Patrick J. Laub
Young Lee
Thomas Taimre

The Elements of Hawkes Processes

Springer

The Elements of Hawkes Processes

Patrick J. Laub • Young Lee • Thomas Taimre

The Elements of Hawkes Processes

Springer

Patrick J. Laub
Faculty of Business and Economics
University of Melbourne
Melbourne, VIC, Australia

Young Lee
Faculty of Arts and Sciences
Harvard University
Cambridge, MA, USA

Thomas Taimre
School of Mathematics and Physics
The University of Queensland
Brisbane, QLD, Australia

ISBN 978-3-030-84638-1 ISBN 978-3-030-84639-8 (eBook)
https://doi.org/10.1007/978-3-030-84639-8

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2021
This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To Viv
. . . and to coffee & croissants

—Pat

To AKM
. . . for the fun times we shared in the great realm down-under

—Young

To my siblings, Peter and Linda
. . . for their constant support

—Tom

Preface

There has been a remarkable interest in Hawkes processes these past few years as seen through their increasing application in the natural and social sciences, culminating in a considerable amount of research work. While there are excellent books available on general point processes, little has been written specifically for Hawkes processes. Hence, we feel there has never been a better time to write a book giving an overview of the crucial aspects of Hawkes processes.

As they have been used in a wide range of disciplines including mathematics, social sciences, machine learning, and earthquake modelling, the diversity of topics cannot be comprehensively covered in a single volume. This book aims to present a selective coverage of the core elements and recent topics from within the broad field of Hawkes processes.

This book is primarily directed at practitioners in the field of applied probability, statisticians, and machine learners. By keeping the mathematical prerequisites simple and to a minimum, with colourblind-friendly illustrations and case studies, the book will be of interest and accessible to the majority of readers. As far as possible, the development is self-contained while necessarily condensed.

The book is divided into three distinct parts, namely Parts I, II, and III, together with supplementary materials. Parts I and II comprise the introduction and the following eight chapters. These chapters discuss basic elements of the Hawkes processes, ranging from fundamental properties to simulation methods, as well as discussing inference. We also elaborate on the theory of random time change and apply it to the Hawkes processes. The focus of Parts I and II is on Hawkes processes and not their application, which is brought into discussion in Part III. This part deals with their applications in seismology and finance. Seismology is the scientific study of earthquakes; we posit the arrivals of earthquakes as a Hawkes process and examine the model's performance. In the financial analysis application, we turn to

an application of Hawkes processes to describe financial contagion, where large movements in one stock propagate to another stock.

Appendix A.1 provides background material on statistical theory and stochastic processes. Additional proof details in the book are summarised in Appendix A.2.

Melbourne, VIC, Australia Patrick J. Laub

Cambridge, MA, USA Young Lee

Brisbane, QLD, Australia Thomas Taimre

December 24, 2021

Contents

Notation

We place great importance on notation throughout this book, and strive for simplicity, descriptiveness, consistency, and compatibility with historical notational conventions (in that order).

We make use of a number of typographical conventions to aid the reader throughout, including:

- The use of boldface font to indicate vectors $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, which are viewed as column vectors unless otherwise specified
- Observing the upper case/lower case distinction between a random variable (vector) X ($\boldsymbol{X}$) and its outcome x ($\boldsymbol{x}$)
- Indicating distributions with a sans serif font, for example, $X \sim \mathsf{Poi}(\lambda)$ means that the random variable X has a Poisson distribution with parameter λ
- Adopting a typical Bayesian notation convention from time to time wherein the same symbol is used to denote different (conditional) probability density functions, for example, $f(x)$, $f(x \mid y)$, and $f(y \mid x)$ would respectively be the (marginal) probability density function of X, the conditional density of $(X \mid Y = y)$, and the conditional density of $(Y \mid X = x)$

Below, we highlight recurring notation, acronyms, and abbreviations. We have deliberately sought to minimise the usage of the latter two.

$\mathbb{R}$	Set of real numbers
$\mathbb{N}$	Set of natural numbers, i.e. $\{1, 2, 3, \ldots\}$
$\mathbb{N}_0$	Set of natural numbers with zero, i.e. $\{0, 1, 2, \ldots\}$
$\mathbb{C}\mathrm{ov}$	Covariance
$\mathbb{E}$	Expectation
$\mathbb{P}$	Probability
$\mathcal{H}$	Filtration
$\sim$	Is distributed as
$\overset{\text{IID}}{\sim}$	Is independently and identically distributed as
Ber	Bernoulli distribution
Beta	Beta distribution

Bin	Binomial distribution
Erl	Erlang distribution
Exp	Exponential distribution
Normal	Normal distribution
Poi	Poisson distribution
Unif	Uniform distribution
e	The number 2.7182818…
f	Probability density (continuous or discrete)
$\mathbb{I}_A$ or $\mathbb{I}\{A\}$	Indicator function of set A
∇f	Gradient of f
ln	(Natural) logarithm
$\mathcal{O}$	Big-O order symbol
$\propto$	Proportional to
$\boldsymbol{x}$, $\boldsymbol{y}$	Vectors
$\boldsymbol{X}$, $\boldsymbol{Y}$	Random vectors

BIC	Bayesian information criterion
CDF	cumulative distribution function
EM	expectation–maximisation
ETAS	epidemic-type aftershock sequence
GMM	generalised method of moments
IID	independent and identically distributed
MCMC	Markov chain Monte Carlo
MH	Metropolis–Hastings
MLE	maximum-likelihood estimator
MM	method of moments
ODE	ordinary differential equation
PDF	probability density function
Q–Q	quantile–quantile

Chapter 1
Introduction

What do the occurrences of earthquakes, gang violence, trade orders, and bank defaults have in common? They all exhibit 'self-exciting' behaviour: an earthquake usually creates aftershocks in the region [55]; a fight between rival gangs can ignite a spate of criminal retaliations [52]; selling a significant quantity of a stock could precipitate a trading flurry or, on a larger scale, the collapse of a Wall Street investment bank could send shockwaves through the world's financial centres [3].

The Hawkes process is a mathematical model for these 'self-exciting' processes, named after its creator Alan G. Hawkes [35]. The Hawkes process is a counting process that models a sequence of 'arrivals' of some type over time, for example, earthquakes, gang violence, trade orders, or bank defaults. Each arrival *excites* the process in the sense that the chance of a subsequent arrival is increased for some time period after the initial arrival. Thus, the Hawkes process is structured around the premise that the history of the process matters, unlike many standard probability models. It can be viewed as a non-Markovian extension of the classical Poisson process.

In addition to background theory, the process of generating, model fitting, model evaluation, and applying Hawkes processes in practice is examined in this book. The book is organised as follows:

Chapter 2 introduces the theory of point processes where it recapitulates some parts of counting process theory needed in what follows in the book. Properties of Poisson point processes are detailed using a constructive approach that is accessible to a broad audience. Furthermore, it describes the conditional intensity function as a way to characterise point processes. The emphasis throughout this chapter is on point processes themselves.

Chapter 3 discusses several of the useful and important properties of Hawkes processes. We formally introduce Hawkes processes through the conditional intensity function. This approach enables those new to the subject to quickly understand the core concepts of Hawkes processes. The description of Hawkes processes which includes the immigrant–birth interpretation is also detailed. The excitation function,

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

P. J. Laub et al., *The Elements of Hawkes Processes*,
https://doi.org/10.1007/978-3-030-84639-8_1

which conveys the positive influence of the past events on the current value of the intensity process is discussed, as are the notions of covariance and power spectral densities.

Chapter 4 discusses simulation methods for Hawkes processes. It starts with a discussion on transformation methods, which are at the heart of producing realisations of Hawkes processes. For the special case of exponential excitation function, we present the exact simulation of the Hawkes process. The well known Ogata's modified thinning method of generating point processes is explained. The mutually exciting Hawkes process which exhibits cross excitation is briefly examined.

Chapter 5 focuses on the maximum likelihood estimation (MLE) of Hawkes process models. The likelihood function is discussed in detail. MLE for Hawkes processes is typically carried out via the numerical maximisation of the log-likelihood function. In the case of an exponential excitation function, the number of operations required to evaluate the log-likelihood can be reduced using a recursive formula. We discuss the recursive formulæ for Hawkes processes.

Chapter 6 introduces the expectation–maximisation (EM) algorithm applied to Hawkes processes. In a nutshell, the EM algorithm is a general iterative algorithm that can be used to find the maximum likelihood estimates where the model depends on some unobserved quantities. In our case, the unobserved branching structure under the immigrant–birth interpretation outlined in Chap. 3 is treated as the missing data and can be used to construct an EM algorithm.

Chapter 7 introduces the problem of parameter estimation from Hawkes processes using the generalised method of moments (GMM) method. The idea of the GMM is to determine estimates of the parameters by setting sample moments to be as close as possible to their population counterparts. This approach of parameter estimation is typically suitable when the only available information is the number of aggregated events over a given interval, rather than the exact event times.

Chapter 8 explores the Bayesian inference method for the Hawkes process. Starting with a slightly generalised formulation of the conditional intensity function, we first define the corresponding likelihood function. We then place suitable priors over the parameters of our model. The posterior distribution is then derived, and the Markov chain Monte Carlo (MCMC) approach is then used for the estimation of Hawkes parameters.

Chapter 9 discusses model diagnostics for Hawkes processes. We detail tests used to check the fit of Hawkes models, known as the goodness of fit tests. The goodness of fit tests that we describe predominantly make use of the random time change theorem. In addition, various tests for Poisson processes are discussed.

Chapter 10 translates the various theories and procedures we have gone through into Python code. We present a detailed guideline to setting parameters for these implementations in an accessible manner. The code is available online as the `hawkesbook` Python package, and it is used in the subsequent applications chapters.

Chapter 11 provides a case study for the use of Hawkes processes in fitting earthquake arrival times. This is a very traditional field where the Hawkes process

first gained its prominence. This section aims to imprint upon the readers the motivation and perspective of early Hawkes researchers to model earthquakes using this model. We apply the Hawkes process methodology to earthquake data in the Japan region from 1973 to 2020 and compare the fits of the Hawkes models to the baseline Poisson process.

Chapter 12 covers financial applications and provides a case study for the use of mutually exciting Hawkes processes in modelling stock-market trades. We review some recent developments of the Hawkes process in finance. Lastly, we break down the work of [28] on using the Hawkes diffusion model; this is an extension of prior models of financial contagion for mid-price changes to include Hawkes processes.

Part I
Basic Theory

Chapter 2
Background

Before discussing Hawkes processes, some key concepts must be elucidated. Firstly, we briefly give definitions for counting processes and point processes, thereby setting essential notation. Secondly, we discuss the lesser-known conditional intensity function and compensator, both core concepts for a clear understanding of Hawkes processes.

2.1 Counting and Point Processes

We begin with the definition of a counting process.

Definition 2.1 (Counting Process and History) A *counting process* is a stochastic process $(N(t) : t \geqslant 0)$ taking values in $\mathbb{N}_0$ that satisfies $N(0) = 0$, is almost surely finite, and is a right-continuous step function with increments of size $+1$.

Further, denote by $(\mathcal{H}(u) : u \geqslant 0)$ the *history* of the arrivals up to time u. (Strictly speaking $\mathcal{H}(\cdot)$ is a filtration, that is, an increasing sequence of σ-algebras.) $\diamond$

A counting process can be viewed as a cumulative count of the number of 'arrivals' into a system up to the current time. Another way to characterise such a process is to consider the sequence of random arrival times $\boldsymbol{T} = \{T_1, T_2, \dots\}$ at which the counting process $N(\cdot)$ has jumped. The process defined as these arrival times is called a point process, described in Definition 2.2 (adapted from [14]); see Fig. 2.1 for an example point process and its associated counting process.

Definition 2.2 (Point Process) If a sequence of random variables $\boldsymbol{T} = \{T_1, T_2, \dots\}$, taking values in $[0, \infty)$, has $\mathbb{P}(0 \leqslant T_1 \leqslant T_2 \leqslant \dots) = 1$, and the number of points in a bounded region is almost surely finite, then $\boldsymbol{T}$ is a *(simple) point process*. $\diamond$

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

P. J. Laub et al., *The Elements of Hawkes Processes*,
https://doi.org/10.1007/978-3-030-84639-8_2

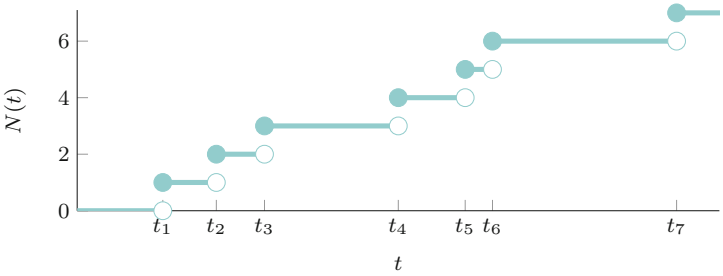

Fig. 2.1 An example point process realisation $\{t_1, t_2, \dots\}$ and corresponding counting process $N(t)$

The counting and point process terminology is often interchangeable. For example, if one refers to a Poisson process or a Hawkes process (both defined shortly), then the reader must infer from the context whether the counting process $N(\cdot)$ or the point process of times $\boldsymbol{T}$ is being discussed.

The definitions above consider processes with points on the semi-infinite time axis $[0, \infty)$. Point processes are also often considered on the entire infinite time axis $(-\infty, \infty)$; however, we shall restrict attention in this work to $[0, \infty)$ (see Remark 3.1).

