}$.
    (b) Draw $u_1 \sim \text{Uni}(0,1)$ and set $d = 1 + \frac{\delta \ln(u_1)}{\lambda^*(T_{i-1}) - a}$.
    (c) If $d > 0$, set $s_1 = \frac{-\ln(d)}{\delta}$ and $\tau_i = \min\{s_0, s_1\}$. Otherwise, set
        $\tau_i = s_0$.
    (d) Record the $i$th event time $T_i = T_{i-1} + \tau_i$.
    (e) Update the event intensity: $\lambda^*(T_i) = (\lambda^*(T_{i-1}) - a)e^{-\delta\tau_i} + a + Y$.
```

7.2.4 Complexity and Limitations

Compared to Ogata's algorithm, the main benefit from this new algorithm is complexity wise. Since the algorithm by Dassios and Zhao does not rely on rejection sampling, the complexity of sampling N events is $\mathcal{O}(N)$. It is also more efficient since the intensity function can be updated in constant time, as can be seen in line 3e in Algorithm 4. Besides, this new Algorithm does not require stationarity conditions. ([11], page 5; [25], page 12)

The main drawback from this algorithm is that it can not be used to simulate Hawkes processes with power law kernels, since these kernels do not have the Markov property. ([25], page 12)

7.3 Algorithm Based on Branching Factor

The last algorithm we are going to give for simulating a Hawkes process, is based on the branching structure as defined in Section 4.4. The reason to also discuss this algorithm in this Thesis, is to show how the branching factor can be used to simulate a Hawkes process. Also the fact that the simulation algorithm is rather simple is a reason to include the algorithm in this Thesis. Besides, we have not yet seen a clear example of a branching structure yet, since they can not be observed for real life situations, thus this gives us a chance to simulate such a process to see what happens. However, since this algorithm is not of that great importance, we will not discuss it in great detail.

It is good to note that the algorithm we are given will again simulate a Hawkes process over the interval $[0, T]$, so the number of events occured is again a random variable.

The algorithm to simulate the Hawkes process as a branching structure can be found below as Algorithm 5. The algorithm, although slightly changed in notation, is taken from [21] (Algorithm 3). The set up of the algorithm will now be explained. We will only discuss the case with an exponential kernel.

To simulate the Hawkes process with exponential kernel, we need to give as input the parameters of our process: λ_0, α and β. Besides, we need to specify the end value of the interval over which we would like to simulate our process T.

For the initiliazation, we only need to initiliaze the set P in which the points in our Hawkes process are stored. At the start of the simulation P is an empty set (line 1).

To simulate the branching structure, we simulate first all immigrant events. In the branching structure, all immigrants arrive according to a homogeneous Poisson process with the background intensity as parameter, thus $\lambda = \lambda_0$. According to Theorem 1, we see that over the interval $[0, T]$ the number of events k is Poisson distributed with mean $\lambda_0 T$, so $k \sim \text{Pois}(\lambda_0 T)$ (line 3). According to [21] (in text, page 23), we can simulate the event times of the immigrants C_i, conditional on knowing there are k immigrants, by simulating uniform random variables over the interval $[0, T]$ (line 4). After we have simulated the

Algorithm 5: Simulation of exponential Hawkes process by clusters.