Before we proceed, we note that point processes can be characterised by different dependence among random variables. If the real-valued random variables

$$N(t_2) - N(t_1),\ N(t_3) - N(t_1), \dots,\ N(t_n) - N(t_{n-1}) \tag{2.1}$$

are independent for all choices of $t_1, t_2, \dots, t_n$, for $t_i \geqslant 0$, satisfying

$$t_1 < t_2 < \cdots < t_n, \tag{2.2}$$

then $(N(t) : t \geqslant 0)$ is said to be a process with *independent increments*. If the distribution of the increments

$$N(t_1 + h) - N(t_1) \tag{2.3}$$

depends only on the length interval h and not on time t_1, the process is said to have *stationary increments*. For a process with stationary increments, the distribution of $N(t_1 + h) - N(t_1)$ is the same as the distribution of $N(t_2 + h) - N(t_2)$, regardless of the values of t_1, t_2, and h.

2.2 Poisson Processes

There are a number of equivalent definitions for the Poisson process. Here, we define a Poisson process in terms of its behaviour on intervals, as follows.

Definition 2.3 (Poisson Process) A counting process $(N(t) : t \geqslant 0)$ is a *homogeneous Poisson process* with rate $\lambda > 0$ if

1. For any interval I, $N(I) \sim \mathsf{Poi}(\lambda|I|)$;
2. For any n disjoint intervals $I_1, I_2, \ldots, I_n$, the random variables $N(I_1)$, $N(I_2), \ldots, N(I_n)$ are independent. ◇

We have directly from Definition 2.3 that

$$\mathbb{P}(N(t) = n) = \frac{(\lambda t)^n}{n!} \mathrm{e}^{-\lambda t}, \qquad n = 0, 1, 2, \ldots .$$

Let T_n be the time of the n-th arrival of a homogeneous Poisson process. To determine the distribution of T_n, we proceed as follows. Using the equivalence of the events $\{T_n > t\} = \{N(t) \leqslant n - 1\}$ for $n = 1, 2, \ldots$, we may write

$$\mathbb{P}(T_n > t) = \mathbb{P}(N(t) \leqslant n - 1) = \sum_{k=0}^{n-1} \frac{(\lambda t)^k}{k!} \mathrm{e}^{-\lambda t} .$$

The variable T_n is said to have an $\mathsf{Erl}(n, \lambda)$ distribution. Note that this is the distribution of the sum of n independent $\mathsf{Exp}(\lambda)$ distributed random variables (a fact easily verified by considering moment generating functions). This demonstrates that the interarrival times of a homogeneous Poisson process are independent and identically distributed (IID).

There is a link between Bernoulli and Poisson processes. Fixing some (small) $h > 0$, partition time into a disjoint intervals $J_1^{(h)} = (0, h]$, $J_2^{(h)} = (h, 2h], \ldots$ and with probability $p = \lambda h$ an arrival occurs in interval $J_n^{(h)}$ (otherwise no arrival occurs). That is, the number of arrivals in $J_n^{(h)}$ is $X_n \sim \mathsf{Ber}(p)$. The sequence $(X_n, n = 1, 2, \ldots)$ forms a Bernoulli process with parameter p. Defining $Y_n = X_1 + \cdots + X_n$, Y_n then denotes the number of arrivals in the interval $(0, nh]$.

Now consider a fixed time $t > 0$, which falls in the interval $n = \lfloor t/h \rfloor$. The random variables $N(t)$ and Y_n have approximately the same distribution.

Indeed, $Y_n \sim \mathsf{Bin}(n, p)$ and

$$\begin{aligned} \mathbb{P}(Y_n = m) &= \binom{n}{m} (\lambda h)^m (1 - \lambda h)^{n-m} \\ &\approx \binom{n}{m} (\lambda t/n)^m (1 - \lambda t/n)^{n-m} \end{aligned}$$

$$= \frac{n!}{(n-m)!n^m} \frac{(\lambda t)^m}{m!} \left(1 - \frac{\lambda t}{n}\right)^{n-m}.$$

In the limit as $n \to \infty$ we can use Stirling's formula $n! \sim \sqrt{2\pi} n^{n+1/2} \mathrm{e}^{-n}$ and the standard fact that $\lim_{n\to\infty}(1 + x/n)^n = \mathrm{e}^x$ to find

$$\lim_{n\to\infty} \mathbb{P}(Y_n = m) = \frac{(\lambda t)^m}{m!} \mathrm{e}^{-\lambda t}.$$

Both the Bernoulli process and Poisson process have stationary and independent increments.

The interarrival times of the Bernoulli process are geometric, and those of the Poisson process are exponential. Note that both geometric and exponential distributions have the memoryless property.

Definition 2.4 (Inhomogeneous Poisson Process) A counting process $(N(t) : t \geqslant 0)$ is an *inhomogeneous Poisson process* with rate function $\lambda(t) > 0$ if

1. For any interval $I = (a, b]$, $N(I) \sim \mathsf{Poi}\left(\int_a^b \lambda(s)\,\mathrm{d}s\right)$;
2. For any n disjoint intervals $I_1, I_2, \ldots, I_n$, the random variables $N(I_1)$, $N(I_2), \ldots, N(I_n)$ are independent. ◇

Unlike the homogeneous Poisson process, the inhomogeneous Poisson process has dependent interarrival times in general.

We now list a few additional useful properties of the Poisson process.

Theorem 2.1 (Superposition of Poisson Processes) *Given two independent Poisson processes $N_1 = (N_1(t) : t \geqslant 0)$ and $N_2 = (N_2(t) : t \geqslant 0)$ with rates λ_1 and λ_2, respectively, create a new process $N = N_1 + N_2$. Then N is a Poisson process with rate $\lambda = \lambda_1 + \lambda_2$.*

This result is known as *superposition*, and it generalises to the inhomogeneous case, as well as to an arbitrary number of processes.

Theorem 2.2 (Thinning of a Poisson Process) *Let $N = (N(t) : t \geqslant 0)$ be a Poisson processes. For each arrival of N, assign it to process N_1 with probability p and to process N_0 with probability $(1-p)$. The new processes N_1 and N_0 created in this way are also Poisson process with rates $p\lambda$ and $(1-p)\lambda$, respectively. Moreover, they are independent!*

This result is known as *thinning*, and it generalises to splitting the process in to $m \geqslant 2$ processes according to probabilities $p_1, \ldots, p_m$ such that $\sum_k p_k = 1$. Moreover, the result generalises to the inhomogeneous case: if points are assigned to N_1 with time-dependent probability $p(t)$, then N_1 is an inhomogeneous Poisson process with rate function $\lambda(t) = \lambda p(t)$. This fact can be used to devise a simulation algorithm for the inhomogeneous Poisson process.

Theorem 2.3 (Poisson Process Conditional Distribution) *Given $N(t) = n$, the times of arrivals $T_1, \ldots, T_n$ are uniformly distributed on* $[0, t]$.

This result generalises: for any interval I, given $N(I) = n$ for an inhomogeneous Poisson process, the n points in I are independent and distributed in I with probability density function (PDF) $f_I(t) = \lambda(t) / \int_I \lambda(s)\, \mathrm{d}s$, $t \in I$.

2.3 Conditional Intensity Functions

One way to characterise a particular point process is to specify the distribution function of the next arrival time conditional on the past. Given the history up until the last arrival u, $\mathcal{H}(u)$, define (as per [57]) the conditional cumulative distribution function (CDF) (and PDF) of the next arrival time T_{k+1} as

$$F(t \mid \mathcal{H}(u)) = \int_u^t \mathbb{P}(T_{k+1} \in [s, s + \mathrm{d}s] \mid \mathcal{H}(u))\, \mathrm{d}s = \int_u^t f(s \mid \mathcal{H}(u))\, \mathrm{d}s\,.$$

The joint PDF for a realisation $\{t_1, t_2, \ldots, t_k\}$ is then, by the chain rule,

$$f(t_1, t_2, \ldots, t_k) = \prod_{i=1}^{k} f(t_i \mid \mathcal{H}(t_{i-1}))\,. \tag{2.4}$$

In the literature the notation rarely specifies $\mathcal{H}(\cdot)$ explicitly, but rather a superscript asterisk is used (see, for example, [19]). We follow this convention and abbreviate $F(t \mid \mathcal{H}(u))$ and $f(t \mid \mathcal{H}(u))$ to $F^*(t)$ and $f^*(t)$, respectively.

Remark 2.1 The function $f^*(t)$ can be used to classify certain classes of point processes. For example, if a point process has an $f^*(t)$ which is independent of $\mathcal{H}(t)$ then the process is a *renewal process*. ◇

Often it is difficult to work with the conditional arrival distribution $f^*(t)$. Instead, another characterisation of point processes is used: the conditional intensity function. Indeed if the conditional intensity function exists it uniquely characterises the finite-dimensional distributions of the point process (see Proposition 7.2.IV of [19]). Originally this function was called the hazard function [16] and was defined as

$$\lambda^*(t) = \frac{f^*(t)}{1 - F^*(t)}\,. \tag{2.5}$$

Although this definition is valid, we prefer an intuitive representation of the conditional intensity function as the expected rate of arrivals conditioned on $\mathcal{H}(t)$:

Definition 2.5 (Conditional Intensity Function) Consider a counting process $N(\cdot)$ with associated histories $\mathcal{H}(\cdot)$. If a (non-negative) function $\lambda^*(t)$ exists such

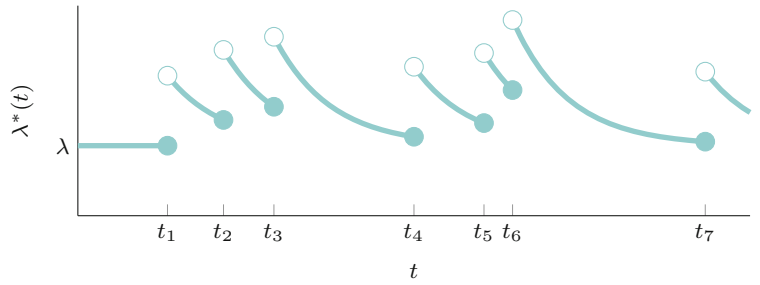

Fig. 2.2 An example conditional intensity function for a self-exciting process

that

$$\lambda^*(t) = \lim_{h \searrow 0} \frac{\mathbb{E}[N(t+h) - N(t) \mid \mathcal{H}(t)]}{h}$$

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

We remark that $\lambda^*(t)$ is interpreted as its left-continuous modification $\lambda^*(t^-)$, to ensure predicability of the counting process $N(\cdot)$.

Further, we remark that the conditional intensity as it is defined here relates to the likelihood of a point appearing in an extension of the observation region, rather than the likelihood of observing a point within the observation region—the latter viewpoint is common in the study of spatial point processes and is otherwise known as the (first order) Papangelou intensity.

The terms 'self-exciting' and 'self-regulating' can be made precise by using the conditional intensity function. If an arrival causes the conditional intensity function to increase then the process is said to be *self-exciting*. This behaviour causes temporal clustering of $\boldsymbol{T}$. In this setting $\lambda^*(t)$ must be chosen to avoid *explosion*, where we use the standard definition of explosion as the event that $N(t)-N(s) = \infty$ for $t - s < \infty$. See Fig. 2.2 for an example realisation of such a $\lambda^*(t)$.

Alternatively, if the conditional intensity function drops after an arrival the process is called *self-regulating* and the arrival times appear quite temporally regular. Such processes are not examined hereafter, though an illustrative example would be the arrival of speeding tickets to a driver over time (assuming each arrival causes a period of heightened caution when driving).

The next definition defines stochastic integral with respect to the counting process, which is used frequently in the calculations that concern Hawkes processes. First, we remark that the jump size of N at time t is denoted $\Delta N(t) = N(t)-N(t^-)$. Hence the stochastic integral (also known as the Itô integral) with respect to N is given as follows:

Definition 2.6 (Stochastic Integral with Respect to N) Consider the counting process $N(t)$ and let $\Psi(s)$ be a measurable process. The stochastic integral of Ψ

with respect to N is defined to be

$$\int_0^t \Psi(u)\,\mathrm{d}N(u) = \sum_{0<u\leqslant t} \Psi(u)\,\Delta N(u). \tag{2.6}$$

◇

We will see that this stochastic integral will be used in the definition of Hawkes process in Chap. 3.

2.4 Compensators

Frequently the integrated conditional intensity function is needed (for example, in parameter estimation and goodness of fit testing); it is defined as follows.

Definition 2.7 (Compensator) For a counting process $N(\cdot)$ the non-decreasing function

$$\Lambda(t) = \int_0^t \lambda^*(s)\ \mathrm{d}s$$

is called the *compensator* of the counting process. ◇

In fact, a compensator is usually defined more generally and exists even when $\lambda^*(\cdot)$ does not exist. Technically $\Lambda(t)$ is the unique $\mathcal{H}(t)$ predictable function, with $\Lambda(0) = 0$, and is non-decreasing, such that $N(t) = M(t) + \Lambda(t)$ almost surely for $t \geqslant 0$ and where $M(t)$ is an $\mathcal{H}(t)$ local martingale, whose existence is guaranteed by the Doob–Meyer decomposition theorem [48]. However, for Hawkes processes $\lambda^*(\cdot)$ always exists (in fact, as we shall see in Chap. 3, a Hawkes process is defined in terms of this function) and therefore Definition 2.7 is sufficient for our purposes.

Chapter 3
Hawkes Process Essentials

With essential background and core concepts outlined in Chap. 2, we now turn to discussing Hawkes processes, including their useful immigration–birth representation and briefly touching on generalisations.

3.1 The Hawkes Process

Point processes gained a significant amount of attention in the field of statistics during the 1950s and 1960s. First, Cox [16] introduced the notion of a doubly stochastic Poisson process (now called the Cox process) and Bartlett [5–7] investigated statistical methods for point processes based on their power spectral densities. At IBM Research Laboratories, Lewis [42] formulated a point process model (for computer failure patterns) which was a step in the direction of the Hawkes process. The activity culminated in the significant monograph by Cox and Lewis [17] on time series analysis; modern researchers appreciate this text as an important development of point process theory since it canvassed their wide range of applications [19, p. 16].

It was in this context that Hawkes [35] set out to bring Bartlett's spectral analysis approach to a new type of process: a self-exciting point process. The process Hawkes described was a one-dimensional point process defined as follows.

Definition 3.1 (Hawkes Process) Consider $(N(t) : t \geqslant 0)$ a counting process, with associated history $(\mathcal{H}(t) : t \geqslant 0)$, that satisfies

$$\mathbb{P}(N(t+h) - N(t) = m \mid \mathcal{H}(t)) = \begin{cases} 1 - \lambda^*(t)\, h + \mathrm{o}(h)\,, & m = 0 \\ \lambda^*(t)\, h + \mathrm{o}(h)\,, & m = 1 \\ \mathrm{o}(h)\,, & m > 1 \end{cases}.$$

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

P. J. Laub et al., *The Elements of Hawkes Processes*,
https://doi.org/10.1007/978-3-030-84639-8_3

Suppose the process' conditional intensity function is of the form

$$\lambda^*(t) = \lambda + \int_0^t \mu(t-u)\, \mathrm{d}N(u) \tag{3.1}$$

for some $\lambda > 0$ and $\mu : (0, \infty) \to [0, \infty)$ which are called the *background intensity* and *excitation function*, respectively. Assume that $\mu(\cdot) \neq 0$ to avoid the trivial case, that is, a homogeneous Poisson process. Such a process $N(\cdot)$ is a *Hawkes process*. ◇

Remark 3.1 The definition above has $t \geq 0$, however, the original form given by Hawkes [35] considers $t \in \mathbb{R}$ and sets $N(t)$ as the number of arrivals in $(0, t]$. Typically Hawkes process results hold for both definitions, though we will use $t \geq 0$ as the primary definition and declare the specific cases when the $t \in \mathbb{R}$ formulation is required. ◇

Remark 3.2 In the standard terminology, Definition 3.1 describes a *linear* Hawkes process—the *nonlinear* version is given later in Definition 3.2. Unless otherwise qualified, the Hawkes processes in this book will refer to this linear form. ◇

A realisation of a Hawkes process is shown in Fig. 3.1 with the associated path of the conditional intensity process. Hawkes [36] soon extended this single point

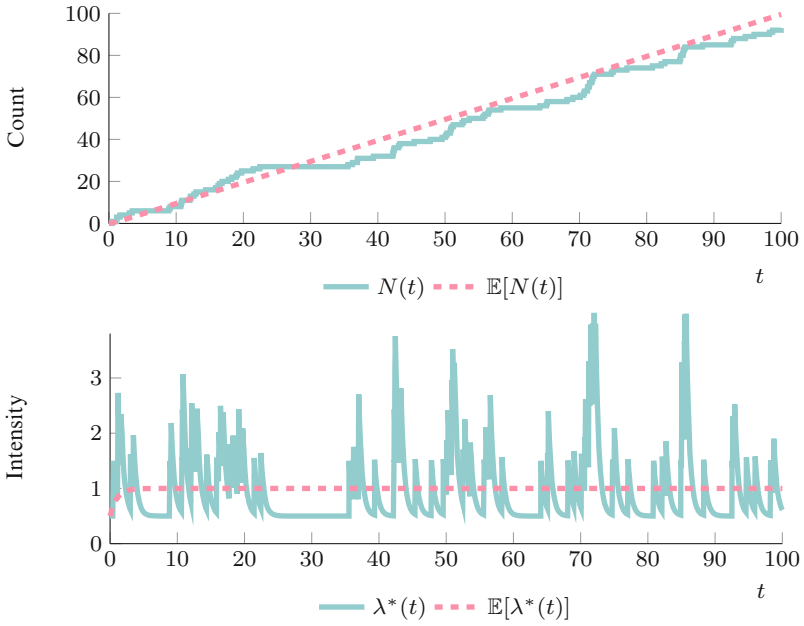

Fig. 3.1 (**a**) A typical Hawkes process realisation $N(t)$, and its associated $\lambda^*(t)$ in (**b**), both plotted against their expected values

process into a collection of self- and mutually exciting point processes, which we will turn to discussing after elaborating upon this one-dimensional process.

3.2 Hawkes Conditional Intensity Function

The form of the Hawkes conditional intensity function in (3.1) is consistent with the literature though it somewhat obscures the intuition behind it. Using $\{t_1, t_2, \dots, \}$ to denote the observed sequence of past arrival times of the point process, the Hawkes conditional intensity is

$$\lambda^*(t) = \lambda + \sum_{t_i < t} \mu(t - t_i)\,. \tag{3.2}$$

The structure of this $\lambda^*(\cdot)$ is quite flexible and only requires specification of the background intensity $\lambda > 0$ and the excitation function $\mu(\cdot)$. A common choice for the excitation function is one which decays exponentially; Hawkes [35] originally used this form as it simplified his theoretical derivations [34]. In this case $\mu(t) = \alpha\, \mathrm{e}^{-\beta t}$, which is parameterised by constants $\alpha, \beta > 0$, and

$$\lambda^*(t) = \lambda + \int_0^t \alpha \mathrm{e}^{-\beta(t-s)}\, \mathrm{d}N(s) = \lambda + \sum_{t_i < t} \alpha \mathrm{e}^{-\beta(t-t_i)}\,. \tag{3.3}$$

The constants α and β have the following interpretation: each arrival in the system instantaneously increases the arrival intensity by α, then over time this arrival's influence decays exponentially at rate β.

Another frequent choice for $\mu(\cdot)$ is a power law function

$$\mu(t) = \frac{k}{(c + t)^p} \tag{3.4}$$

with $k, c > 0$ and $p > 1$. The resulting conditional intensity function is

$$\lambda^*(t) = \lambda + \int_0^t \frac{k}{(c + (t - s))^p}\, \mathrm{d}N(s) = \lambda + \sum_{t_i < t} \frac{k}{(c + (t - t_i))^p}. \tag{3.5}$$

The power law form was popularised by the geological model called Omori's law, used to predict the rate of aftershocks caused by an earthquake [56]. More computationally efficient than either of these excitation functions is a piecewise linear function as in [15]. However, the remaining discussion will focus on the exponential form of the excitation function, sometimes referred to as the Hawkes process with *exponentially decaying intensity*.

One can consider the impact of setting an initial condition $\lambda^*(0) = \lambda_0$, perhaps in order to model a process from some time after it is started. In this scenario the conditional intensity process (using the exponential form of $\mu(\cdot)$) satisfies the stochastic differential equation

$$\mathrm{d}\lambda^*(t) = \beta(\lambda - \lambda^*(t))\,\mathrm{d}t + \alpha\,\mathrm{d}N(t)\,, \quad t \geqslant 0\,.$$

Applying stochastic calculus yields the general solution of

$$\lambda^*(t) = \mathrm{e}^{-\beta t}(\lambda_0 - \lambda) + \lambda + \int_0^t \alpha \mathrm{e}^{\beta(t-s)}\,\mathrm{d}N(s)\,, \quad t \geqslant 0\,,$$

which is a natural extension of (3.3) [18].

3.3 Immigration–Birth Representation

Stability properties of the Hawkes process are often simpler to divine if it is viewed as a branching process. Imagine counting the population in a country where people arrive either via *immigration* or by *birth*. Say that the stream of immigrants to the country form a homogeneous Poisson process at rate λ. Each individual then produces zero or more children independently of one another, and the arrival of births forms an inhomogeneous Poisson process.

An illustration of this interpretation can be seen in Fig. 3.2. In branching theory terminology, this *immigration–birth representation* describes a Galton–Watson process with a modified time dimension. Hawkes [37] used the representation to derive asymptotic characteristics of the process, such as the following result.

Theorem 3.1 (Hawkes Process Asymptotic Normality) *If*

$$0 < n \coloneqq \int_0^\infty \mu(s)\,\mathrm{d}s < 1 \text{ and } \int_0^\infty s\mu(s)\,\mathrm{d}s < \infty,$$

then the number of Hawkes process arrivals in $(0, t]$ *is asymptotically* $(t \to \infty)$ *normally distributed. More precisely,*

$$\mathbb{P}\left(\frac{N(t) - \lambda t/(1-n)}{\sqrt{\lambda t/(1-n)^3}} \leqslant y\right) \to \Phi(y)\,,$$

where $\Phi(\cdot)$ *is the CDF of the standard normal distribution.*

For an individual who enters the system at time $t_i \in \mathbb{R}$, the rate at which they produce offspring at future times $t > t_i$ is $\mu(t - t_i)$. Say that the direct offspring of this individual comprise the *first-generation*, and their offspring comprise the

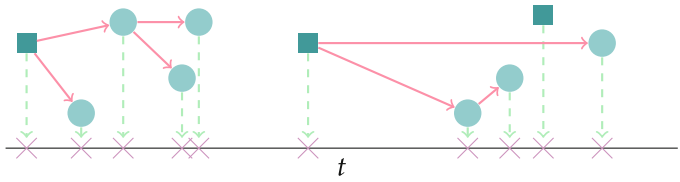

Fig. 3.2 Hawkes process represented as a collection of family trees (immigration–birth representation). Squares indicate immigrants, circles are offspring/descendants, and the crosses denote the generated point process

second-generation, and so on; members of the union of all these generations are called the *descendants* of this t_i arrival.

Using the notation from [30, Section 5.4], define Z_i to be the random number of offspring in the ith generation (with $Z_0 = 1$). As the first-generation offspring arrived from a Poisson process $Z_1 \sim \mathsf{Poi}(n)$ where the mean n is known as the *branching ratio*. This branching ratio (which can take values in $(0, \infty)$) is defined in Theorem 3.1 and in the case of an exponentially decaying intensity is

$$n = \int_0^\infty \alpha \mathrm{e}^{-\beta s} \, \mathrm{d}s = \frac{\alpha}{\beta} \,. \tag{3.6}$$

Knowledge of the branching ratio can inform development of simulation algorithms. For each immigrant i, the times of the first-generation offspring arrivals—conditioned on knowing the total number of them Z_1—are each IID with density $\mu(t - t_i)/n$. Chapter 4 explores Hawkes process simulation methods inspired by the immigration–birth representation in more detail.

The value of n also determines whether or not the Hawkes process explodes. To see this, let $g(t) = \mathbb{E}[\lambda^*(t)]$. A renewal-type equation will be constructed for g and then its limiting value will be determined. Conditioning on the time of the first jump,

$$g(t) = \mathbb{E}\left[\lambda^*(t)\right] = \mathbb{E}\left[\lambda + \int_0^t \mu(t-s)\, \mathrm{d}N(s)\right] = \lambda + \int_0^t \mu(t-s)\, \mathbb{E}[\mathrm{d}N(s)] \,.$$

In order to calculate this expected value, start with

$$\lambda^*(s) = \lim_{h \searrow 0} \frac{\mathbb{E}[N(s+h) - N(s) \mid \mathcal{H}(s)]}{h} = \frac{\mathbb{E}[\mathrm{d}N(s) \mid \mathcal{H}(s)]}{\mathrm{d}s}$$

and take expectations (and apply the tower property)

$$g(s) = \mathbb{E}[\lambda^*(s)] = \frac{\mathbb{E}[\mathbb{E}[\mathrm{d}N(s) \mid \mathcal{H}(s)]]}{\mathrm{d}s} = \frac{\mathbb{E}[\mathrm{d}N(s)]}{\mathrm{d}s}$$

to see that

$$\mathbb{E}[\mathrm{d}N(s)] = g(s)\,\mathrm{d}s\,.$$

Therefore

$$g(t) = \lambda + \int_0^t \mu(t-s)\,g(s)\,\mathrm{d}s = \lambda + \int_0^t g(t-s)\,\mu(s)\,\mathrm{d}s\,.$$

This renewal–type equation (in convolution notation is $g = \lambda + g \star \mu$) then has different solutions according to the value of n. Asmussen [2] splits the cases into: the *defective* case ($n < 1$), the *proper* case ($n = 1$), and the *excessive* case ($n > 1$). Asmussen's Proposition 7.4 states that for the defective case

$$g(t) = \mathbb{E}[\lambda^*(t)] \to \frac{\lambda}{1-n}\,, \quad \text{as } t \to \infty\,. \tag{3.7}$$

However, in the excessive case, $\lambda^*(t) \to \infty$ exponentially quickly, and hence $N(\cdot)$ eventually explodes almost surely.

Explosion for $n > 1$ is supported by viewing the arrivals as a branching process. Since $\mathbb{E}[Z_i] = n^i$ (see Section 5.4 Lemma 2 of [30]), the expected number of descendants for one individual is

$$\mathbb{E}\left[\sum_{i=1}^{\infty} Z_i\right] = \sum_{i=1}^{\infty} \mathbb{E}[Z_i] = \sum_{i=1}^{\infty} n^i = \begin{cases} \frac{n}{1-n}, & n < 1 \\ \infty, & n \geqslant 1 \end{cases}.$$

Therefore $n \geqslant 1$ means that one immigrant would generate infinitely many descendants on average.