```
   Input : $\lambda_0, \alpha, \beta, T$.
1  $P \to \{\}$.
2  Immigrants:
3     $k \to \text{Pois}(\lambda_0 T)$
4     $C_1, ..., C_k \to \text{Uni}(0, T)$
5  Offspring:
6     $D_1, ..., D_k \to \text{Pois}(\alpha/\beta)$
7  for $i \to 1$ to $k$ do
8     if $D_i > 0$ then
9        $E_1, ..., E_{D_i} \to Exp(\beta)$.
10       $P \to P \cup \{C_i + E_1, ..., C_i + E_{D_i}\}$.
11    end
12 end
13 Remove offspring outside $[0, T]$:
14    $P \to \{P_i : P_i \in P, P_i \leq T\}$.
15 Add in immigrants and sort:
16    $P \to \text{Sort}(P \cup \{C_1, ..., C_k\})$.
17 return $P$
```

immigrant events, we are going to simulate their offspring. In the branching structure all offspring events form a nonhomogeneous Poisson process. We know the i-th immigrant his offspring to arrive with a conditional intensity of $\mu(t - C_i) = \alpha e^{-\beta(t-C_i)}$ for $t > C_i$. Denote by D_i the number of offspring associated with the i-th immigrant. We know the branching factor to be defined as the expected number of direct offspring spawned by a single event, thus $E[D_i] = n^* = \frac{\alpha}{\beta}$. According to [21] (in text, page 23), the expected number of offspring are now Poisson distributed with mean $n^* = \frac{\alpha}{\beta}$, thus we can simulate our D_i as $D_i \sim \text{Pois}(\frac{\alpha}{\beta})$ (line 6). Now we have simulated the number of offspring for each immigrant, we would like to simulate the offspring events. For each immigrant we simulate the offspring events (line 7). First we check whether D_i is bigger than zero, since no offspring should be generated if there is no offspring (line 8). Say all the offspring events of the i-th immigrant arrive at the times $(C_i + E_1, ..., C_i + E_{D_i})$. According to [21] (in text, page 23), the E_j, conditional on knowing D_i, are random variables distributed with the probability density function $\frac{\mu(.)}{n^*}$. In the case of the exponentional kernel, this becomes $E_j \sim \text{Exp}(\beta)$ (line 9). The set of events in the Hawkes process should now be the old set of events plus all the offspring events just generated (line 10). We now have to check whether some of the offspring events fall outside the interval $[0, T]$. If this is the case these events will be removed from P (lines 13 and 14). Lastly we should add all the immigrant events to our set of events in the Hawkes process (lines 15 and 16).

In Figure 7.6 an example of a simulation following Algorithm 4 is shown. The simulation has not been performed by ourselves, but is instead taken from [21].

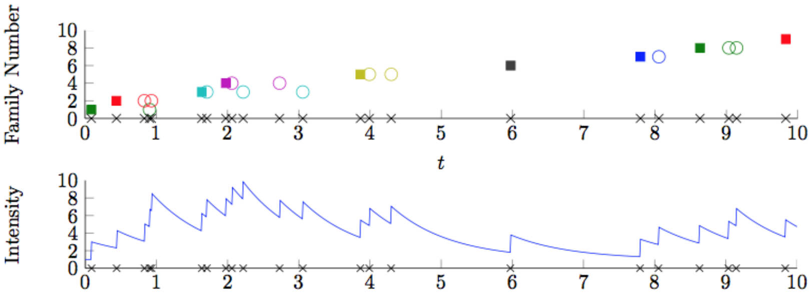


Figure 7.6: The branching structure of a Hawkes process with exponential kernel has been simulated. The parameters of the process are $\lambda_0 = 1$, $\alpha = 2$ and $\beta = 1.2$. In the plot above we see all the points that are in our branching structure. The immigrant events are plotted as squares. The offspring of an immigrant event are plotted as circles with the same color and height as the immigrant event. The intensity function is plotted below. All events that are in the Hawkes process, are drawn as crosses on the x-axis, both in the plot above and below. (Figure taken from [21], Figure 10)

Part IV

Applications

Chapter 8

Application in Finance: A Model for Sell and Buy Intensities

The use of Hawkes processes in financial mathematics and econometrics is widespread. The possibility to use self-exciting processes makes it possible to capture the effect of past events, something which is really important in a branch of study that mainly looks at the estimation of future events as a consequence of past events.

In this Chapter we will look a model that can be used to predict future imbalance of buy and sell trades in the FX market, conditional on the history of recent trade activities.

This Chapter is based on Patrick Hewlett his 2006 paper *Clustering of order arrivals, price impact and trade optimisation* ([16]). We would like to stress that all results and figures are taken from this paper, so no own estimations or figures are made. This Chapter is mainly here to give a brief introduction to how Hawkes processes can be used to model real life situations. Besides only a small part of the paper is discussed. Readers who wish to read more about this subject are encouraged to read the whole paper.

8.1 Introduction

In Hewlett his paper a model for the FX market is presented. The FX market is an abbreviation for the foreign exchange market. In [28] we read that all claims on foreign currency payable abroad, are foreign exchange. These claims may consist of funds in foreign currency held with banks abroad, or bills or cheques, again in foreign currency and payable abroad. The market in which all these foreign exchanges are regulated is called the foreign exchange market. In the FX market the exchange rates for global currencies are determined. The exchange rate is the rate of one currency in terms of another currency.

The FX market is arranged as a so called electronic limit order book. ([16]) In [18] we read a limit order to be an order to buy or sell a given quantity of stock at a specified limit price or better. So say, for example, a trader is looking to buy stocks from Apple, but he has a limit order of $50. This means our trader will only buy the stock when the price of the stock is $50 or lower. The electronic limit order book is the electronic collection of all outstanding limit orders for a certain stock. In an electronic limit order book, there is no specific liquidity provider. ([16]) A liquidity provider, which we will refer to as "market-maker", can be seen as the person who maintains the order book. In the model presented by Hewlett, it is assumed that there is only a single market-maker who sets competitive prices. Prices can be set, among others, competitively or consumer-oriented. If prices are set competitively, then prices are set based on the competition, rather than at the behaviour of consumers, which is what is looked at in consumer-oriented pricing.

For the FX market, several features are well known ([16]):

- trading activity tends to cluster in time,
- trades of the same sign tend to cluster together in the sequence of buys and sells.

The fact that in the FX market certain clusters are formed, explains why we are going to fit a Hawkes process to the data. In the Hawkes process certain clusters of events are expected, due to the self-exciting nature of these processes. If an event occurs, the likelihood of having a new event increases, so the forming of clusters is a characteristic of a Hawkes process.

If we would have had the data that has been used in [16], we would have been able to make a graph in which all the events occured were shown. In this graph the clustering effect should be noticable.

Using multivariate point processes the clustering of these buy and sell intensities can be modelled.

8.2 The Model

We will now consider how market-makers would set prices if order arrivals were governed by a Hawkes process. We denote the counting process for arrival of buy and sell orders by N_t^{buy} and N_t^{sell} respectively. We suppose that the order arrival process $(N_t^{\text{buy}}, N_t^{\text{sell}})$ is a bivariate Hawkes process.

In his paper, Hewlett shows that in our model of the FX market, market-makers set the prices at time t based on the observed flow till time t. This makes clear why we fit a Hawkes process to the data.

We fit a bivariate Hawkes process with exponential kernel to our data. The intensity functions take the form

$$\lambda_t^{*(i)} = \mu^{(i)} + \sum_{j=1}^{I} \sum_{k:t>T_k^{(j)}} \alpha_{ij} e^{-\beta_{ij}(t-T_k^{(j)})}.$$

In [16] a form of the intensity function with stochastic integral is shown, but this has been rewritten to an intensity function with a sum for our purposes.

In our model $j \in \{\text{buy}, \text{sell}\}$, thus we have a bivariate Hawkes process with the following two conditional intensity functions

$$\begin{aligned}\lambda_t^{*(\text{buy})} = \mu^{(\text{buy})} &+ \alpha_{\text{buybuy}} \sum_{k:t>T_k^{(\text{buy})}} e^{-\beta_{\text{buybuy}}(t-T_k^{(\text{buy})})} \\ &+ \alpha_{\text{buysell}} \sum_{l:t>T_l^{(\text{sell})}} e^{-\beta_{\text{buysell}}(t-T_l^{(\text{sell})})},\end{aligned}$$

$$\begin{aligned}\lambda_t^{*(\text{sell})} = \mu^{(\text{sell})} &+ \alpha_{\text{sellsell}} \sum_{l:t>T_l^{(\text{sell})}} e^{-\beta_{\text{sellsell}}(t-T_l^{(\text{sell})})} \\ &+ \alpha_{\text{sellbuy}} \sum_{k:t>T_k^{(\text{buy})}} e^{-\beta_{\text{sellbuy}}(t-T_k^{(\text{buy})})}.\end{aligned}$$

We see that there are ten parameters to be estimated. However, we can impose some symmetry constraints in order to reduce the number of parameters to be estimated and thus the complexity of the estimation. To reduce the number of parameters to be estimated, we say the mutual excitation and the self-excitation should be the same for both processes, as should be the background intensity ([31]):

- $\mu^{(\text{buy})} = \mu^{(\text{sell})} = \mu$,
- $\alpha_{\text{buybuy}} = \alpha_{\text{sellsell}} = \alpha_{\text{same}}$,
- $\alpha_{\text{sellbuy}} = \alpha_{\text{buysell}} = \alpha_{\text{cross}}$,
- $\beta_{\text{buybuy}} = \beta_{\text{sellsell}} = \beta_{\text{same}}$,
- $\beta_{\text{sellbuy}} = \beta_{\text{buysell}} = \beta_{\text{cross}}$.

The conditional intensity functions now simplify to

$$\begin{aligned}\lambda_t^{*(\text{buy})} = \mu &+ \alpha_{\text{same}} \sum_{k:t>T_k^{(\text{buy})}} e^{-\beta_{\text{same}}(t-T_k^{(\text{buy})})} \\ &+ \alpha_{\text{cross}} \sum_{l:t>T_l^{(\text{sell})}} e^{-\beta_{\text{cross}}(t-T_l^{(\text{sell})})},\end{aligned}$$

$$\begin{aligned}\lambda_t^{*(\text{sell})} &= \mu + \alpha_{\text{same}} \sum_{l:t>T_l^{(\text{sell})}} e^{-\beta_{\text{same}}(t-T_l^{(\text{sell})})} \\ &+ \alpha_{\text{cross}} \sum_{k:t>T_k^{(\text{buy})}} e^{-\beta_{\text{cross}}(t-T_k^{(\text{buy})})}.\end{aligned}$$

There are now only five parameters to be estimated.

8.3 The Results

In Hawkes paper, the model we have set up in the previous Section, is fitted on a real life dataset. The dataset used, are the market orders recorded on EBS (Electronic Broking Services is a wholesale electronic trading platform used to trade foreign exchange with market making banks) over two months for EUR/PLN.

As said in Section 8.2, we want to estimate values for the parameters ($\mu, \alpha_{\text{same}}, \alpha_{\text{cross}}, \beta_{\text{same}}, \beta_{\text{cross}}$). We fit our model to the data to trades in the week commencing 7th May 2005, aggregating simultaneous trades.

Estimating the parameters using maximum likelihood, subject to a non-negativity constraint, Hewlett finds the following values:

- $\hat{\mu} = 0.0033$,
- $\hat{\alpha}_{\text{same}} = 0.0169$,
- $\hat{\alpha}_{\text{cross}} = 0$,
- $\hat{\beta}_{\text{same}} = 0.0286$,
- $\hat{\beta}_{\text{cross}}$ is not identified.

The reason $\hat{\beta}_{\text{cross}}$ is not identified, is because $\hat{\alpha}_{\text{cross}} = 0$, thus looking at the conditional intensity function we see that in this case the term with β_{cross} is dropped, so its value can not be estimated.

The zero value of the α_{cross} seem to indicate that there is no influence of buy orders on sell orders, and conversely. ([31])

In Figure 8.1 a plot is shown in which both the estimated sell and buy intensities are shown. As said in Section 8.1, a feature of the FX market is the forming of clusters, which is something we observe in Figure 8.1. This was something to be expected, since the shown intensities follow a Hawkes process, and clusters are characteristic for Hawkes processes.

In Section 8.1 we explained why a Hawkes process is fitted to the data. However, we would like to also perform a reference fit, so we can check if the Hawkes process really gives us a better fit than another model. As a reference fit, Hewlett estimated a homogeneous Poisson process, so the conditional intensity functions described above reduces to $\lambda_t^{*(\text{buy})} = \lambda_t^{*(\text{sell})} = \mu$. Using maximum likelihood, Hewlett found $\hat{\mu} = 0.0080$.

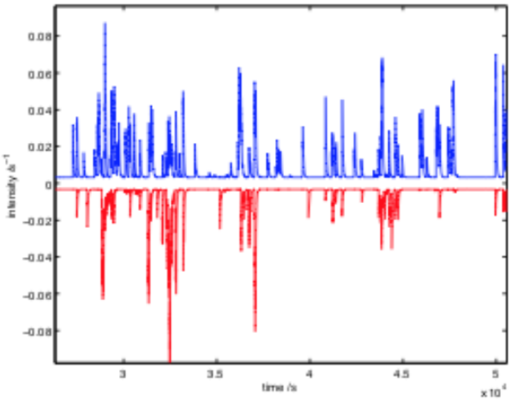


Figure 8.1: The sell and buy intensities corresponding to our fitted Hawkes process are shown in this graph. They are plotted over a period of about a day. On the x- axis the time per secondes is denoted and on the y-axis the itensity per second is denoted. In blue the buy intensities are shown and in red the sell intensities. The sell itensities are shown as negative only for the ease of interpretation. (Figure taken from [16], Figure 3)

We want to know if the Hawkes process provides a better fit compared to the Poisson process. To check this we are going to make a QQ plot for both estimated processes. In a quantile-quantile (QQ) plot we can check whether the data found corresponds to a given distribution. ([4]) In our case we have found a specific form of the Hawkes and homogeneous Poisson process based on our data. For intensity based point process models in general, the intensity-weighted waiting times $\int_{T_{n-1}}^{T_n} \left(\lambda_t^{*(\text{buy})} + \lambda_t^{*(\text{sell})} \right) dt$ between arrival times T_n of consecutive events have a standard exponential distribution. ([16]) What we can do in the QQ plot is plot the observed quantiles of our found Hawkes and homogeneous Poisson process, against the quantiles corresponding to the exponential distribution of the intensity-weighted waiting times. The better all points are on the $y = x$-line, the better the distributions are in comparison to each other. The QQ plot is shown in Figure 8.2.

We see that the Hawkes process follows the $y = x$-line very closely. For the homogeneous Poisson process, the points follow the $y = x$-line very badly, except perhaps in the upper tail. A possible reason why the point of the homogeneous Poisson process in the upper tail follow the $y = x$-line acceptable, is that larger waiting times tend to occur at times when the process is unexcited; at these times the behaviour of a self-exciting process is more Poissonian. ([16])

To conclude, we see that the Hawkes process is more suitable for describiding this model than the homogeneous Poisson process.

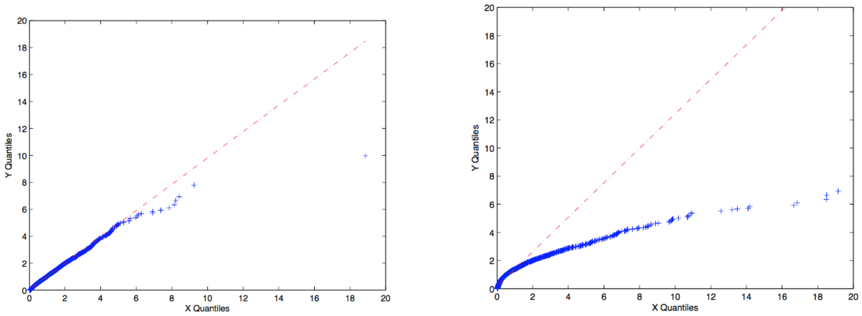


Figure 8.2: In both QQ plots the observed quantiles of the integrated intensities are plotted against the quantiles of the theoretical, exponential distribution. The QQ plot on the left shows the Hawkes process, whereas the QQ plot on the right shows the homogeneous Poisson process. (Figure taken from [16], Figure 2)

Chapter 9

Application in Social Media: Retweet Cascades

The coming of social media, has set up a whole new area to study. One may want to set up a model to determine the expected number of likes on an Instagram post or a model that can be used to determine the effect of ones post. In the setting up of this models, Hawkes processes proves to be very useful, since the self-exciting part makes it possible to use the information of the past, to better predict the future outcomes.

In this Chapter we will look at a model that can be used to predict the retweet cascade of a certain tweet.

This Chapter is based on Rizoiu, Lee, Mishra en Xie theirs 2017 paper *A Tutorial on Hawkes Processes for Events in Social Media* ([25]). We would like to stres that all results and figures are taken from this paper, so no own estimations are figures are made. This Chapter is mainly here to give a brief introduction to how Hawkes processes can be used to model real life situations.