When $n \in (0, 1)$ the branching ratio can be interpreted as a probability. It is the ratio of the number of descendants for one immigrant, to the size of their entire family (all descendants plus the original immigrant); that is

$$\frac{\mathbb{E}\left[\sum_{i=1}^{\infty} Z_i\right]}{1 + \mathbb{E}\left[\sum_{i=1}^{\infty} Z_i\right]} = \frac{\frac{n}{1-n}}{1 + \frac{n}{1-n}} = \frac{\frac{n}{1-n}}{\frac{1}{1-n}} = n\,.$$

Therefore, any Hawkes process arrival selected at random was generated *endogenously* (a child) with probability n or *exogenously* (an immigrant) with probability $1-n$. Most properties of the Hawkes process rely on the process being *stationary*, which is another way to insist that $n \in (0, 1)$ (a rigorous definition is given in Sect. 3.5), so this is assumed hereinafter.

3.4 Markov Property for Exponential Decay

When we have the exponentially decaying excitation function $\mu(s) = \alpha e^{-\beta s}$ we have the important consequence that $\lambda^*(t)$ itself becomes an exponentially decaying function (except at jump times). To see this, imagine that we observe the first k arrivals $\{t_1, t_2, \ldots, t_k\}$ and look at a future time s where $t_k < s < T_{k+1}$:

$$\begin{aligned}
\lambda^*(s) &= \lambda + \sum_{t_i < s} \alpha e^{-\beta(s-t_i)} \\
&= \lambda + \sum_{t_i \leq t_k} \alpha e^{-\beta(s-t_k+t_k-t_i)} \\
&= \lambda + e^{-\beta(s-t_k)} \sum_{t_i \leq t_k} \alpha e^{-\beta(t_k-t_i)} \\
&= \lambda + (\lambda^*(t_k) + \alpha - \lambda)\, e^{-\beta(s-t_k)} . \qquad (3.8)
\end{aligned}$$

This form of the intensity makes it possible to evaluate $\lambda^*(s)$ or $\Lambda(s)$ over a collection of time points extremely efficiently. The practical benefits of (3.8) are so great that many applications only consider the exponentially decaying form of the Hawkes process.

3.5 Covariance and Power Spectral Densities

Hawkes processes originated from the spectral analysis of general stationary point processes. The concept of stationary is only relevant or interesting when the Hawkes process is defined on $t \in \mathbb{R}$ — as opposed to $t \geq 0$ — so we will assume this formulation for the remainder of Sect. 3.5 (see Remark 3.1). Finding the power spectral density of the Hawkes process gives access to many techniques from the spectral analysis field; for example, model fitting can be achieved by using the observed periodogram of a realisation. The power spectral density is defined in terms of the covariance density. Once again the exposition is simplified by using the shorthand that

$$dN(t) = \lim_{h \searrow 0} N(t+h) - N(t) .$$

Unfortunately the term 'stationary' has many different meanings in probability theory. In this context the Hawkes process is stationary when the jump process $(dN(t) : t \in \mathbb{R})$ — which takes values in $\{0, 1\}$ — is *weakly stationary*. This means that $\mathbb{E}[dN(t)]$ and $\mathbb{C}\mathrm{ov}(dN(t), dN(t+s))$ do not depend on t. Stationarity in this sense does not imply stationarity of $N(\cdot)$ or stationarity of the inter-arrival times [44]. One consequence of stationarity is that $\lambda^*(\cdot)$ will have a long term mean

(as given by (3.7))

$$\overline{\lambda^*} \coloneqq \mathbb{E}[\lambda^*(t)] = \frac{\mathbb{E}[\mathrm{d}N(t)]}{\mathrm{d}t} = \frac{\lambda}{1-n}\,. \tag{3.9}$$

The *(auto)covariance density* is defined, for $\tau > 0$, to be

$$R(\tau) = \mathbb{C}\mathrm{ov}\left(\frac{\mathrm{d}N(t)}{\mathrm{d}t}, \frac{\mathrm{d}N(t+\tau)}{\mathrm{d}\tau}\right).$$

Due to the symmetry of covariance, $R(-\tau) = R(\tau)$, however, $R(\cdot)$ cannot be extended to the whole of $\mathbb{R}$ because there is an atom at 0. For simple point processes $\mathbb{E}[(\mathrm{d}N(t))^2] = \mathbb{E}[\mathrm{d}N(t)]$ (since $\mathrm{d}N(t) \in \{0, 1\}$) therefore for $\tau = 0$

$$\mathbb{E}[(\mathrm{d}N(t))^2] = \mathbb{E}[\mathrm{d}N(t)] = \overline{\lambda^*}\,\mathrm{d}t\,.$$

The *complete covariance density* (complete in that its domain is all of $\mathbb{R}$) is defined as

$$R^{(c)}(\tau) = \overline{\lambda^*}\delta(\tau) + R(\tau), \tag{3.10}$$

where $\delta(\cdot)$ is the Dirac delta function.

Remark 3.3 Typically $R(0)$ is defined such that $R^{(c)}(\cdot)$ is everywhere continuous. Lewis [44, p. 357] states that strictly speaking $R^{(c)}(\cdot)$ "does not have a 'value' at $\tau = 0$". See [6, 17], and [35] for further details. ◇

The corresponding *power spectral density function* is then

$$S(\omega) \coloneqq \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathrm{e}^{-i\tau\omega} R^{(c)}(\tau)\,\mathrm{d}\tau = \frac{1}{2\pi}\left[\overline{\lambda^*} + \int_{-\infty}^{\infty} \mathrm{e}^{-i\tau\omega} R(\tau)\,\mathrm{d}\tau\right]. \tag{3.11}$$

Up to now the discussion (excluding the final value of (3.9)) has considered general stationary point processes. To apply the theory specifically to Hawkes processes we need the following result.

Theorem 3.2 (Hawkes Process Power Spectral Density) *Consider a Hawkes process with an exponentially decaying intensity with $\alpha < \beta$. The intensity process then has covariance density, for $\tau > 0$,*

$$R(\tau) = \frac{\alpha\beta\lambda(2\beta-\alpha)}{2(\beta-\alpha)^2}\mathrm{e}^{-(\beta-\alpha)\tau}\,.$$

Hence, its power spectral density is, $\forall\,\omega \in \mathbb{R}$,

$$S(\omega) = \frac{\lambda\beta}{2\pi(\beta-\alpha)}\left(1 + \frac{\alpha(2\beta-\alpha)}{(\beta-\alpha)^2+\omega^2}\right).$$

Proof *(Adapted from [35])* Consider the covariance density for $\tau \in \mathbb{R} \setminus \{0\}$:

$$R(\tau) = \mathbb{E}\left[\frac{\mathrm{d}N(t)}{\mathrm{d}t}\frac{\mathrm{d}N(t+\tau)}{\mathrm{d}\tau}\right] - \overline{\lambda^*}^2. \tag{3.12}$$

Firstly note that, via the tower property,

$$\begin{aligned}\mathbb{E}\left[\frac{\mathrm{d}N(t)}{\mathrm{d}t}\frac{\mathrm{d}N(t+\tau)}{\mathrm{d}\tau}\right] &= \mathbb{E}\left[\mathbb{E}\left[\frac{\mathrm{d}N(t)}{\mathrm{d}t}\frac{\mathrm{d}N(t+\tau)}{\mathrm{d}\tau}\,\middle|\,\mathcal{H}(t+\tau)\right]\right]\\ &= \mathbb{E}\left[\frac{\mathrm{d}N(t)}{\mathrm{d}t}\mathbb{E}\left[\frac{\mathrm{d}N(t+\tau)}{\mathrm{d}\tau}\,\middle|\,\mathcal{H}(t+\tau)\right]\right]\\ &= \mathbb{E}\left[\frac{\mathrm{d}N(t)}{\mathrm{d}t}\lambda^*(t+\tau)\right].\end{aligned}$$

Hence (3.12) can be combined with (3.1) to see that $R(\tau)$ equals

$$\mathbb{E}\left[\frac{\mathrm{d}N(t)}{\mathrm{d}t}\left(\lambda + \int_{-\infty}^{t+\tau}\mu(t+\tau-s)\,\mathrm{d}N(s)\right)\right] - \overline{\lambda^*}^2,$$

which yields

$$\begin{aligned}R(\tau) &= \overline{\lambda^*}\mu(\tau) + \int_{-\infty}^{\tau}\mu(\tau-v)R(v)\,\mathrm{d}v\\ &= \overline{\lambda^*}\mu(\tau) + \int_0^{\infty}\mu(\tau+v)R(v)\,\mathrm{d}v + \int_0^{\tau}\mu(\tau-v)R(v)\,\mathrm{d}v.\end{aligned} \tag{3.13}$$

Refer to Appendix A.2.1 for details; this is a Wiener–Hopf-type integral equation. Taking the Laplace transform of (3.13) gives

$$\mathscr{L}\left\{R(\tau)\right\}(s) = \frac{\alpha\overline{\lambda^*}(2\beta-\alpha)}{2(\beta-\alpha)(s+\beta-\alpha)}.$$

Refer to Appendix A.2.2 for details. Note that (3.6) and (3.9) supply $\overline{\lambda^*} = \beta\lambda/(\beta-\alpha)$, which implies that

$$\mathscr{L}\left\{R(\tau)\right\}(s) = \frac{\alpha\beta\lambda(2\beta-\alpha)}{2(\beta-\alpha)^2(s+\beta-\alpha)}.$$

Therefore,

$$R(\tau) = \mathscr{L}^{-1}\left\{\frac{\alpha\beta\lambda(2\beta-\alpha)}{2(\beta-\alpha)^2(s+\beta-\alpha)}\right\} = \frac{\alpha\beta\lambda(2\beta-\alpha)}{2(\beta-\alpha)^2}\mathrm{e}^{-(\beta-\alpha)\tau}\,.$$

The values of $\overline{\lambda^*}$ and $\mathscr{L}\left\{R(\tau)\right\}(s)$ are then substituted into the definition given in (3.11):

$$\begin{aligned}
S(\omega) &= \frac{1}{2\pi}\left[\overline{\lambda^*} + \int_{-\infty}^{\infty} \mathrm{e}^{-i\tau\omega} R(\tau)\,\mathrm{d}\tau\right] \\
&= \frac{1}{2\pi}\left[\overline{\lambda^*} + \int_{0}^{\infty} \mathrm{e}^{-i\tau\omega} R(\tau)\,\mathrm{d}\tau + \int_{0}^{\infty} \mathrm{e}^{i\tau\omega} R(\tau)\,\mathrm{d}\tau\right] \\
&= \frac{1}{2\pi}\left[\overline{\lambda^*} + \mathscr{L}\left\{R(\tau)\right\}(i\omega) + \mathscr{L}\left\{R(\tau)\right\}(-i\omega)\right] \\
&= \frac{1}{2\pi}\left[\overline{\lambda^*} + \frac{\alpha\overline{\lambda^*}(2\beta-\alpha)}{2(\beta-\alpha)(i\omega+\beta-\alpha)} + \frac{\alpha\overline{\lambda^*}(2\beta-\alpha)}{2(\beta-\alpha)(-i\omega+\beta-\alpha)}\right] \\
&= \frac{\lambda\beta}{2\pi(\beta-\alpha)}\left[1 + \frac{\alpha(2\beta-\alpha)}{(\beta-\alpha)^2+\omega^2}\right].
\end{aligned}$$

□

Remark 3.4 The power spectral density appearing in Theorem 3.2 is a shifted scaled Cauchy PDF. ◇

Remark 3.5 As $R(\cdot)$ is a real-valued symmetric function, its Fourier transform $S(\cdot)$ is also real-valued and symmetric, that is,

$$S(\omega) = \frac{1}{2\pi}\left[\overline{\lambda^*} + \int_{-\infty}^{\infty} \mathrm{e}^{-i\tau\omega} R(\tau)\,\mathrm{d}\tau\right] = \frac{1}{2\pi}\left[\overline{\lambda^*} + \int_{-\infty}^{\infty} \cos(\tau\omega) R(\tau)\,\mathrm{d}\tau\right],$$

and

$$S_+(\omega) \coloneqq S(-\omega) + S(\omega) = 2S(\omega)\,.$$

It is common that $S_+(\cdot)$ is plotted instead of $S(\cdot)$, as in Section 4.5 of [17]; this is equivalent to wrapping the negative frequencies over to the positive half-line. ◇

3.6 Generalisations

The immigration–birth representation is useful both theoretically and practically. However, it can only be used to describe *linear* Hawkes processes. Brémaud and Massoulié [11] generalised the Hawkes process to its nonlinear form:

Definition 3.2 (Nonlinear Hawkes Process) Consider a counting process with conditional intensity function of the form

$$\lambda^*(t) = \Psi\left(\int_{-\infty}^{t} \mu(t-s)\, N(\mathrm{d}s)\right),$$

where $\Psi : \mathbb{R} \to [0, \infty)$, $\mu : (0, \infty) \to \mathbb{R}$. Then $N(\cdot)$ is a *nonlinear Hawkes process*. Selecting $\Psi(x) = \lambda + x$ reduces $N(\cdot)$ to the linear Hawkes process of Definition 3.1. ◇

Modern work on nonlinear Hawkes processes is much rarer than the original linear case (for simulation see pp. 96–116 of [14], and associated theory in [67]). This is due to a combination of factors; firstly, the generalisation was introduced relatively recently, and secondly, the increased complexity frustrates even simple investigations.

Now to return to the extension mentioned earlier, that of a collection of self- and *mutually exciting* Hawkes processes. The processes being examined are collections of one-dimensional Hawkes processes which 'excite' themselves and each other.

Definition 3.3 (Mutually Exciting Hawkes Process) Consider a collection of m counting processes $\{N_1(\cdot), \ldots, N_m(\cdot)\}$ denoted $\boldsymbol{N}$. Say $\{T_{i,j} : i \in \{1, \ldots, m\}, j \in \mathbb{N}\}$ are the random arrival times for each counting process (and $t_{i,j}$ for observed arrivals). If for each $i = 1, \ldots, m$ $N_i(\cdot)$ has a conditional intensity of the form

$$\lambda_i^*(t) = \lambda_i + \sum_{j=1}^{m} \int_{-\infty}^{t} \mu_{i,j}(t-u)\ \mathrm{d}N_j(u) \tag{3.14}$$

for some $\lambda_i > 0$ and $\mu_{i,j} : (0, \infty) \to [0, \infty)$, then $\boldsymbol{N}$ is called a *mutually exciting Hawkes process*. ◇

When the excitation functions are set to be exponentially decaying, (3.14) can be written as

$$\lambda_i^*(t) = \lambda_i + \sum_{j=1}^{m} \int_{-\infty}^{t} \alpha_{i,j} \mathrm{e}^{-\beta_{i,j}(t-s)}\ \mathrm{d}N_j(s) = \lambda_i + \sum_{j=1}^{m} \sum_{t_{j,k}<t} \alpha_{i,j} \mathrm{e}^{-\beta_{i,j}(t-t_{j,k})} \tag{3.15}$$

for non-negative constants $\{\alpha_{i,j}, \beta_{i,j} : i, j = 1, \ldots, m\}$. For the examples of this process in later chapters, we assume that $\beta_{i,j} \equiv \beta_i$, that is, each process has a decay

rate which reduces the effect of past arrivals without discriminating based on which particular processes they arrived to.

We will discuss this particular mutually exciting Hawkes process in the following chapters. At different points, it is preferable to label the times of arrivals using different notations. If we need to emphasise the stream of arrivals to each process separately, then the arrivals to N_k will be denoted $\{t_1^{(k)}, t_2^{(k)}, \dots\}$. However, it is often more useful to label the times of all the arrivals mixed together as $\{t_1, t_2, \dots\}$ and then also store which process each arrival belonged to in $\{d_1, d_2, \dots\}$. As an example, the arrival times to a bivariate $m = 2$ process

$$\{t_1^{(1)} = 1, t_2^{(1)} = 2\} \text{ and } \{t_1^{(2)} = 1.5\}$$

would be equivalent to

$$\{t_1 = 1, t_2 = 1.5, t_3 = 2\} \text{ and } \{d_1 = 1, d_2 = 2, d_3 = 1\}.$$

Remark 3.6 There are models for Hawkes processes where the points themselves are multi-dimensional, for example, spatial Hawkes processes or temporo-spatial Hawkes processes [52]. One should not confuse mutually exciting Hawkes processes with these multi-dimensional Hawkes processes. ◇

Chapter 4
Simulation Methods

Simulation is an increasingly indispensable tool in probability modelling. Here, we give details of four fundamental approaches to producing realisations of Hawkes processes.

4.1 Transformation Methods

For general point processes, a simulation algorithm is suggested by the converse of the random time change theorem (given in Sect. 9.1). In essence, points of a unit rate Poisson process $\{t_1^*, t_2^*, \dots\}$ are transformed through the inverse compensator $\Lambda(\cdot)^{-1}$ of a point process defined by that compensator. The method, sometimes called the *inverse compensator method*, iteratively solves the equations

$$t_1^* = \Lambda(t_1), \quad t_{i+1}^* - t_i^* = \Lambda(t_{k+1}) - \Lambda(t_i) \tag{4.1}$$

for the points $\{t_1, t_2, \dots\}$ of the desired point process (see [29] and Algorithm 7.4.III of [19]).

For Hawkes processes, the algorithm was first suggested by Ozaki [57], who did not explicitly state any relation to the notion of time change. Instead, Ozaki focused on the following equation:

$$-\ln(1 - F^*(t)) = \int_{t_k}^{t} \lambda^*(s)\, \mathrm{d}s\,, \tag{4.2}$$

which relates the conditional CDF of the next arrival to the previous history of arrivals $\{t_1, t_2, \dots, t_k\}$ and the specified $\lambda^*(t)$. This relation means that the next arrival time T_{k+1} can easily be generated by using the inverse transform method, that is, by drawing $U \sim \mathsf{Unif}(0, 1)$ and then finding t_{k+1} by solving

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

P. J. Laub et al., *The Elements of Hawkes Processes*,
https://doi.org/10.1007/978-3-030-84639-8_4

$$
\begin{aligned}
-\ln(U) &= \int_{t_k}^{t_{k+1}} \lambda^*(s)\,\mathrm{d}s \\
&= \lambda[t_{k+1} - t_k] + \sum_{i=1}^{k} \int_{t_k}^{t_{k+1}} \mu(s - t_i)\,\mathrm{d}s\,.
\end{aligned}
\tag{4.3}
$$

Note that (4.3) has used the typical substitution of $1-U$ by U, as $1-U \sim \mathsf{Unif}(0, 1)$. This expression can be solved numerically, for example, using Newton's method [54], which entails a significant computational effort.

For the exponentially decaying excitation function $\mu(t) = \alpha\,\mathrm{e}^{-\beta t}$, (4.3) becomes

$$
\ln(U) + \lambda[t_{k+1} - t_k] - \frac{\alpha}{\beta}(\mathrm{e}^{-\beta t_{k+1}} - \mathrm{e}^{-\beta t_k}) \sum_{i=1}^{k} \mathrm{e}^{\beta t_i} = 0\,.
$$

Solving for t_{k+1} can be achieved in linear time using the recursion given in (5.7).

4.2 Exact Simulation with Exponential Decay

If we restrict attention to Hawkes processes with exponential decay, we also have the option of using *exact simulation*. As noted in Sect. 3.4, the $\lambda^*(t)$ for $t_k < t < t_{k+1}$ takes the form

$$
\lambda^*(t) = \lambda + (\lambda^*(t_k) + \alpha - \lambda)\,\mathrm{e}^{\beta t_k}\,\mathrm{e}^{-\beta t}\,. \tag{4.4}
$$

Combining this with (4.2) yields

$$
\begin{aligned}
F^*(t) &= 1 - \exp\left\{-\int_{t_k}^{t} \lambda^*(s)\,\mathrm{d}s\right\} \\
&= 1 - \exp\left\{-(t - t_k)\lambda - (\lambda^*(t_k) + \alpha - \lambda)\beta^{-1}[1 - \mathrm{e}^{-\beta(t-t_k)}]\right\}.
\end{aligned}
\tag{4.5}
$$

Instead of inverting this relationship directly, we apply the *composition method* [23, Section VI.2.3]. Define independent random variables $T_{k+1}^{(1)}$ and $T_{k+1}^{(2)}$ by

$$
\begin{aligned}
\mathbb{P}(T_{k+1}^{(1)} > t) &= \exp\left\{-(t - t_k)\lambda\right\} \\
\mathbb{P}(T_{k+1}^{(2)} > t) &= \exp\left\{-(\lambda^*(t_k) + \alpha - \lambda)\beta^{-1}[1 - \mathrm{e}^{-\beta(t-t_k)}]\right\}.
\end{aligned}
$$

Then, $\min\{T_{k+1}^{(1)}, T_{k+1}^{(2)}\}$ has the same distribution as T_{k+1}:

$$\begin{aligned}\mathbb{P}(\min\{T_{k+1}^{(1)}, T_{k+1}^{(2)}\} \leq t) &= 1 - \mathbb{P}(\min\{T_{k+1}^{(1)}, T_{k+1}^{(2)}\} > t) \\ &= 1 - \mathbb{P}(T_{k+1}^{(1)} > t)\,\mathbb{P}(T_{k+1}^{(2)} > t) \\ &= \mathbb{P}(T_{k+1} \leq t)\,.\end{aligned}$$

We can simulate these two new random variables quite easily by the inverse transform method. Given independent uniform variables $U^{(1)}$ and $U^{(2)}$, we simulate

$$\begin{aligned}T_{k+1}^{(1)} &= t_k - \lambda^{-1}\ln(U^{(1)}) \text{ and} \\ T_{k+1}^{(2)} &= t_k - \beta^{-1}\ln\left(1 + \beta(\lambda^*(t_k) + \alpha - \lambda)^{-1}\ln(U^{(2)})\right).\end{aligned}$$

In this way, we can generate the Hawkes process arrival times one by one by setting $T_{k+1} = \min\{T_{k+1}^{(1)}, T_{k+1}^{(2)}\}$, so long as we keep calculating $\lambda^*(t_1), \lambda^*(t_2), \ldots$ by (4.4).

This procedure is described by Dassios and Zhao [20], though the composition method was first applied to simulate from (4.5) by the actuarial scientist Pai [58]. This is because the interarrival times of this Hawkes process follow a kind of *Gompertz–Makeham distribution*, a distribution which was created to model the length of a person's lifespan.

4.3 Ogata's Modified Thinning Algorithm

Generating the Hawkes process realisations is akin to generating realisations from an inhomogeneous Poisson process. The standard approach to generate from an inhomogeneous Poisson process driven by intensity function $\lambda(\cdot)$ is via thinning. Formally, the procedure is described by Algorithm 1 [45]. The intuition is to generate a 'faster' homogeneous Poisson process and remove points probabilistically so that the remaining points satisfy the time-varying intensity $\lambda(\cdot)$. The first process rate M cannot be less than $\lambda(\cdot)$ over $[0, T]$.

A similar approach can be used for the Hawkes process, called *Ogata's modified thinning algorithm* [46, 54]. The conditional intensity $\lambda^*(\cdot)$ does not have an almost sure asymptotic upper bound. Although there is no fixed M that can be predetermined for all $t \in [0, T]$, the value of M can be updated *during* each simulation run after every arrival. Algorithm 2 describes the process and Fig. 4.1 shows an example of each thinning procedure.

It is typically the case that the intensity is non-increasing during periods without any arrivals. In this case, we can use the bound $\lambda^*(t) \leqslant \lambda^*(T_i^+) =: M$ for $t \in (T_i, T_{i+1}]$. Recall that T_i^+ denotes the time just after an arrival T_i. The intensity will be updated by the time T_i^+ to reflect this new arrival, for example,

Algorithm 1: Generate an inhomogeneous Poisson process by thinning

Input: T, $\lambda(\cdot)$, M; require $\lambda(\cdot) \leqslant M$ on $[0, T]$
Result: Poisson points P
begin
  $P \leftarrow [], t \leftarrow 0$
  while $t < T$ **do**
    `// Generate next candidate point`
    $t \leftarrow t + \mathsf{Exp}(M)$
    `// Keep it with some probability`
    $U \leftarrow \mathsf{Unif}(0, M)$
    if $t < T$ **and** $U \leqslant \lambda(t)$ **then**
      $P \leftarrow [P, t]$
    end
  end
  return P
end

$$\lambda^*(T_i^+) = \lambda^*(T_i) + \alpha$$

for the exponentially decaying Hawkes process. Most of the time $\lambda^*(t^+) = \lambda^*(t)$ when t does not correspond to an arrival time (and when $\mu(\cdot)$ is continuous).

Algorithm 2: Generate a Hawkes process by thinning

Input: T, $\lambda^*(\cdot)$; require $\lambda^*(\cdot)$ to be non-increasing in periods of no arrivals.
begin
  $P \leftarrow []$
  $t \leftarrow 0$
  while $t < T$ **do**
    `// Find new upper bound`
    $M \leftarrow \lambda^*(t^+)$
    `// Generate next candidate point`
    $t \leftarrow t + \mathsf{Exp}(M)$
    `// Keep it with some probability`
    $U \leftarrow \mathsf{Unif}(0, M)$
    if $t < T$ **and** $U \leqslant \lambda^*(t)$ **then**
      $P \leftarrow [P, t]$
    end
  end
  return P
end

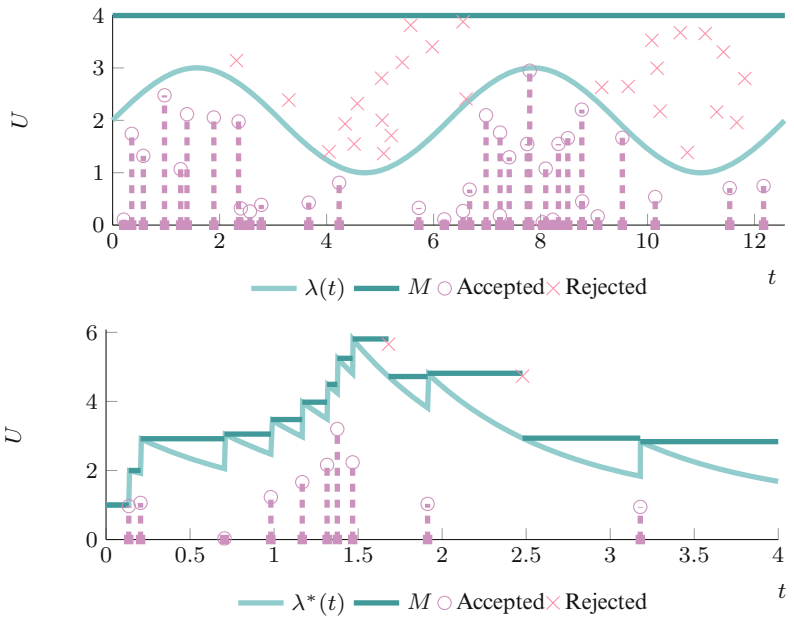

Fig. 4.1 Processes generated by thinning. (**a**) A Poisson process with intensity $\lambda(t) = 2 + \sin(t)$, bounded above by $M = 4$. (**b**) A Hawkes process with $(\lambda, \alpha, \beta) = (1, 1, 1.1)$. Each (t, U) point describes a suggested arrival at time t whose U value is given in Algorithms 1 and 2. Plus signs indicate rejected points, circles accepted, and squares the resulting point processes

4.4 Superposition of Poisson Processes

The immigration–birth representation gives rise to a simple simulation procedure: generate the immigrant arrivals and then generate the descendants for each immigrant. Algorithm 3 describes the procedure in full, with Fig. 4.2 showing an example realisation.