9.1 Introduction

In Rizoiu et al. their paper, the word of mouth diffusion of online information on Twitter is modelled: one user posts a tweet to which other users can respond by either reacting or reposting, such that the tweet is broadcasted to even more users. In the model we will set up, we will only look at the number of retweets, which is the number of times a person his tweet is reshared by another person. So the reactions to ones tweet are not taken into account in our model. The retweets will be considered as events in a point process.

The information diffusion will be modeled as a self-exciting process. Three key intuitions of Twitter should be reflected in our model ([25]):

1. the magnitude of influence: tweets by users with many followers tend to get retweeted more;

2. memory over time: most retweeting occurs in fast response to the post;

3. content quality: the better/more influential/controversial the content of the tweet, the more retweets are to be expected.

The reason we model this information diffusion as a Hawkes process is because of a combination of the three key intuitions of Twitter. If a tweet is posted, other users get the new tweet in their timeline. The newer the tweet, the more likely people are to retweet the tweet (key intuition 2). By the retweeting even more people see the tweet, so more retweeting will occur. In this way certain "retweet clusters" are formed. These clusters are characteristic for Hawkes processes, which is why we expect that we can model the information diffusion as a Hawkes process. In Figure 9.1(a) we indeed observe the forming of certain clusters for the example we will look at in more detail in the following Sections.

9.2 The Model

The model proposed by Rizoiu et al. should be able to describe a retweet cascade, which is the set of retweets of an initial tweet, including the original tweet. This application, shows us the usefulness of the branching structure, which we have defined in Section 4.4, since a retweet cascade is something that can be easily seen in terms of the branching structure: the initial tweet is the immigrant event and all the retweets are its offspring. Since we are looking at the retweet cascade of a certain tweet, our model will consist of only one immigrant event, such that the background intensity equals zero: $\lambda_0 = 0$. Because we want to incorporate the magnitude of influence (key intuition 1), we use a marked Hawkes process, which we have seen before in Section 4.2.2 and Section 5.1: not all reposters have the same influence, so not all marks should be equal to each other. The conditional intensity function now looks like

$$\lambda^*(t) = \sum_{t>T_i} \mu_{m_i}(t - T_i).$$

For this application, we will construct a power law kernel:

$$\mu_m(t) = \frac{Km^\beta}{(c+t)^p}.$$

Let us look at the meaning of the different parameters ([25]):

- K describes the virality (or quality) of the tweet content, so it addreses key intuition 3;

- β affects the mark of a user, so it adresses key intuition 1, since it affects the magnitude of influence;

- p is the power law exponent (so $p > 1$), which addreses key intuition 2, by describing how fast an event is forgotten;

- c is just a temporal shift to keep $\mu_m(t)$ bounded when $t \approx 0$.

Overall, we see that Km^β describes the magnitude of influence and the content quality, whereas $(c+t)^p$ models the memory over time.

We could also have constructed the exponentional kernel $\mu_m(t) = Km^\beta pe^{-pt}$, but this kernel turns out to describes the model in a less precise way, as we will see in Section 9.4.

9.3 The Estimation Methods

9.3.1 Maximum-Likelihood Estimators

To model the retweet cascade, we need to obtain estimates for $\theta = \{K, \beta, c, p\}$. These estimates can be obtain by finding the maximum-likelihood estimators. Using Theorem 5, we can find that the log-likelihood function over the observation window $[0, T]$ is in this case given by

$$
\begin{aligned}
l(\theta) = \sum_{i=2}^{n} \ln(K) + \sum_{i=2}^{n} \ln\left(\sum_{t_j < t_i} \frac{(m_j)^\beta}{(t_i - t_j + c)^p}\right) \\
- K \sum_{i=1}^{n} (m_i)^\beta \left(\frac{1}{(p-1)c^{(p-1)}} - \frac{(T + c - t_i)^{-(p-1)}}{p-1}\right).
\end{aligned}
$$

The log-likelihood function found is clearly non-linear and should be maximized. For the model parameters there are some natural constraints: $p > 1$, $K > 0$, $c > 0$, and $\beta > 0$. For the branching factor, we are not interested in the case where $n^* > 1$. Although, $n^* > 1$ is mathematically valid, the super-critical phase (see Subection 4.4.3) is not realistic is terms of interpretation, since an infinite retweet cascade will not occur. Also the critical phase (see Subection 4.4.3) $n^* = 1$ is not realistic, since a large fluctuation in the expected number of retweets is not realistic. So instead, we will be looking at the sub-critical phase (see Subection 4.4.3), and we incorporate $n^* < 1$ as a non-linear constraint for the maximum-likelihood estimation.

9.3.2 Expected Number of Events

When a certain retweet cascade is observed until time T, we can simulate the continuation of the cascade by using the algorithms described in Chapter 7. Since, we are assuming we are in the sub-critical phase $n^* < 1$, our retweet cascade will die out after some time.

Instead of simulating some possible continuations, there is in this case a closed form solution for the expected number of events in a retweet cascade. ([25])

To compute the expected number of events in a retweet cascade, we have to look at three steps. First, we compute the expected size of a direct offspring to an event at time T_i after T. Then, we have to calculate the expected number of all offspring events. Finally, we put these two ideas together, to calculate the expected number of events in a retweet cascade. ([25])

We know that in the marked Hawkes process, each event $i = 1, ..., n$ that happened at $T_i < T$ adds $\mu_{m_i}(t - T_i)$ to the overall event intensity. So to find the expectation of A_i, which is the total number of events triggered by event $i = 1, ..., n$, we just have to integrate over the memory kernels of each event:

$$\begin{aligned} A_i &= \int_T^\infty \lambda^*(t)dt = \int_T^\infty \sum_{t>T_i} \mu_{m_i}(t - T_i)dt \\ &= \sum_{t>T_i} \int_T^\infty \mu_{m_i}(t - T_i)dt = K \sum_{i=1}^{n} \frac{m_i^\beta}{(1-p)(T + c - T_i)^{(1-p)}}. \end{aligned}$$

Rizoiu et al. state that the branching factor of the marked Hawkes process has the closed form

$$n^* = K \cdot \frac{\alpha - 1}{\alpha - \beta - 1} \cdot \frac{1}{(1-p)c^{(1-p)}},$$

for $\beta < \alpha - 1$, $p > 1$ and where the event marks m_i are identically and independently distributed according to $\mathbb{P}(m) = (\alpha - 1)m^{-\alpha}$. ([25])

The total number of events in a retweet cascade can now be obtained by combining what we found. Each expected event in A_1 is expected to generate n^* direct offspring events, n^{*2} “grand-offspring” events, ..., n^{*k} kth generation offspring events, thus for the expected number of events in a retweet cascade N_∞ we find

$$N_\infty = n + A_1(1 + n^* + n^{*2} + ...) = n + \frac{K}{1-n^*}\left(\sum_{i=1}^{n} \frac{m_i^\beta}{(1-p)(T + c - T_i)^{(1-p)}}\right),$$

where $n^* < 1$. ([25])

9.4 Results

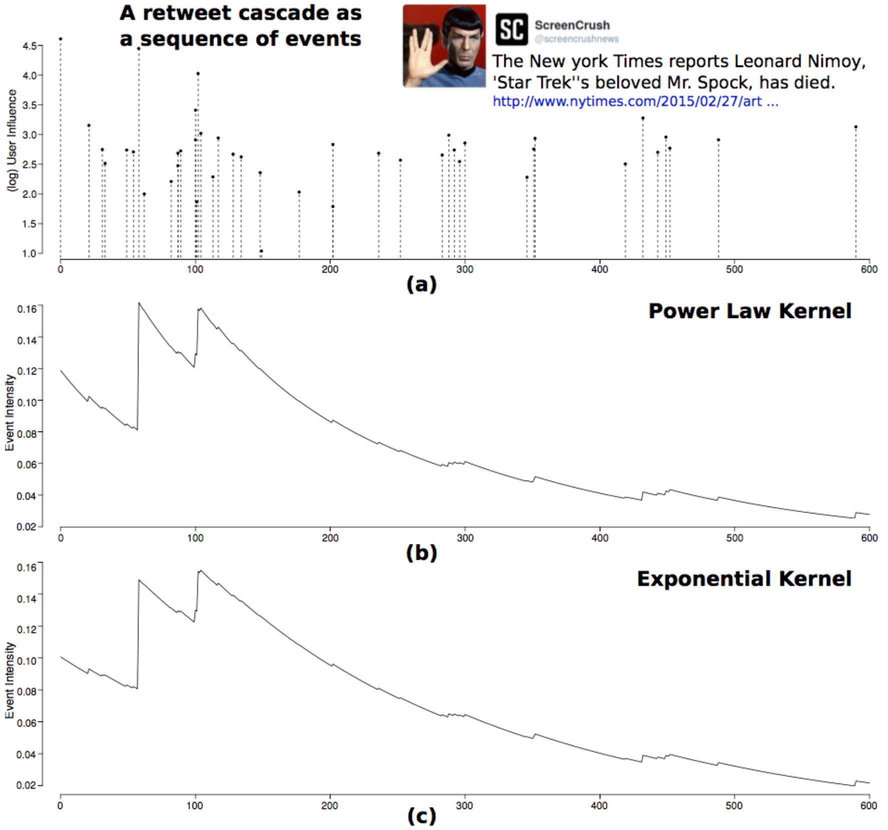
The descriptive methods above will be tested to some real life example. The retweet cascade of a New York Times article about the dead of Leonard Nimoy is shown in Figure 9.1(a). This retweet cascade will be analysed.

Before we are able to predict the expected number of events in the retweet cascade, we first have to find estimates for our parameters in the Hawkes process with power law kernel. We are observing the retweet cascade over the interval $[0, 10]$, where the time is in minutes. The parameters are estimated as the maximum-likelihood estimates, which we have looked at in Subsection 9.3.1. Riziou et al. reports the following values for the estimates: $\hat{\theta} = \{\hat{K} = 1.00, \hat{\beta} = 1.01, \hat{c} = 250.65, \hat{p} = 2.33\}$. In Figure 9.1(b) the fitted Hawkes intensity function is shown. Figures 9.1(a) and 9.1(b) are temporally aligned in the way that each occured event causes the conditional intensity function to increase. The big value of K tells us this article has high content virality, and the big value of c

tells us this article has a large waiting time. These two effects lead to a slow diffusion of the article.

We also want to find the parameters we would have in the case of an exponential kernel, so we can compare both kernels. Using maximum-likelihood estimates, Rizoiu et al. reports the following parameters for the exponential kernel: $\hat{\theta} = \{\hat{K} = 0.0003, \hat{\beta} = 1.0156, \hat{p} = 1.0054\}$. In Figure 9.1(c) the fitted Hawkes intensity function is shown. Figures 9.1(a) and 9.1(c) are temporally aligned in the way that each occured event causes the conditional intensity function to increase.

Based on the found estimates of the parameters we find a branching factor of $n^* = 0.92$ for the power law kernel and a branching factor of $n^* = 0.997$ for the exponentional kernel. Using the result of Subection 9.3.2 we find for the power law kernel an expected number of 216 events in the retweet cascade, whereas we find an expected number of 1603 events in the retweet cascade of the exponential kernel. The real number of retweets observed was 216, so it may be clear that using the power law kernel is indeed more appropriate for this situation than using an exponential kernel.



ScreenCrush
@screencrushnews
The New york Times reports Leonard Nimoy, 'Star Trek''s beloved Mr. Spock, has died.
http://www.nytimes.com/2015/02/27/art ...

Figure 9.1: An example retweet cascade on a news article by The New York Times. (a) The number of retweets for the first 10 minutes is shown. Each retweet corresponds to a new event. The height of the event, corresponds to the influence of the user who retweeted the article. (b) The conditional intensity function for the fitted Hawkes process with power law kernel is shown. (c) The conditional intensity function for the fitted Hawkes process with exponential kernel is shown. (Figure taken from [25], Figure 1.3)

Bibliography

[1] K.B. Athreya and S.N. Lahiri. 2006. *Measure Theory and Probability Theory*. Springer New York.

[2] E. Bacry, I. Mastromatteo and J.-F. Muzy. 2015. *Hawkes Processes in Finance*. Market Microstructure and Liquidity, 1(1) 1550005.

[3] C.G. Bowsher. 2007. *Modelling Security Market Events in Continuous Time: Intensity Based, Multivariate Point Process Models*. Nuffield Economics Working Paper. Retrieved from: `www.nuff.ox.ac.uk/economics/papers/2002/w22/NuffWPf15.pdf` (accessed June 8 2019.)

[4] C. Chapman and E. McDonell Feit. 2019. *R for Marketing Research and Analytics* , 2nd. Springer Nature Switzerland.

[5] Y. Chen. 2016. *Multivariate Hawkes Processes and Their Simulations*. Available online at: `https://www.math.fsu.edu/ ychen/research/multiHawkes.pdf` (accessed May 31 2019.)

[6] Y. Chen. 2016. *Thinning Algorithm for Simulating Point Processes*. Available online at: `https://www.math.fsu.edu/ ychen/.../Thinning%20algorithm.pdf` (accessed May 31 2019.)

[7] D. Coculescu and A. Nikeghbali. 2007. *Filtrations*. Preprint arXiv:0712.0622.

[8] S.N. Cohen and R.J. Elliott. 2015. *Stochastic Calculus and Applications*, 2nd. Birkhäuser New York.

[9] D.L. Cohn. 2013. *Measure Theory*, 2nd. Birkhäuser New York.

[10] D.J. Daley and D. Vere-Jones. 2003. *An Introduction to the Theory of Point Processes Volume I: Elementary Theory and Methods*, 2nd. Springer New York.

[11] A. Dassios and H. Zhao. 2013. *Exact Simulation of Hawkes Process with Exponentially Decaying Intensity*. Electronic Communications in Probability, 18(62): 1-13.

[12] F.M. Dekking, C. Kraaikamp, H.P. Lopuhaä and L.E. Meester. 2005. *A Modern Introduction to Probability and Statistics*. Springer-Verlag London.

[13] P.H. Franses, D. van Dijk and A. Opschoor. 2014. *Time Series Models for Business and Economic Forecasting*, 2nd. Cambridge University Press Cambridge.

[14] A.G. Hawkes. 1971. *Spectra of Some Self-Exciting and Mutually Exciting Point Processes*. Biometrika, 58(1): 83-90.

[15] A.G. Hawkes and D. Oakes. 1974. *A Cluster Process Representation of a Self-Exciting Process*. Journal of Applied Probability, 11(3): 493-503.

[16] P. Hewlett. 2006. *Clustering of Order Arrivals, Price Impact and Trade Cluster Optimisation*. Workshop on Financial Modeling with Jump Processes, Ecole Polytechnique.

[17] D.H. Johnson. 1996. *Point Process Models of Single-Neuron Discharges*. Journal of Computational Neuroscience, 3(4): 275-299.

[18] D. Kane, A. Liu and K. Nguyen. 2011. *Analizing an Electronic Limit Order Book*. The R Journal, 3(1): 64-68.

[19] J.F.C. Kingsman and S.J. Taylor. 1966. *Introduction to Measure and Probability*. Cambridge University Press Cambridge.

[20] G. Last and M. Penrose. 2017. *Lectures on the Poisson Process*. Cambridge University Press Cambridge.

[21] P.J. Laub, T. Taimre and P.K. Pollett. 2015. *Hawkes Processes*. Preprint arXiv:1507.02822.

[22] Y. Ogata. 1981. *On Lewis' Simulation Method for Point Processes*. IEEE Transactions on Information Theory, 27(1): 23-31.

[23] Y. Ogata. 1988. *Statistical Models for Earthquake Occurrences and Residual Analysis for Point Processes*. Journal of the American Statistical Association, 83(401): 9-27.

[24] T. Ozaki. 1979. *Maximum Likelihood Estimation of Hawkes' Self-Exciting Point Processes*. Annals of the Institute of Statistical Mathematics, 31(1): 145–155.

[25] M.-A. Rizoiu, Y. Lee, S. Mishra and L. Xie. 2017. *A Tutorial on Hawkes Processes for Events in Social Media*. Preprint arXiv:1708.06401.

[26] S.M. Ross. 2014. *Introduction to Probability Models*, 11nd. Elsevier Boston.

[27] F.R. Schoenberg. 1997. *Assessment of Multi-dimensional Point Process Models*, Doctoral dissertation. Retrieved from: `http://www.stat.ucla.edu/ frederic/papers/Rickthesis1997.pdf` (accessed June 8 2019.)

[28] J. Sharpe. 2012. *Foreign Exchange: The Complete Deal* . Harriman House Petersfield.

[29] A.H. Studenmund. 2017. *A Practical Quide to Using Econometrics*, 7th. Pearson Education Limited Harlow.

[30] A. Swishchuk. 2017. *Hawkes Processes and their Applications in Finance and Insurance.* Talk from the Department of Mathematics and Statistics of the University of Calgary. Available online at: `people.ucalgary.ca/ aswish/Swishchuk_LaLTalk.pdf` (accessed June 10 2019.)

[31] I.M. Toke. 2011. *An Introduction to Hawkes Processes with Applications to Finance.* Lectures Notes from Ecole Centrale Paris, BNP Paribas Chair of Quantitative Finance. Available online at: `www.smallake.kr/wp-content/uploads/2015/01/HawkesCourseSlides.pdf` (accessed May 31 2019.)

Appendices

Appendix A

Concepts of Measure Theory

Stochastic processes can be defined in many different ways. In this Thesis the definitions are rather concrete: many technical difficulties are not specified. However, stochastic processes can be defined in a very theoretical and precise way. To define stochastic processes very precisely measure theory is needed. In this Appendix we give a short introduction to the measure theoretical concepts which are important for the theory of stochastic processes. This Appendix follows Chapter 1 and Chapter 2 from [9]. For a more detailed measure theoretical overview the reader is encouraged to look at [9].

A.1 Sigma-Algebras