Immigrants form a homogeneous Poisson process of rate λ, so over an interval $[0, T]$, the number of immigrants is $\mathsf{Poi}(\lambda T)$ distributed. Conditional on knowing that there are k immigrants, their arrival times $C_1, C_2, \ldots, C_k$ are distributed as the order statistics of IID $\mathsf{Unif}(0, T)$ random variables.

Each immigrant's descendants form an inhomogeneous Poisson process. The ith immigrant's descendants arrive with intensity $\mu(t - C_i)$ for $t > C_i$. Let D_i denote the number of descendants of immigrant i. Then, $\mathbb{E}[D_i] = \int_0^\infty \mu(s)\,\mathrm{d}s = n$, and hence $D_i \overset{\text{IID}}{\sim} \mathsf{Poi}(n)$. Say that the descendants of the ith immigrant arrive at times $(C_i + E_1, C_i + E_2, \ldots, C_i + E_{D_i})$. Conditional on knowing D_i, the $\{E_j\}_{j=1}^{D_i}$ are IID random variables distributed with PDF $\mu(\cdot)/n$. For the exponentially decaying excitation function, $\mu(t) = \alpha\,\mathrm{e}^{-\beta t}$,

one may easily find that $n = \frac{\alpha}{\beta}$, so in this case we simply have that $E_j \overset{\text{IID}}{\sim} \mathsf{Exp}(\beta)$, for $j = 1, \ldots, D_i$.

Algorithm 3: Generate a Hawkes process by clusters

Input: $T, \lambda, \alpha, \beta$
begin
  $P \leftarrow \{\}$
  // Get the number of immigrants
  $k \leftarrow \mathsf{Poi}(\lambda T)$
  // Get the immigrants arrival times
  $C_1, C_2, \ldots, C_k \overset{\text{IID}}{\longleftarrow} \mathsf{Unif}(0, T)$
  // Get number of offspring for each immigrant
  $D_1, D_2, \ldots, D_k \overset{\text{IID}}{\longleftarrow} \mathsf{Poi}(\alpha/\beta)$
  // Generate their descendants
  for $i \leftarrow 1$ **to** k **do**
    // Get the descendants of immigrant i
    if $D_i > 0$ **then**
      $E_1, E_2, \ldots, E_{D_i} \overset{\text{IID}}{\longleftarrow} \mathsf{Exp}(\beta)$
      $P \leftarrow P \cup \{C_i + E_1, \ldots, C_i + E_{D_i}\}$
    end
  end
  // Remove descendants outside $[0, T]$
  $P \leftarrow \{P_i : P_i \in P, P_i \leqslant T\}$
  // Add in immigrants and sort
  $P \leftarrow \mathrm{Sort}(P \cup \{C_1, C_2, \ldots, C_k\})$
  return P
end

4.5 Mutually Exciting Hawkes Processes

This section discusses how to simulate mutually exciting Hawkes processes. Recall from Definition 3.3 that a mutually exciting Hawkes process is a collection of m counting processes $\{N_1(\cdot), \ldots, N_m(\cdot)\}$ with conditional intensities that are affected not only by self-arrivals but also by arrivals at the other processes.

While most of the algorithms above have an equivalent version in the mutually exciting case, it is the thinning algorithm which extends most naturally to this case. The key steps of Algorithm 4 to generate mutually exciting Hawkes processes are identical to the original Algorithm 2. Figure 4.3 shows an example realisation generated using this thinning method.

Bacry et al. [4] provide a similar extension of the immigration–birth simulation method (Sect. 4.4) for the mutually exciting case. Another method to simulate mutually exciting Hawkes processes is given by Dassios and Zhao [20].

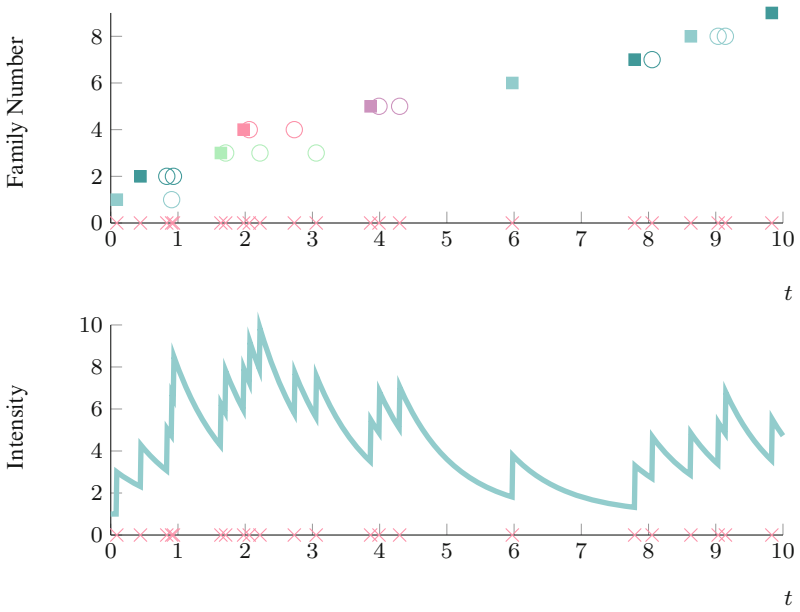

Fig. 4.2 A Hawkes Poisson process generated by clusters. Plot (**a**) shows the points generated by the immigrant–birth representation; it can be seen as a sequence of vertically stacked 'family trees'. The immigrant points are plotted as squares, and the following circles of the same height and colour are its offspring. The intensity function, with $(\lambda, \alpha, \beta) = (1, 2, 1.2)$, is plotted in (**b**). The resulting Hawkes process arrivals are drawn as crosses on the axis

Algorithm 4: Generate a mutually exciting Hawkes process by thinning

Input: $T, \boldsymbol{\lambda}(\cdot) : \mathbb{R}^m \to \mathbb{R}^+$; require $\boldsymbol{\lambda}(\cdot)$ to be non-increasing in periods of no arrivals.
begin
  $P \leftarrow []$
  $t \leftarrow 0$
  while $t < T$ **do**
    // Find new upper bound
    $M \leftarrow \sum_{k=1}^{m} \lambda_k^*(t^+)$
    // Generate next candidate point
    $t \leftarrow t + \mathsf{Exp}(M)$
    // Keep it with some probability
    $U \leftarrow \mathsf{Unif}(0, M)$
    if $t < T$ **and** $U \leqslant \sum_{k=1}^{m} \lambda_k^*(t)$ **then**
      $d \leftarrow$ the smallest k such that $U \leqslant \sum_{l=1}^{k} \lambda_l^*(t)$
      $P \leftarrow [P, (t, d)]$
    end
  end
  return P
end

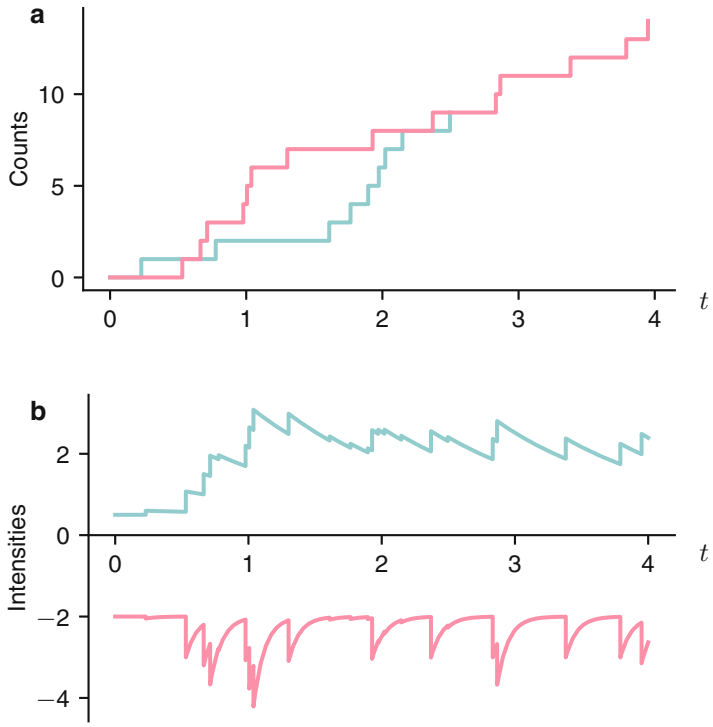

Fig. 4.3 A pair of mutually exciting Hawkes processes. (**a**) The two counting processes $N_1(t)$ and $N_2(t)$ with parameters: $\lambda_1 = 0.5$, $\lambda_2 = 2.0$, $\alpha_{1,1} = 0.1$, $\alpha_{1,2} = 0.05$, $\alpha_{2,1} = 0.5$, $\alpha_{2,2} = 1.0$, $\beta_1 = 1.0$, and $\beta_2 = 12.0$. (**b**) The processes' realised intensities. The second intensity $\lambda_2^*(t)$ has been negated to more clearly show the differences with $\lambda_1^*(t)$

Part II
Inference

Chapter 5
Maximum Likelihood Estimation

We now turn to the problem of fitting Hawkes processes to real-world data. The following chapters discuss Hawkes processes and how they fit into the common frameworks for statistical inference: maximum likelihood estimation, moment matching, the EM algorithm, and the Bayesian inference. The performance of these inference methods depends upon the excitation function $\mu(\cdot)$ chosen and the number of observations there are to fit.

The first method considered is the *maximum likelihood estimation*, which begins by finding the *likelihood function*, and estimates the model parameters as the inputs which maximise this function. This is a (frequentist) method which will produce parameter estimates of the background intensity $\widehat{\lambda}$ and for the excitation function parameters (e.g. $\widehat{\alpha}$ and $\widehat{\beta}$ for the exponential $\mu(t) = \alpha \mathrm{e}^{-\beta t}$ form), which are collected into a vector $\widehat{\boldsymbol{\theta}}$. The inference is based on some observed arrival times $\boldsymbol{t} = \{t_1, \ldots, t_{n(T)}\}$ presumed to be from a Hawkes process.

5.1 Likelihood Function

In general, likelihood functions and probability density functions are two sides of the same coin. A probability density function $f(\boldsymbol{t}; \boldsymbol{\theta})$ assumes that the model, specified by $\boldsymbol{\theta}$, is fixed and calculates the probability of seeing an observation $\boldsymbol{t}$. If the interpretation of this function is flipped, so that the observation $\boldsymbol{t}$ is fixed and the model parameter $\boldsymbol{\theta}$ is allowed to vary, then it is called a likelihood function denoted L:

$$L(\boldsymbol{\theta};\, \boldsymbol{t}) = f(\boldsymbol{t}; \boldsymbol{\theta})\,.$$

Usually, the logarithm of this function is easier to maximise, which we denote $\ell(\boldsymbol{\theta}; \boldsymbol{t}) = \ln(L(\boldsymbol{\theta}; \boldsymbol{t}))$. For brevity, we will not keep writing the observation vector

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

P. J. Laub et al., *The Elements of Hawkes Processes*,
https://doi.org/10.1007/978-3-030-84639-8_5

$\boldsymbol{t}$ and the parameter vector $\boldsymbol{\theta}$, so in our notation $L = L(\boldsymbol{\theta}; \boldsymbol{t})$, $\ell = \ell(\boldsymbol{\theta}; \boldsymbol{t})$, $\lambda^*(t) = \lambda^*(t; \boldsymbol{t}, \boldsymbol{\theta})$, and $\Lambda(t) = \Lambda(t; \boldsymbol{t}, \boldsymbol{\theta})$. The general form of the likelihood function for a large class of point processes (including Hawkes processes) is given in the following theorem. This theorem is based on Proposition 7.2.III from Daley and Vere-Jones [19].