The first thing we have to define are sigma-algebras. The reason we need sigma-algebras is because they are used in the definition of measures, which we'll deal with in Section A.2, and these measures are important in probability theory, which we'll deal with in Appendix B.

Defintion A1 ([9], in text, page 1): Let X be an arbitrary set. A collection $\mathcal{A}$ of subsets of X is a σ-algebra (or σ-field) on X if

1. $X \in \mathcal{A}$,
2. for each set A that belongs to $\mathcal{A}$, the set A^C belongs to $\mathcal{A}$,
3. for each infinite sequence $\{A_i\}$ of sets that belong to $\mathcal{A}$, the set $\cup_{i=1}^{\infty} A_i$ belongs to $\mathcal{A}$,
4. for each infinite sequence $\{A_i\}$ of sets that belong to $\mathcal{A}$, the set $\cap_{i=1}^{\infty} A_i$ belongs to $\mathcal{A}$,

Because the definition of a sigma-algebra is rather abstract, we give an example of a specific sigma-algeba.

Example A1: Let $X = \{1, 2, 3\}$ and $\mathcal{A} = \{\emptyset, \{1, 2, 3\}\}$. Obviously $\{1, 2, 3\} \in \mathcal{A}$, thus the first condition is fulfilled. We have $\emptyset^C = \{1, 2, 3\} \in \mathcal{A}$ and $\{1, 2, 3\}^C = \emptyset \in \mathcal{A}$, thus the second condition is fulfilled. Also $\emptyset \cup \{1, 2, 3\} = \{1, 2, 3\} \in \mathcal{A}$ and $\emptyset \cap \{1, 2, 3\} = \emptyset \in \mathcal{A}$, thus both the third and fourth condition are also fulfilled. So we see that $\mathcal{A}$ is a σ-algebra on X.

Although the Definition of the sigma-algebra as given in Deinition A1 is absolutely correct, it can be rewritten in a shorter form. ([9])

Let us first note that a σ-algebra on A can be seen as a family of subsets of X that contains X (condition 1), is closed under complementation (condition 2), is closed under the formation of countable unions (condition 3), and is closed under the formation of countable intersections (condition 4).

The thing about Definition A1 is that the fourth condition of the definition is fulfilled as soon as the third condition is fulfilled (and the other way around). From set theory we know that $\cap_{i=1}^{n} A_i = \left(\cup_{i=1}^{n} A_i{}^C\right)^C$. Let us now assume that the first three conditions are true, so we know that for each infinite sequence $\{A_i\} \in \mathcal{A}$ it is true that $\cup_{i=1}^{\infty} A_i \in \mathcal{A}$. Since condition two is true as well, we also have $\{A_i\}^C \in \mathcal{A}$, thus $\cup_{i=1}^{\infty} A_i{}^C \in \mathcal{A}$. Making use of condition two again we now also have $\left(\cup_{i=1}^{\infty} A_i{}^C\right)^C = \cap_{i=1}^{n} A_i \in \mathcal{A}$, thus the first three condition imply the fourth condition. Thus the fourth condition in Definition A1 is not necessary to check as soon as the second and third condition are fulfilled.

If condition one and two are both fulfilled, we know that $X^C = \emptyset \in \mathcal{A}$, thus condition one can also be written as $\emptyset \in \mathcal{A}$.

There is even another way to rewrite condition one. If $\mathcal{A}$ is a family of subsets of X that is nonempty, closed under complementation, and closed under the formation of finite or countable unions, then $\mathcal{A}$ must contain X: if $A \in \mathcal{A}$, then X, which is the union of A and A^C, must also belong to $\mathcal{A}$. So we can also replace condition one by the condition that $\mathcal{A}$ is nonempty.

A.2 Sub-Sigma-Algebras

Like we have a subset of a set, we also can have a sub-σ-algebra of a σ-algebra.

Defintion A2: A sub-σ-algebra of a σ-algebra is a subset of a σ-algebra such that the sub-σ-algebra still fulfills all the conditions of the σ-algebra.

A.3 Measures

Now that we have defined sigma-algebras we are able to define measures.

Defintion A3 ([9], in text, page 7): Let X be an arbitrary set and let $\mathcal{A}$ be a σ-algebra on X. A measure on $\mathcal{A}$ is a function $\mu : \mathcal{A} \to [0, \infty)$ that satisfies

1. $\mu(\emptyset) = 0$,
2. for each infinite sequence $\{A_i\}$ of disjoint sets that belong to $\mathcal{A}$ we have $\mu(\cup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \mu(A_i)$.

As was the case with the definition of the sigma-algebra, also the definition of a measure is rather abstract, so we'll give a concrete example of a measure.

Example A3: Let $X = \{1, 2, 3\}$ and $\mathcal{A} = \{\emptyset, \{1, 2, 3\}\}$. As we have seen in Example A1, $\mathcal{A}$ is a σ-algebra on X. Let us now define the function $\mu : \mathcal{A} \to [0, \infty)$ as

$$\mu(A) = \begin{cases} 0 & \text{if } A = \emptyset, \\ 1 & \text{if } A = \{1, 2, 3\}, \end{cases}$$

We clearly see $\mu(\emptyset) = 0$, thus the first condition for being a measure is fulfilled. Since $1 = \mu(\{1, 2, 3\}) = \mu(\emptyset \cup \{1, 2, 3\}) = \mu(\emptyset) + \mu(\{1, 2, 3\}) = 0 + 1 = 1$ also the second condition of being a measure is fulfilled. So μ is a measure on $\mathcal{A}$.

To be more precise we could call the measure as defined in Definition A3 a countably additive measure, based on the fact that condition two says that the function must be countably additive. Compared to countably additive, a function could also be finitely additive. A function is finitely additive if

$$\mu(\cup_{i=1}^{n} A_i) = \sum_{i=1}^{n} \mu(A_i)$$

for each infinite sequence $\{A_i\}$ of disjoint sets that belong to $\mathcal{A}$. ([9])

A.4 Measure Space

Now that we have defined the concepts of a sigma-algebra $\mathcal{A}$ on an arbitrary set X and a measure μ on $\mathcal{A}$, we can define a measure space.

Defintion A4 ([9], in text, page 8): If X is a set, if $\mathcal{A}$ is a σ-algebra on X, and if μ is a measure on $\mathcal{A}$, then the triplet $(X, \mathcal{A}, \mu)$ is called a measure space.

Based on Definition A4 a measure space is nothing more than a triplet that contains certain information. The X tells us at which set we are looking, the $\mathcal{A}$ tells us which subsets of X form a σ-algebra and are thus suitable for measuring, and the μ tells us which measure we will use.

A.5 Measurable Space

Related to the concept of a measure space, is the concept of a measurable space.

Defintion A5 ([9], in text, page 8): If X is a set and if $\mathcal{A}$ is a σ-algebra on X, then the pair $(X, \mathcal{A})$ is called a measurable space.

As the name suggests, a measurable space is a space that contains information about what sets will be measured.

If we know that $(X, \mathcal{A}, \mu)$ is a measure space, then we can say that μ is a measure on the measurable space $(X, \mathcal{A})$. ([9])

A.6 Finite and σ-Finite Measures

Now we have defined the basic concepts of measures, we'll look at two different kind of measures: finite measures and σ-finite measures.

Defintion A5 ([9], in text, page 9): Let μ be a measure on a measurable space $(X, \mathcal{A})$. Then μ is a finite measure if $\mu(X) < \infty$.

A finite measure is thus simply a measure that always takes on finite values. An example of a finite measure, is the measure we have defined in Example A3.

Defintion A6 ([9], in text, page 9): Let μ be a measure on a measurable space $(X, \mathcal{A})$. Then μ is a σ-finite measure if X is the union of a sequence $A_1, A_2, ...$ of sets that belong to $\mathcal{A}$ and satisfy $\mu(A_i) < \infty$ for each i.

We call a measure space $(X, \mathcal{A}, \mu)$ finite or σ-finite if the μ is finite or respectively σ-finite. ([9])

A.7 Measurable Functions

Now that we know what measure spaces are, we can look at a function between two measure spaces.

Defintion A7 ([9], in text, page 73): Let $(X, \mathcal{A})$ and $(Y, \mathcal{B})$ be measurable spaces. A function $f : X \to Y$ is measurable with respect to $\mathcal{A}$ and $\mathcal{B}$ if for each B in $\mathcal{B}$ the set $f^{-1}(B)$ belongs to $\mathcal{B}$.

Appendix B

Concepts of Probability Theory

Appendix B builds upon the knowledge of Appendix A to give an overwiew of the probability theory needed to fully understand stochastic processes. This Appendix follows Chapter 2 from [12], Chapter 10 from [9] and Chapters 10 and 11 from [19], but also builds upon the knowledge of introductory probability theory courses, so the Definitions sometimes use different words/symbols.

B.1 Sample Space

One of the first concepts that's defined in every introductory probability theory course, is the concept of a sample space. So let us also start by defining a sample space.

> **Defintion B1 ([12], in text, page 13):** The set of all possible outcomes of an experiment is known as the sample space of an experiment.

In this Thesis, the sample space will be denoted by Ω.

Besides the concept of a sample space, also the concept of an event is important in probability theory.

> **Defintion B2 ([12], in text, page 14):** An event E is a subset of the sample space.

Although the concept of a sample space and an event is almost certainly known to the reader, we will give an Example of the sample space and an event for a certain experiment. This Example is more a build-up to the more difficult concepts defined in the following sections.

Example B1: A fair coin is tossed twice. The fact that the coin is fair, means that the probability of tossing heads is equal to the probability of tossing tails (both have a probability of 50% of occuring). The sample space of the experiment is obviously $\Omega = \{HH, HT, TH, TT\}$, where H denotes heads and T denotes tails. The subset $E = \{HH, HT, TH\}$ of the sample space Ω is an event; this event denotes all the tosses in which at least one of the tosses results in heads.

B.2 Probability Measure

A probability measure is, as the name suggests, a measure with some additional conditions.

Defintion B3 ([19], in text, page 269): Let X be an arbitrary set and let $\mathcal{A}$ be a σ-algebra on X. A probability measure on $\mathcal{A}$ is a function $\mathbb{P} : \mathcal{A} \to [0, 1]$ that satisfies

1. $\mathbb{P}(\emptyset) = 0$,
2. $\mathbb{P}(X) = 1$,
3. for each infinite sequence $\{A_i\}$ of disjoint sets that belong to $\mathcal{A}$ we have $\mathbb{P}(\cup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \mathbb{P}(A_i)$.

It may be clear that the first two conditions satisfy that every probability measure is finite.

In the scope of probability theory the set X is mostly seen as the sample space Ω of an experiment. We know that $\mathcal{A}$ is a collection of subsets of X (thus now Ω), so $\mathcal{A}$ consists of events of the experiment. So the probability measure assigns a certain probability to every event of the experiment.

Looking at a probability measure in the light of probability theory makes all the new conditions as described in Definition B3 rather logical:

- a probability is always between zero and one, so the probability measure is a function that always results in a value between zero and one,
- an experiment always has an outcome, so the probability measure assigns a value of zero to the empty set, since always something occurs,
- one of the elements in the sample space always occurs, so the probability measure assigns a value of one to the whole sample space.

Also the additive condition is rather logical in the light of probability theory. To understand this we extend our example from the previous section.

Example B2: Let us first define a σ-algebra $\mathcal{A}$ on the sample space Ω. Since a σ-algebra is a collection of subsets of Ω, a natural choice for $\mathcal{A}$ is the power set $\mathcal{P}(\Omega)$; the power set of Ω is the set that contains all the subsets of Ω. So we'll choose $\mathcal{A} = \mathcal{P}(\Omega) = \{\emptyset, \{HH\}, \{HT\}, \{TH\}, \{TT\}, \{HH, HT\}, \{HH, TH\}, \{HH, TT\}, \{HT, TH\}, \{HT, TT\}, \{TH, TT\}, \{HH, HT, TH\}, \{HH, HT, TT\}, \{HH, TH, TT\}, \{HT, TH, TT\}, \{HH, HT, TH, TT\}\}$. Since we are throwing a fair dice all the elements in the sample space have a equal probability to occur, so we know that $\mathbb{P}(HH) = \mathbb{P}(HT) = \mathbb{P}(TH) = \mathbb{P}(TT) = \frac{1}{4}$. If we, for example, want to know $\mathbb{P}(HH, HT)$, it's best to look at what $\mathbb{P}(HH, HT)$ means: it is the probability of throwing heads first regardless of what's thrown after. Since we are having a fair dice, the probability of throwing heads the first round is $\frac{1}{2}$, thus $\mathbb{P}(HH, HT) = \frac{1}{2}$. We now see $\mathbb{P}(HH, HT) = \mathbb{P}(HH) + \mathbb{P}(HT)$, which is condition three of a probability measure. It may now be clear to the reader that $\mathbb{P} : \mathcal{A} \to [0, 1]$, where

$$\mathbb{P}(A) = \begin{cases} 0 & \text{if } A = \emptyset, \\ 1/4 & \text{if } A = \{HH\}, \{HT\}, \{TH\}, \{TT\}, \\ 1/2 & \text{if } A = \{HH, HT\}, \{HH, TH\}, \{HH, TT\}, \{HT, TH\}, \\ & \{HT, TT\}, \{TH, TT\} \\ 3/4 & \text{if } A = \{HH, HT, TH\}, \{HH, HT, TT\}, \{HH, TH, TT\}, \\ & \{HT, TH, TT\}, \\ 1 & \text{if } A = \{HH, HT, TH, TT\}, \end{cases}$$

is a probability measure on $\mathcal{A}$.

A word of caution is given to the special case $\mathbb{P}(E) = 1$, where $E \in \mathcal{A}$. An event E for which it's the case that $\mathbb{P}(E) = 1$ is said to occur almost surely or almost certainly. The word 'almost' is a reminder of the fact that $\mathbb{P}(E) = 1$, but this does not have to imply that $E = \Omega$. (Iy may be clear that in Example B2 there isn't an event, besides the whole sample space, which occurs almost surely.)

B.3 Probability Space

As was the case for a measure space, the definition of a probability space is fairly obvious.

Defintion B4 ([9], in text, page 307): If Ω is a set, if $\mathcal{A}$ is a σ-algebra on Ω, and if $\mathbb{P}$ is a measure on $\mathcal{A}$, then the triplet $(\Omega, \mathcal{A}, \mathbb{P})$ is called a probability space.

So a probability space just tells us what sample space our experiment has, what events we are considering, and how we can assign a certain probability to a particular event.

B.4 Random variable

Instead of looking at the probability that a certain event, say $E = \{HH, HT\}$, occurs, we can also look at the number of times heads is thrown. Define X to be the number of times heads is thrown. Throwing a fair dice twice, shows $X \in \{0, 1, 2\}$. The quantity X is what we'll call a random variable.

Defintion B5 ([19], in text, page 285): A random variable X is a measurable function from a probability space $(\Omega, \mathcal{A}, \mathbb{P})$ to a measurable space $(S, \mathcal{B})$.

The definition of a random variable X can be heuristically 'determined' when looking at a concrete example.

Example B3: Again, we look at the case of throwing a fair dice twice. Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space as defined in Example B1. Let X be a random variable that represents the number of times heads is thrown, thus $X \in \{0, 1, 2\}$, so X is a function from Ω to $\{0, 1, 2\}$ defined in the following way:

$$X(A) = \begin{cases} 0 & \text{if } A = TT, \\ 1 & \text{if } A = TH, HT, \\ 2 & \text{if } A = HH \end{cases}$$

Generally, X cannot be any function, since in the end we want to talk about such things a 'the probability that X lies in the interval (a, b), or $\mathbb{P}(A | a \leq X(A) \leq b)$. As an example: $\mathbb{P}(A | 0 \leq X(A) \leq 1) = \frac{3}{4}$, since X is between 0 and 1 for the cases $\{TT, TH, HT\}$ and $\mathbb{P}(\{TT, TH, HT\}) = \frac{3}{4}$ as seen in Example B2. So we demand that the sets of the form $\{A : a \leq X(A) \leq b\}$ should belong to $\mathcal{B}$, which is exactly the same as saying that X is a measurable function from $(\Omega, \mathcal{A}, \mathbb{P})$ to $(\{0, 1, 2\}, \mathcal{B})$.

Appendix C

Mathematica Code For Graphs

The Mathematica codes that have been used to produce the graphs in Chapter 4 are shown in this Appendix.

```
(* First we define the function for the expected number of ...
    events *)
EXP[\[Alpha]_, \[Beta]_, \[Lambda]_,
T_] := \[Lambda]*
T + (\[Lambda]*\[Alpha])/(\[Alpha] - \[Beta])^2*(Exp[(\[Alpha] - \
\[Beta])*T] - 1 - (\[Alpha] - \[Beta])*T)

(* We make a graph of the case where \[Alpha] is variable *)
Plot[
EXP[\[Alpha], 2, 2, 2], {\[Alpha], 0, 4},
GridLines -> {{{2, Directive[Red, Thick]}}, None},
ImageSize -> Large, PlotRange -> {{0, 4}, {0, 120}},
PlotStyle -> {Thickness[0.005]}, AxesStyle -> Directive[Black, 14],
AxesLabel -> {"\[Alpha]", "E[N(2)]"},
PlotLabel ->
Style["Expected number of events as a function of \[Alpha]",
FontSize -> 23, Directive[Black, Bold]]]

(* We make a graph of the case where \[Beta] is variable *)
Plot[
EXP[2, \[Beta], 2, 2], {\[Beta], 0, 4},
GridLines -> {{{2, Directive[Red, Thick]}}, None},
ImageSize -> Large, PlotRange -> {{0, 4}, {0, 120}},
PlotStyle -> {Thickness[0.005]}, AxesStyle -> Directive[Black, 14],
AxesLabel -> {"\[Beta]", "E[N(2)]"},
PlotLabel ->
Style["Expected number of events as a function of \[Beta]",
FontSize -> 23, Directive[Black, Bold]]]