Theorem 5.1 (Point Process Likelihood) *Let $N(\cdot)$ be a simple point process with conditional intensity $\lambda^*(\cdot)$ and compensator $\Lambda(\cdot)$. If we observe all the arrival times over the time period $[0, T]$, denoted $\{t_1, \ldots, t_{n(T)}\}$, then the likelihood function L for $N(\cdot)$ is*

$$L = \left[\prod_{i=1}^{n(T)} \lambda^*(t_i)\right] \mathrm{e}^{-\Lambda(T)} . \tag{5.1}$$

The log-likelihood ℓ is of the form

$$\ell = \sum_{i=1}^{n(T)} \ln(\lambda^*(t_i)) - \Lambda(T) . \tag{5.2}$$

Proof First, we pretend that the process is only observed up to the time of the last arrival which is at time $t_{n(T)}$—the time period $(t_{n(T)}, T]$ will be added later. The joint density function from (2.4) is

$$L = f(t_1, t_2, \ldots, t_{n(T)}) = \prod_{i=1}^{n(T)} f^*(t_i) .$$

This function can be written in terms of the conditional intensity function. Rearrange (2.5) to find $\lambda^*(t)$ in terms of $F^*(t)$ (as per [62]):

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

To be concrete, we say that $\mathcal{H}(t)$ contains k points, so $F^*(t)$ for $t \geq t_k$ describes the CDF of T_{k+1}. Integrate both sides over the interval (t_k, t):

$$-\int_{t_k}^{t} \lambda^*(s)\, \mathrm{d}s = \ln(1 - F^*(t)) - \ln(1 - F^*(t_k)) .$$

We have assumed that $N(\cdot)$ is a *simple* point process, which means multiple arrivals cannot occur at the exact same time. Therefore, $T_{k+1} \neq t_k$ and $F^*(t_k) = 0$, so

$$-\int_{t_k}^{t} \lambda^*(s)\, \mathrm{d}s = \ln(1 - F^*(t)) . \tag{5.3}$$

Further rearranging yields

$$\begin{aligned} F^*(t) &= 1 - \exp\left(-\int_{t_k}^{t} \lambda^*(s)\,\mathrm{d}s\right), \\ f^*(t) &= \lambda^*(t)\exp\left(-\int_{t_k}^{t} \lambda^*(s)\,\mathrm{d}s\right). \end{aligned} \tag{5.4}$$

Thus, the likelihood becomes

$$\begin{aligned} L &= \prod_{i=1}^{n(T)} f^*(t_i) = \prod_{i=1}^{n(T)} \lambda^*(t_i)\exp\left(-\int_{t_{i-1}}^{t_i} \lambda^*(s)\,\mathrm{d}s\right) \\ &= \left[\prod_{i=1}^{k} \lambda^*(t_i)\right]\exp\left(-\int_{0}^{t_{n(T)}} \lambda^*(s)\,\mathrm{d}s\right). \end{aligned}$$

Now suppose that the process is observed over some time period $[0, T]$ and $t_{n(T)}$ is the last arrival in this period. The likelihood will then include the probability of seeing no arrivals in the time interval $(t_{n(T)}, T]$:

$$L = \left[\prod_{i=1}^{n(T)} f^*(t_i)\right](1 - F^*(T)).$$

Using the formulation of $F^*(t)$ from (5.4), then,

$$L = \left[\prod_{i=1}^{n(T)} \lambda^*(t_i)\right]\exp\left(-\int_{0}^{T} \lambda^*(s)\,\mathrm{d}s\right) = \left[\prod_{i=1}^{n(T)} \lambda^*(t_i)\right]\mathrm{e}^{-\Lambda(T)}.$$

□

Theorem 5.1 gives us a high-level view of the likelihood function. We can substitute the form of the Hawkes process conditional intensity function to get a more detailed version. First note that the integral in $\Lambda(T)$ over $[0, T]$ can be broken up into the segments

$$[0, T] = [0, t_1] \cup (t_1, t_2] \cup \cdots \cup (t_{n(T)-1}, t_{n(T)}] \cup (t_{n(T)}, T],$$

and therefore

$$\Lambda(T) = \int_{0}^{t_1} \lambda^*(s)\,\mathrm{d}s + \sum_{i=1}^{n(T)-1} \int_{t_i}^{t_{i+1}} \lambda^*(s)\,\mathrm{d}s + \int_{t_{n(T)}}^{T} \lambda^*(s)\,\mathrm{d}s\,. \tag{5.5}$$

When we substitute the form of $\lambda^*(\cdot)$ and use $M(t) \coloneqq \int_0^t \mu(s)\,\mathrm{d}s$, each term simplifies to

$$\begin{aligned}\int_{t_i}^{t_{i+1}} \lambda^*(s)\,\mathrm{d}s &= \int_{t_i}^{t_{i+1}} \Big[\lambda + \sum_{t_j<s} \mu(s-t_j)\Big]\,\mathrm{d}s \\ &= \lambda(t_{i+1}-t_i) + \sum_{j=1}^{i} \int_{t_i}^{t_{i+1}} \mu(s-t_j)\,\mathrm{d}s \\ &= \lambda(t_{i+1}-t_i) + \sum_{j=1}^{i} \big[M(t_{i+1}-t_j) - M(t_i-t_j)\big]\,.\end{aligned}$$

This simplification holds for $i = 1, \ldots n(T)-1$, and for $i = 0$ (the summation term disappears) and also for $i = n(T)$ if we abuse the notation and say that $T = t_{n(T)+1}$. So, substituting into (5.5) gives

$$\begin{aligned}\Lambda(T) &= \sum_{i=0}^{n(T)} \left\{ \lambda(t_{i+1}-t_i) + \sum_{j=1}^{i} \big[M(t_{i+1}-t_j) - M(t_i-t_j)\big] \right\} \\ &= \lambda T + \sum_{i=1}^{n(T)} \sum_{j=1}^{i} \big[M(t_{i+1}-t_j) - M(t_i-t_j)\big]\,.\end{aligned}$$

Finally, many of the terms of this double sum cancel leaving

$$\Lambda(T) = \lambda T + \sum_{i=1}^{n(T)} \big[M(T-t_i) - M(0)\big] = \lambda T + \sum_{i=1}^{n(T)} M(T-t_i)\,. \tag{5.6}$$

The log-likelihood (5.2), combined with the Hawkes form of the intensity $\lambda^*(\cdot)$ and the compensator $\Lambda(\cdot)$ from (5.6), simplifies to

$$\ell = \sum_{i=1}^{n(T)} \ln \left[\lambda + \sum_{j=1}^{i-1} \mu(t_i - t_j) \right] - \lambda T - \sum_{i=1}^{n(T)} M(T-t_i)\,.$$

5.2 Simplifications for Exponential Decay

If we are fitting a Hawkes process with an excitation function with exponential decay $\mu(t) = \alpha \mathrm{e}^{-\beta t}$ (so $M(t) = \frac{\alpha}{\beta}(1-\mathrm{e}^{-\beta t})$), then the log-likelihood function becomes

$$\ell = \sum_{i=1}^{n(T)} \ln \left[\lambda + \alpha \sum_{j=1}^{i-1} \mathrm{e}^{-\beta(t_i - t_j)} \right] - \lambda T - \frac{\alpha}{\beta} \sum_{i=1}^{n(T)} \left[1 - \mathrm{e}^{-\beta(T - t_i)} \right].$$

Unfortunately, there is no analytic form for the maximiser of this log-likelihood. If we try to numerically maximise this ℓ, the first term's double summation means we have to evaluate an $\mathcal{O}(n(T)^2)$ function many times, which quickly becomes computationally infeasible as $n(T)$ increases. Fortunately, the similar structure of the inner summations allows l to be computed with $\mathcal{O}(n(T))$ complexity [57]. For $i \in \{2, \ldots, n(T)\}$, let $A(i) = \sum_{j=1}^{i-1} \mathrm{e}^{-\beta(t_i - t_j)}$, so that

$$\begin{aligned} A(i) &= \mathrm{e}^{-\beta t_i + \beta t_{i-1}} \sum_{j=1}^{i-1} \mathrm{e}^{-\beta t_{i-1} + \beta t_j} = \mathrm{e}^{-\beta(t_i - t_{i-1})} \left(1 + \sum_{j=1}^{i-2} \mathrm{e}^{-\beta(t_{i-1} - t_j)} \right) \\ &= \mathrm{e}^{-\beta(t_i - t_{i-1})} (1 + A(i-1)) . \end{aligned} \tag{5.7}$$

With the added base case of $A(1) = 0$, l can be rewritten as

$$\ell = \sum_{i=1}^{n(T)} \ln(\lambda + \alpha A(i)) - \lambda T - \frac{\alpha}{\beta} \sum_{i=1}^{n(T)} \left[1 - \mathrm{e}^{-\beta(T - t_i)} \right]. \tag{5.8}$$

Evaluating (5.8) is now possible in $\mathcal{O}(n(T))$ time. Numerical optimisation, using the Newton-style methods, also requires the evaluation of derivatives of ℓ. Fortunately, Ozaki [57] also gives the partial derivatives and the Hessian for this log-likelihood function.[1] Of particular note is that each derivative calculation can be achieved in order $\mathcal{O}(n(T))$ complexity when a recursive approach (similar to (5.7)) is taken [54]. For an exponential excitation function, each step of the Newton-style algorithm is of order $\mathcal{O}(n(T))$ so that MLE is attractive for small or moderately sized sample sizes.

Remark 5.1 The recursion (5.7) implies that the joint process $(n(t), \lambda^*(t))$ is Markovian; see Sect. 3.4 or Remark 1.22 of [46]. ◇

5.3 Likelihood for Mutually Exciting Hawkes Processes

The extension of the MLE approach to mutually exciting Hawkes processes is surprisingly straightforward. Again, we start with the general formulation of the likelihood for a large class of point processes which includes the mutually exciting

[1] This paper's $\frac{\partial \ell}{\partial \beta}$ partial is missing a '-1' in brackets of the $\frac{1}{\beta^2}$ term.

Hawkes, see Section 4 of [54] for the bivariate case and [46] for the multivariate version.

Theorem 5.2 (Multivariate Point Process Likelihood) *Consider a collection of m simple point processes $\boldsymbol{N}(\cdot) = (N_1(\cdot), \ldots, N_m(\cdot))$ with a joint conditional intensity $\boldsymbol{\lambda}^*(\cdot) = (\lambda_1^*(\cdot), \ldots, \lambda_m^*(\cdot))$ and compensators $\boldsymbol{\Lambda}(t) = (\Lambda_1(t), \ldots, \Lambda_m(t))$. If we observe all the arrival times over the time period $[0, T]$, denoted $\{(t_1, d_1), \ldots, (t_{n(T)}, d_{n(T)})\}$, where $N(T) = \sum_{k=1}^m N_k(T)$, then the likelihood function L for $\boldsymbol{N}(\cdot)$ is*

$$L = \Big[\prod_{i=1}^{n(T)} \lambda_{d_i}^*(t_i) \Big] \mathrm{e}^{-\sum_{k=1}^m \Lambda_k(T)} . \tag{5.9}$$

The log-likelihood ℓ is of the form

$$\ell = \sum_{i=1}^{n(T)} \ln(\lambda_{d_i}^*(t_i)) - \sum_{k=1}^{m} \Lambda_k(T) . \tag{5.10}$$

As with the univariate case, when we consider exponentially decaying excitation functions, we can simplify the calculation of the likelihood. In particular, let us take the particularly simple case of a mutually exciting Hawkes process with exponentially decaying excitations; that is,

$$\lambda_k^*(t) = \lambda_k + \sum_{(t_i, d_i): t_i < t} \alpha_{d_i, k} \mathrm{e}^{-\beta_k (t - t_i)} \quad \text{for } k = 1, \ldots, m, \tag{5.11}$$

for some $\lambda_k > 0$, $\alpha_{1,k}, \ldots, \alpha_{m,k} \geq 0$, $\beta_k > 0$. In vector notation, this is

$$\boldsymbol{\lambda}^*(t) = \boldsymbol{\lambda} + \sum_{(t_i, d_i): t_i < t} \boldsymbol{\alpha}_{d_i} \mathrm{e}^{-\boldsymbol{\beta}(t - t_i)}, \tag{5.12}$$

where $\boldsymbol{\alpha}_i = (\alpha_{i,1}, \ldots, \alpha_{i,m})^\top$ for $i = 1, \ldots, m$. In other words, if $A = (\alpha_{i,j})_{i,j=1,\ldots,m}$, then $\boldsymbol{\alpha}_i$ is the ith row of the matrix A as a column vector. The compensators for this process are

$$\boldsymbol{\Lambda}(t) = \boldsymbol{\lambda} t + \sum_{(t_i, d_i): t_i < t} \frac{1}{\boldsymbol{\beta}} \boldsymbol{\alpha}_{d_i} (1 - \mathrm{e}^{-\boldsymbol{\beta}(t - t_i)}) . \tag{5.13}$$

The advantage of this model is that the intensity between arrivals is very easy to calculate. For some time t which is after arrival t_i and before t_{i+1}, we can say

$$\boldsymbol{\lambda}^*(t) = \boldsymbol{\lambda} + (\boldsymbol{\lambda}^*(t_i) + \boldsymbol{\alpha}_{d_i} - \boldsymbol{\lambda}) \mathrm{e}^{-\boldsymbol{\beta}(t - t_i)}. \tag{5.14}$$

This is effectively the vector equivalent of (3.8), which was used to show the Markovian nature of $(N(\cdot), \lambda^*(\cdot))$ for the univariate exponentially decaying case.

The log-likelihood (5.10) requires us to evaluate $\lambda^*_{d_i}(t_i)$ for $i = 1, \ldots, n(T)$. Using (5.12), this would lead to an $\mathcal{O}(n(T)^2)$ computational burden, whereas with (5.14) we can evaluate these intensities in $\mathcal{O}(n(T))$. Equation (5.14) also allows for the compensator to be evaluated more efficiently, which will be helpful later when considering the random time change theorem in Theorem 9.3.