(* We make a graph of the case where Subscript[\[Lambda], 0] is \
variable *)
```

```
Plot[EXP[1, 2, \[Lambda], 2], {\[Lambda], 0, 4},
ImageSize -> Large, PlotStyle -> {Thickness[0.005]},
AxesStyle -> Directive[Black, 14],
AxesLabel -> {"\!\(\*SubscriptBox[\(\[Lambda]\), \(0\)]\)",
"E[N(2)]"},
PlotLabel ->
Style["Expected number of events as a function of \
\!\(\*SubscriptBox[\(\[Lambda]\), \(0\)]\)", FontSize -> 23,
Directive[Black, Bold]]]

(* We make a graph of the case where t is variable *)
Plot[
EXP[1, 2, 1, t], {t, 0, 4}, ImageSize -> Large,
PlotStyle -> {Thickness[0.005]}, AxesStyle -> Directive[Black, 14],
AxesLabel -> {"t", "E[N(t)]"},
PlotLabel ->
Style["Expected number of events as a function of t",
FontSize -> 23, Directive[Black, Bold]]]
```

Appendix D

Matlab Codes Used for the Simulations

The Matlab codes that have been used to produce the graphs of the simulations are shown in this Appendix.

D.1 Code to Simulate the Homogeneous Poisson Process

```matlab
e = 1; % We start the counter to run the simulation 10000 times
while e < 10001 % As long as there haven't been 10000 ...
    simulations, a new simulation will take place

% The input variables:
T = 10; % We simulate over [0,T]
lambda = 4; % The arrival rate

% Initialization
n = 1; % Counter for the Poisson process (Matlab can't start at 0)
setT(1) = - log(rand)/lambda; % First event time

% The while-loop to simulate the process
while setT(n) < T % As long as the last event time is smaller ...
    than T, we generate new points
setT(n+1) = setT(n) - log(rand)/lambda; % We calculate the new ...
    event time
n = n + 1; % The counter, i.e. the number of events occured, is ...
    raised by one
end

% Check if the last point in the process occured after T: if ...
    this happened, the point is removed and one is subtracted ...
    from the counter
if setT(n) > T
```

```
setT(n)=[];
n=n-1;
end

% Create a set in which a zero is added before and a T is added ...
    after the event times
newsetT=[0 setT T];

% Code for the repeated simulation:
M(e) = n; % For every round of the simulation, the number of ...
    events in the Poisson process will be saved
e = e + 1; % The counter for the number of simulations will be ...
    raised
end

% Make a graph
for i = 0 : 1 : size(newsetT,2)-2 % So many lines will be drawn
hold on % All lines should be seen in one graph
plot([newsetT(i+1) newsetT(i+2)], [i i],'-b'); % Line lies ...
    between consecutive points in newsetT and has the same y value
end
scatter(setT, 1:n, 80, '.b'); % Show all the events in the graph
xlim([0 T]) % Axis value for the x axis
ylim([0 n+1]) % Axis value for the y axis
xlabel('Time t','FontSize',12)
ylabel('Number of events','FontSize',12)
title('Simulation of a Poisson process with rate 4', 'FontSize', 16)

% Code to make the histogram
histogram(M,'Normalization','probability', 'DisplayName', ...
    'Number of events at T')
xlabel('Number of events','FontSize',12)
ylabel('Probability density','FontSize',12)
title('Number of event times for \lambda = 4','FontSize',16)

% Code to obtain parameters for histogram plot with pdf
parameters = fitdist(M.', 'poisson') % Fit data according to a ...
    Poisson distribution

% Code to plot histogram, fitted distribution and actual ...
    distribution in one graph
hold on % Make sure everything ends up in one graph
x = min(M)-2 : 1 : max(M)+2; % Specify part over which we plot ...
    the distributions
fitpois = pdf(parameters, x); % Calculate density for our ...
    distribution for the right values
plot(x, fitpois, 'DisplayName','Fitted ...
    distribution','LineWidth',2) % Plot fitted distribution in ...
    histogram
actualpois = pdf('poisson', x, lambda*T); % Calculate density ...
    for the true distribution for the right values
plot(x, actualpois, 'DisplayName','Actual ...
    distribution','LineWidth',2) % Plot actual distribution in ...
    histogram
legend % Create a legend
hold off
```

D.2 Code to Simulate the Nonhomogeneous Poisson Process

```
e = 1; % We start the counter to run the simulation 10000 times
while e < 10001 % As long as there haven't been 10000 ...
    simulations, a new simulation will take place

% The input variable:
T = 10; % We simulate over [0,T]

% Determine the sup for the intensity function over [0,T]:
xmin = 0;  xmax = T; % Interval on which we would like to know sup
stepsize = 0.01; % Step size: we can't evaluate all points, so ...
    stepsize must be small enough to be precise, but not too ...
    small to make the simulation run too slow
x = xmin : stepsize : xmax; % Create array: for these points we ...
    calculate the intensity function
y = 1 + sin(x); % The intensity function
[val, idx] = max(y); % Determine the maximum/supremum
lambda = y(idx); % Set our lambda equal to the found supremum

% Initialization:
n = 1; % Counter for the nonhomogeneous Poisson process (Matlab ...
    can't start at 0)
m = 1; % Counter for the homogeneous Poisson process (Matlab ...
    can't start at 0)
s = 0; % First "event" time for the homogeneous Poisson process
t = 0; % First "event" time for the nonhomogeneous Poisson process

% Define the intensity function
intensity = @(a) 1 + sin(a); % make sure it's equal to the ...
    intensity function above

% The while-loop:
while s(m) < T % As long as the last event time of the ...
    homogeneous Poisson is smaller than T, we generate new points
s(m+1) = s(m) - log(rand)/lambda; % Generate the new event time
D(m+1) = rand; % Generate a random variable
if D(m+1) ≤ intensity(s(m+1))/lambda % Determine if homogeneous ...
    point will end up in the nonhomogeneous Poisson process
t(n+1) = s(m+1); % The homogeneous point will be in the ...
    nonhomogeneous process
n = n + 1; % Counter of the nonhomogeneous process will be raised
end
m = m + 1; % Counter of the homogeneous process will be raised
end

% ?Remove first points from s, t and D, since nothing occured at ...
    T = 0, but Matlab made us start at 0. The counters are also ...
    lowered, since points are removed:
s(1) = [];
m = m - 1;
t(1)=[];
n = n - 1;
```

```
D(1) = [];

% ?Check if the last point in the processes occured after T: if ...
    this happened,?the point is removed and one is subtracted ...
    from the counter
if s(m) > T
s(m) = [];
D(m) = []; % ?In this case the last value of D isn't used
m = m - 1;
end
if t(n) > T
t(n) = [];
n = n - 1;
end

% Code for the repeated simulation:
M(e) = n; % For every round of the simulation, the number of ...
    events in the nonhomogeneous Poisson process will be saved
e = e + 1; % The counter for the number of simulations will be ...
    raised
end

% Make an array of the points which are in the homogeneous ...
    Poisson process but not in the nonhomogeneous Poisson process
srem = setdiff(s,t);

% Make the right values of D correspond to the right values of t ...
    and srem
o = 1; % Start a counter for the Dt list
p = 1; % Start a counter for the Dsrem list
for l = 1 : 1 : size(s, 2) % We check it for all the elements in s
if ismember(s(l),t) % If an element in s also belongs to t
Dt(o)=D(l); % we add the value of D on place l to Dt
o = o + 1; % we raise the counter to make sure t corresponds to ...
    the right Dt value
end
end
for q = 1 : 1 : size(s, 2) % We check it for all the elements in s
if ismember(s(q),srem) % If an element in s also belongs to srem
Dsrem(p)=D(q); % we add the value of D on place q to Dsrem
p = p + 1; % we raise the counter to make sure srem corresponds ...
    to the right Dsrem value
end
end

% Make a graph
b = 0 : 0.01 : T; % Interval over which we plot
f = 1 + sin(b); % Intensity function
g = lambda; % Sup of intensity
plot(b,f,'-b',[0 T],[1 1]*g,'--k'); % Plot both graphs in one figure
xlim([0 T]) % x values for the plot
ylim([0 2.1]) % y values for the plot
hold on
for i = 1 : 1 : size(srem,2) % Plot all the points in ...
    homogeneous Poisson process as red
plot(srem(i),0,'r.','MarkerSize',30) % Plot event times on axis
plot(srem(i),Dsrem(i)*lambda,'r.','MarkerSize',30) % Plot ...
    probability density
plot([srem(i) srem(i)], [0 Dsrem(i)*lambda], ...
    ':r','LineWidth',2); % Connect event times and probability ...
    densities
end

for j = 1 : 1 : size(t,2) % Plot all the point in the ...
    nonhomogeneous Poisson process as green
plot(t(j),0,'g.','MarkerSize',30) % Plot event times on axis
plot(t(j),Dt(j)*lambda,'g.','MarkerSize',30) % Plot probability ...
    density
plot([t(j) t(j)], [0 Dt(j)*lambda], ':g','LineWidth',2); % ...
    Connect event times and probability densities
end

xlabel('Time t','FontSize',12)
ylabel('Intensity','FontSize',12)
title('Simulation of a nonhomogeneous Poisson process', ...
    'FontSize', 16)

% Code to make the histogram
histogram(M,'Normalization','probability', 'DisplayName', ...
    'Number of events at T')
xlabel('Number of events','FontSize',12)
ylabel('Probability density','FontSize',12)
title('Number of event times for \lambda(t) = 1 + ...
    sin(t)','FontSize',16)

% Code to obtain parameters for histogram plot with pdf
parameters = fitdist(M.', 'poisson') % Fit data according to a ...
    Poisson distribution

% Code to plot histogram, fitted distribution and actual ...
    distribution in one graph
hold on % Make sure everything ends up in one graph
x = min(M)-2 : 1 : max(M)+2; % Specify part over which we plot ...
    the distributions
fitpois = pdf(parameters, x); % Calculate density for our ...
    distribution for the right values
plot(x, fitpois, 'DisplayName','Fitted ...
    distribution','LineWidth',2) % Plot fitted distribution in ...
    histogram
actualpois = pdf('poisson', x, 11 - cos(10)); % Calculate ...
    density for the true distribution for the right values
plot(x, actualpois, 'DisplayName','Actual ...
    distribution','LineWidth',2) % Plot actual distribution in ...
    histogram
legend % Create a legend
hold off
```

D.3 Code to Simulate the Hawkes Process with Exponential Kernel According to Ogata

```
% While-loop to create the histogram

e=1; % We start the counter to run the simulation 10000 times
while e < 10001 % As long as there haven't been 10000 ...
    simulations, a new simulation will take place

% Code to simulate one Hawkes process

% The input variable:
T = 2; % we simulate over [0,T]
lambda0 = 1.2; % background intensity term
alpha = 0.6; % first parameter exponential kernel
beta = 0.8; % second parameter exponential kernel

% Initialization:
n = 1; % counter for the process (Matlab can't start at 0)
m = 1; % counter for the candidate points (Matlab can't start at 0)
s(1) = 0; % first candidate point of Hawkes process
setT = []; % array of event times

% The while-loop:
while s(m) < T % as long as the candidate points are smaller ...
    than the interval of the simulation we generate new points
% Determine the value of lambda bar
if n > 1 % Check if there are already point in the Hawkes ...
    process, if so calculate the right value of lambda bar
sum = 0;
for i = 1 : 1 : size(setT, 2)
sum = sum + alpha * exp(-beta*(s(m) - setT(i))); % If setT isn't ...
    empty, lambabar is the sum of the intensity of the ...
    individual points
end
lambdabar(m) = lambda0 + sum(end) + alpha; % However we have to ...
    add the background intensity and alpha since the new points ...
    increases the intensity with alpha
else % When there aren't any points in the Hawkes process yet, ...
    the intensity is the background intensity
lambdabar(m) = lambda0;
end

% Update the candidate point
s(m+1) = s(m) - log(rand)/lambdabar(m); % generate the new ...
    candidate point

% Determine the value of the intensity for the new candidate point
if n > 1 % Check if there are already point in the Hawkes ...
    process, if so calculate the right value of lambda
sumnew = 0;
for i = 1 : 1 : size(setT, 2)
sumnew = sumnew + alpha * exp(-beta*(s(m+1) - setT(i))); % If ...
    setT isn't empty, lambabar is the sum of the intensity of ...
    the individual points
end
lambdanew(m) = lambda0 + sumnew(end) + alpha; % However we have ...
    to add the background intensity and alpha since the new ...
    points increases the intensity with alpha
else
```

```
lambdanew(m) = lambda0; % When there aren't any points in the ...
    Hawkes process yet, the intensity is the background intensity
end

% Check the new point
D(m+1) = rand; % Generate a random variable
if D(m+1) * lambdabar(m) ≤ lambdanew(m) % Determine if point ...
    needs to be rejected
setT(n) = s(m+1); % If not rejected the point is added to the ...
    Hawkes process
n = n + 1; % So the counter is raised by one
end

m = m +1; % Otherwise, we go on to the next candidate point
end

% Remove first points from s and D, since nothing occured at T = 0
s(1) = []; % remove starting point homogeneous Process
m = m - 1; % one element of array is removed thus lower the ...
    arraylength
D(1) = []; % remove first point from D
n = n - 1; % had to start at 1

% Remove points that happened after T
if s(m) > T % check for the homogeneous Poisson process
s(m) = []; % if the last value is bigger than T we remove it ...
    from the array
D(m) = []; % in this case the last value of D isn't used anymore
m = m - 1; % in this case the arraysize decreases by one since ...
    one point is removed
end
if setT(n) > T % check for the nonhomogeneous Poisson process
setT(n) = []; % if the last value is bigger than T we remove it ...
    from the array
n = n - 1; % in this case the arraysize decreases by one since ...
    one point is removed
end

% Code for the repeated simulation
M(e)=n; % For every round of the simulation, the number of ...
    events in the Hawkes process will be saved
e = e + 1; % The counter for the number of simulations will be ...
    raised

end

% Code to make the histogram
histogram(M,'Normalization','probability')
xlabel('Counter','FontSize',16)
ylabel('Probability density','FontSize',16)
title('Number of event times for \lambda_0 = 1.2, \alpha = 0.6 ...
    and \beta = 0.8','FontSize',16)

%Code to make the graph if there are five event times
hold on

```

```
g = lambda0; % Plot the background intensity before the first ...
    event occurs
plot([0 setT(1)],[1 1]*g,'-b')

x = setT(1) : 0.001 : setT(2); % Plot the intensity between the ...
    first and second event time
intensity = @(e) lambda0 + alpha*exp(-beta*(e-setT(1))); % Just ...
    sum over one product
scatter(setT(1), intensity(setT(1)),80,'.b')
plot(x, intensity(x), '-b')

y = setT(2)+0.000001 : 0.001 : setT(3); % Plot the intensity ...
    between the second and third event time
intensity2 = @(e) lambda0 + alpha*exp(-beta*(e-setT(1))) + ...
    alpha*exp(-beta*(e-setT(2))); % Sum over two products
scatter(setT(2), intensity2(setT(2)),80,'.b')
plot(y, intensity2(y),'-b')

z = setT(3)+0.000001 : 0.001 : setT(4); % Plot the intensity ...
    between the third and fourth event time
intensity3 = @(e) lambda0 + alpha*exp(-beta*(e-setT(1))) + ...
    alpha*exp(-beta*(e-setT(2))) + alpha*exp(-beta*(e-setT(3))); ...
    % Sum over three products
scatter(setT(3), intensity3(setT(3)),80,'.b')
plot(z, intensity3(z),'-b')

xx = setT(4)+0.000001 : 0.001 : setT(5); % Plot the intensity ...
    between the fourth and fifth event time
intensity4 = @(e) lambda0 + alpha*exp(-beta*(e-setT(1))) + ...
    alpha*exp(-beta*(e-setT(2))) + alpha*exp(-beta*(e-setT(3))) ...
    + alpha*exp(-beta*(e-setT(4))); % Sum over four products
scatter(setT(4), intensity4(setT(4)),80,'.b')
plot(xx, intensity4(xx),'-b')

yy = setT(5)+0.000001 : 0.001 : T; % Plot the intensity between ...
    the fifth event time and T
intensity5 = @(e) lambda0 + alpha*exp(-beta*(e-setT(1))) + ...
    alpha*exp(-beta*(e-setT(2))) + alpha*exp(-beta*(e-setT(3))) ...
    + alpha*exp(-beta*(e-setT(4)))+ ...
    alpha*exp(-beta*(e-setT(5))); % Sum over five products
scatter(setT(5), intensity5(setT(5)),80,'.b')
plot(yy, intensity5(yy),'-b')

xlim([0 T]) % x values for the plot
ylim([0 3.5]) % y values for the plot
xlabel('Time t','FontSize',16)
ylabel('Intensity','FontSize',16)
title('Simulation of a Hawkes process with parameters \lambda_0 ...
    = 1.2, \alpha = 0.6 and \beta = 1.6','FontSize',18)

hold off
```

D.4 Code to Simulate the Hawkes Process with Power Law Kernel According to Ogata

```
% While-loop to create the histogram

e=1; % We start the counter to run the simulation 10000 times
while e < 10001 % As long as there haven't been 10000 ...
    simulations, a new simulation will take place

% Code to simulate one Hawkes process

% The input variable:
T = 2; % we simulate over [0,T]
lambda0 = 1; % background intensity term
K = 3; % first parameter power law kernel
c = 1; % second parameter power law kernel
p = 1; % third parameter power law kernel

% Initialization:
n = 1; % counter for the process (Matlab can't start at 0)
m = 1; % counter for the candidate points (Matlab can't start at 0)
s(1) = 0; % first candidate point of Hawkes process
setT = []; % array of event times

% The while-loop:
while s(m) < T % as long as the candidate points are smaller ...
    than the interval of the simulation we generate new points
% Determine the value of lambda bar
if n > 1 % Check if there are already point in the Hawkes ...
    process, if so calculate the right value of lambda bar
sum = 0;
for i = 1 : 1 : size(setT, 2)
sum = sum + K./((c+(s(m) - setT(i))).^p); % If setT isn't empty, ...
    lambabar is the sum of the intensity of the individual points
end
lambdabar(m) = lambda0 + sum(end) + K./(c.^p); % However we have ...
    to add the background intensity and alpha since the new ...
    points increases the intensity with alpha
else % When there aren't any points in the Hawkes process yet, ...
    the intensity is the background intensity
lambdabar(m) = lambda0;
end

% Update the candidate point
s(m+1) = s(m) - log(rand)/lambdabar(m); % generate the new ...
    candidate point

% Determine the value of the intensity for the new candidate point
if n > 1 % Check if there are already point in the Hawkes ...
    process, if so calculate the right value of lambda
sumnew = 0;
for i = 1 : 1 : size(setT, 2)
sumnew = sumnew + K./((c+(s(m+1) - setT(i))).^p); % If setT ...
    isn't empty, lambabar is the sum of the intensity of the ...
    individual points
end
lambdanew(m) = lambda0 + sumnew(end) + K./(c.^p); % However we ...
    have to add the background intensity and alpha since the new ...
    points increases the intensity with alpha
else
lambdanew(m) = lambda0; % When there aren't any points in the ...
    Hawkes process yet, the intensity is the background intensity
end

% Check the new point
D(m+1) = rand; % Generate a random variable
if D(m+1) * lambdabar(m) ≤ lambdanew(m) % Determine if point ...
    needs to be rejected
setT(n) = s(m+1); % If not rejected the point is added to the ...
    Hawkes process
n = n + 1; % So the counter is raised by one
end

m = m +1; % Otherwise, we go on to the next candidate point
end

% Remove first points from s and D, since nothing occured at T = 0
s(1) = []; % remove starting point homogeneous Process
m = m - 1; % one element of array is removed thus lower the ...
    arraylength
D(1) = []; % remove first point from D
n = n - 1; % had to start at 1

% Remove points that happened after T
if s(m) > T % check for the homogeneous Poisson process
s(m) = []; % if the last value is bigger than T we remove it ...
    from the array
D(m) = []; % in this case the last value of D isn't used anymore
m = m - 1; % in this case the arraysize decreases by one since ...
    one point is removed
end
if setT(n) > T % check for the nonhomogeneous Poisson process
setT(n) = []; % if the last value is bigger than T we remove it ...
    from the array
n = n - 1; % in this case the arraysize decreases by one since ...
    one point is removed
end

% Code for the repeated simulation
M(e)=n; % For every round of the simulation, the number of ...
    events in the Hawkes process will be saved
e = e + 1; % The counter for the number of simulations will be ...
    raised

end

% Code to make the histogram
histogram(M,'Normalization','probability')
xlabel('Counter','FontSize',16)
ylabel('Probability density','FontSize',16)
```

```
title('Number of event times for \lambda_0 = 1, K = 3, c = 1 and ...
    p = 1','FontSize',18)

% Code to make the graph if there are five event times
hold on

g = lambda0; % Plot the background intensity before the first ...
    event occurs
plot([0 setT(1)],[1 1]*g,'-b')

x = setT(1) : 0.001 : setT(2); % Plot the intensity between the ...
    first and second event time
intensity = @(e) lambda0 + K./((c+(e-setT(1))).^p); % Just sum ...
    over one product
scatter(setT(1), intensity(setT(1)),80,'.b')
plot(x, intensity(x), '-b')

y = setT(2)+0.000001 : 0.001 : setT(3); % Plot the intensity ...
    between the second and third event time
intensity2 = @(e) lambda0 + K./((c+(e-setT(2))).^p) + ...
    K./((c+(e-setT(1))).^p); % Sum over two products
scatter(setT(2), intensity2(setT(2)),80,'.b')
plot(y, intensity2(y),'-b')

z = setT(3)+0.000001 : 0.001 : setT(4); % Plot the intensity ...
    between the third and fourth event time
intensity3 = @(e) lambda0 + K./((c+(e-setT(3))).^p) + ...
    K./((c+(e-setT(2))).^p) + K./((c+(e-setT(1))).^p); % Sum ...
    over three products
scatter(setT(3), intensity3(setT(3)),80,'.b')
plot(z, intensity3(z),'-b')

xx = setT(4)+0.000001 : 0.001 : setT(5); % Plot the intensity ...
    between the fourth and fifth event time
intensity4 = @(e) lambda0 + K./((c+(e-setT(4))).^p) + ...
    K./((c+(e-setT(3))).^p) + K./((c+(e-setT(2))).^p) + ...
    K./((c+(e-setT(1))).^p); % Sum over four products
scatter(setT(4), intensity4(setT(4)),80,'.b')
plot(xx, intensity4(xx),'-b')

yy = setT(5)+0.000001 : 0.001 : T; % Plot the intensity between ...
    the fifth event time and T
intensity5 = @(e) lambda0 + K./((c+(e-setT(4))).^p) + ...
    K./((c+(e-setT(4))).^p) + K./((c+(e-setT(3))).^p) + ...
    K./((c+(e-setT(2))).^p) + K./((c+(e-setT(1))).^p); % Sum ...
    over five products
scatter(setT(5), intensity5(setT(5)),80,'.b')
plot(yy, intensity5(yy),'-b')

xlim([0 T]) % x values for the plot
ylim([0 4.5]) % y values for the plot
xlabel('Time t','FontSize',16)
ylabel('Intensity','FontSize',16)
title('Simulation of a Hawkes process with parameters \lambda_0 ...
    = 1, K = 1, c = 1 and p = 2','FontSize',18)

hold off
```