We note that the restriction that the $\boldsymbol{\beta}$ parameter be a vector of size m is not a firm requirement. If $\boldsymbol{\beta}$ is allowed to be a matrix and many $\beta_{i,j}$ parameters are permitted (as in (3.15)), then it is still possible to achieve $\mathcal{O}(n(T))$ complexity for the log-likelihood calculation using the Ozaki-style recursions.

5.4 Discussion

Understanding of the MLE method for the Hawkes process has changed significantly over time. The general form of the log-likelihood function (5.2) was known by Rubin [63]. It was applied to the Hawkes process by Ozaki [57] who derived the improved recursive form (5.8). Ozaki also found (as noted earlier) an efficient method for calculating the derivatives and the Hessian matrix. Consistency, asymptotic normality, and efficiency of the MLE were proved by Ogata [53].

It is clear that the MLE will usually be very effective for model fitting. However, the authors [28] found that, for small samples, the estimator produces significant bias, encounters many local optima, and is highly sensitive to the selection of excitation function. The $\mathcal{O}(n(T))$ complexity can be prohibitively slow when sample sizes become large; remember that any iterative optimisation routine would calculate the likelihood function perhaps thousands of times.

This performance bottleneck is one motivation for using the GMM to perform parameter estimation. The method uses sample moments and the sample autocorrelation function, which are smoothed via a user-selected procedure. We discuss the GMM approach in Chap. 7.

Chapter 6
EM Algorithm

The challenging part of fitting Hawkes processes to arrival data is that we do not know whether some particular observation t_i represents an immigrant or a birth in the immigration–birth interpretation from Sect. 3.3. If we were in an alternate universe where we knew both the arrival times and also the 'family tree' of the arrivals (see Fig. 4.2a for an example), then inferring the parameters of the Hawkes process turns out to be relatively easy. Indeed, this is a classic situation for which the *expectation–maximisation* (EM) algorithm should be used.

The EM algorithm is a general iterative algorithm to find parameter estimates when the observations themselves do not tell the whole story [22]. In the language of EM, we say the observations are the result of some underlying unobserved states called *latent variables*. As noted above, it is the unobserved branching structure which is unobserved for Hawkes processes.

Section 6.1 describes a particular EM algorithm for Hawkes processes [65], and it is quite computationally intensive to execute. Several approximations have been proposed to alleviate this computational bottleneck, and we discuss one such approximate 'quasi-EM algorithm' in Sect. 6.2. In Sect. 6.3, we give a worked example, comparing the EM algorithm to its quasi counterpart and to the straightforward numerical maximisation of log-likelihood discussed in the previous chapter.

6.1 EM Algorithm for Hawkes Processes

Using the notation $\boldsymbol{t}_k \coloneqq (t_1, \ldots, t_k)$, we imagine that we observe $\boldsymbol{t}_{n(T)}$ the sequence of all arrival times for the point process in the interval $[0, T]$. With this, we want to infer $\boldsymbol{\theta}$ the vector of parameters for the Hawkes process which includes λ and any parameters needed to specify μ.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

P. J. Laub et al., *The Elements of Hawkes Processes*,
https://doi.org/10.1007/978-3-030-84639-8_6

The EM algorithm can be described as follows: given an initial estimate $\boldsymbol{\theta}^{(0)} \in \boldsymbol{\Theta}$, the E and M steps are cyclically applied from iteration to iteration in order to find a (local) maximiser of the log-likelihood. The algorithm is terminated once either the log-likelihood or the parameter values have converged—that is to say, once a (local) maximum is deemed to have been found. The corresponding maximiser—the final parameter vector—is taken to be 'the' MLE.

6.1.1 Complete Data Log-Likelihood

In Theorem 5.1, we calculated the likelihood of arrival t_i given the history of previous arrivals

$$\begin{aligned} f^*(t_i) &= \lambda^*(t_i) \exp\Big(- \int_{t_{i-1}}^{t_i} \lambda^*(s)\, \mathrm{d}s \Big) \\ &= \big(\lambda + \sum_{j=1}^{i-1} \mu(t_i - t_j)\big)\, \mathrm{e}^{-\lambda[t_i - t_{i-1}]} \exp\Big(- \sum_{j=1}^{n(T)} \int_{t_{i-1}}^{t_i} \mu(s - t_j)\, \mathrm{d}s \Big) . \end{aligned}$$

We needed this to construct the likelihood over the time period $[0, T]$ as

$$L(\boldsymbol{\theta}) = [1 - F^*(T)] \times \prod_{i=1}^{n(T)} f^*(t_i) ,$$

which we maximised using an iterative numerical procedure. With EM we instead condition on the hidden data $\boldsymbol{Z}$ and create a new likelihood based on $f^*(t_i \mid \boldsymbol{Z})$.

The hidden data $\boldsymbol{Z}$ encodes the branching structure of the Hawkes process. We define the variables $z_{i0} = 1$ if t_i is an immigrant (and zero otherwise), and $z_{ij} = 1$ if t_i is a child of t_j (and zero otherwise). We denote the triangular array of auxiliary variables as $\boldsymbol{Z} = \{z_{ij}, i = 1, \ldots, n(T), j = 1, \ldots, i-1\}$. It is convenient to also define

$$c_j(t) = \max_{t_i < t} z_{ij} t_i$$

to be the arrival time for the most recent child of t_i to arrive before time t and zero if there are no children. With these variables, we can derive

$$\begin{aligned} f^*(t_i \mid \boldsymbol{Z}) = &\Big\{ \lambda \mathrm{e}^{-\lambda[t_i - c_0(t_i)]} \Big\}^{z_{i0}} \\ &\times \prod_{j=1}^{i-1} \Big\{ \mu(t_i - t_j) \exp\Big(- \int_{c_j(t_i)}^{t_i} \mu(s - t_j)\, \mathrm{d}s \Big) \Big\}^{z_{ij}} , \end{aligned} \tag{6.1}$$

where the second term is identically equal to 1 when the product is empty (i.e., when $i = 1$, in which case $z_{i0} = z_{10} \equiv 1$).

Therefore, the likelihood conditional on the $\boldsymbol{Z}$ variables, called the *complete data likelihood*, is

$$L(\boldsymbol{\theta}, \boldsymbol{Z}) = \Big[\prod_{i=1}^{n(T)} f^*(t_i \mid \boldsymbol{Z})\Big] \times \mathrm{e}^{-\lambda[T - c_0(T)]} \times \prod_{i=1}^{n(T)} \exp\left(-\int_{c_i(T)}^{T} \mu(s - t_i)\,\mathrm{d}s\right). \tag{6.2}$$

Here, the second term $\mathrm{e}^{-\lambda[T-c_0(T)]}$ represents the fact that there are no immigrants arriving between time $c_0(T)$ and T, and similarly the third term represents the lack of children from immigrant i between times $c_i(T)$ and T for each i.

The *complete data log-likelihood* is the natural logarithm of (6.2). It is denoted $\ell(\boldsymbol{\theta}, \boldsymbol{Z}) \coloneqq \ln L(\boldsymbol{\theta}, \boldsymbol{Z})$ and takes the form

$$\ell(\boldsymbol{\theta}, \boldsymbol{Z}) = \sum_{i=1}^{n(T)} \ln\big(f^*(t_i \mid \boldsymbol{Z})\big) - \lambda[T - c_0(T)] - \sum_{i=1}^{n(T)} \int_{c_i(T)}^{T} \mu(s - t_i)\,\mathrm{d}s\,. \tag{6.3}$$

We substitute (6.1) into (6.3) to obtain

$$\ell(\boldsymbol{\theta}, \boldsymbol{Z}) = -\lambda T - \sum_{i=1}^{n(T)} \int_{t_i}^{T} \mu(t - t_i)\ \mathrm{d}t + \ln\lambda \sum_{i=1}^{n(T)} z_{i0} + \sum_{i=1}^{n(T)} \sum_{0<j<i} z_{ij} \ln \mu(t_i - t_j)\,.$$

We arrived at this form by using two simplifications:

$$\lambda[T - c_0(T)] + \sum_{i=1}^{n(T)} z_{i0}\lambda[t_i - c_0(t_i)] = \lambda T\,,$$

and

$$\sum_{i=1}^{n(T)} \int_{c_i(T)}^{T} \mu(s - t_i)\,\mathrm{d}s \ + \ \sum_{i=1}^{n(T)} \sum_{j=1}^{i-1} z_{ij} \int_{c_j(t_i)}^{t_i} \mu(s - t_j)\,\mathrm{d}s = \sum_{i=1}^{n(T)} \int_{t_i}^{T} \mu(s - t_i)\,\mathrm{d}s\,.$$

If the hidden variables $\boldsymbol{Z}$ were not hidden, we could fit the Hawkes process by directly maximising $\ell(\boldsymbol{\theta}, \boldsymbol{Z})$. As the $\boldsymbol{Z}$ are not observed, they are viewed as random variables, and we try to maximise $\mathbb{E}[\ell(\boldsymbol{\theta}, \boldsymbol{Z})]$. The EM algorithm splits up this task by first obtaining the triangular array $\boldsymbol{Z}^{(r)} = (z_{ij}^{(r)})$ defined by

$$z_{ij}^{(r)} \coloneqq \mathbb{E}[Z_{ij} \mid \boldsymbol{\theta}^{(r)}, \boldsymbol{t}_i] \tag{6.4}$$

in the E step. Then, the algorithm obtains an improved parameter estimate

$$\boldsymbol{\theta}^{(r+1)} \in \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \ \ell(\boldsymbol{\theta}, \boldsymbol{Z}^{(r)})$$

in the M step. Note that $\boldsymbol{\Theta}$ will depend on the choice of excitation function and its parameterisation. The EM algorithm repeats this two-step procedure until the parameter estimates converge to the MLE.

6.1.2 The E Step

To calculate the $z_{ij}^{(r)}$ values, we note that the Z_{ij} random variables are 0–1 valued (each Z_{ij} is a conditionally independent Bernoulli trial), so in fact $\mathbb{E}[Z_{ij} \mid \boldsymbol{\theta}^{(r)}, \boldsymbol{t}_i] = \mathbb{P}(Z_{ij} = 1 \mid \boldsymbol{\theta}^{(r)}, \boldsymbol{t}_i)$. One can derive

$$z_{ij}^{(r)} = \mathbb{P}(Z_{ij} = 1 \mid \boldsymbol{\theta}^{(r)}, \boldsymbol{t}_i) = \begin{cases} \frac{\mu^{(r)}(t_i - t_j)}{\lambda^{(r)} + \sum_{t_k < t_i} \mu^{(r)}(t_i - t_k)}\,, & 0 < j < i\,, \\ \frac{\lambda^{(r)}}{\lambda^{(r)} + \sum_{t_k < t_i} \mu^{(r)}(t_i - t_k)}\,, & j = 0\,. \end{cases} \tag{6.5}$$

Here, we use the notation that $\mu^{(r)}$ is the excitation function specified by the parameters in $\boldsymbol{\theta}^{(r)}$. This equation is called the law of the branching structure for Hawkes processes.

6.1.3 The M Step

The M step of the EM algorithm maximises the expected complete data log-likelihood $\mathbb{E}[\ell(\boldsymbol{\theta}, \boldsymbol{Z}) \mid \boldsymbol{\theta}^{(r)}, \boldsymbol{t}_{n(T)}]$. As we have the equivalence

$$\mathbb{E}[\ell(\boldsymbol{\theta}, \boldsymbol{Z}) \mid \boldsymbol{\theta}^{(r)}, \boldsymbol{t}_{n(T)}] = \ell(\boldsymbol{\theta}, \boldsymbol{Z}^{(r)})\,,$$

we can just maximise

$$\begin{aligned} \ell(\boldsymbol{\theta}, \boldsymbol{Z}^{(r)}) = &- \lambda T - \sum_{i=1}^{n(T)} \int_{t_i}^{T} \mu(t - t_i)\, \mathrm{d}t + \ln\lambda \sum_{i=1}^{n(T)} z_{i0}^{(r)} \\ &+ \sum_{i=1}^{n(T)} \sum_{0<j<i} z_{ij}^{(r)} \ln \mu(t_i - t_j)\,, \end{aligned} \tag{6.6}$$

with respect to $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ to obtain the next set of estimated parameters $\boldsymbol{\theta}^{(r+1)}$.

For general excitation functions, closed-form expressions for $\boldsymbol{\theta}^{(r+1)}$ are not known, and so one would typically need to resort to numerical maximisation over $\boldsymbol{\Theta}$. However, as we shall see in Sect. 6.3, an analytic expression *does* exist for the background rate $\lambda^{(r+1)}$.

6.1.4 The Algorithm

The following gives the EM algorithm for a generic Hawkes process:

Algorithm 5: The EM algorithm for Hawkes processes

begin
  Choose $\boldsymbol{\theta}^{(0)} \in \boldsymbol{\Theta}$, a small $\eta > 0$, set $a \leftarrow 1$, $r \leftarrow 0$;
  while $a > \eta$ **do**
    E step: Calculate $\boldsymbol{Z}^{(r)}$ from (6.4) using $\boldsymbol{\theta}^{(r)}$;
    M step: Find $\boldsymbol{\theta}^{(r+1)} \in \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \ell(\boldsymbol{\theta}, \boldsymbol{Z}^{(r)})$;
    Set $a \leftarrow \ell(\boldsymbol{\theta}^{(r+1)}) - \ell(\boldsymbol{\theta}^{(r)})$ and $r \leftarrow r + 1$;
  end
  return $\boldsymbol{\theta}^{(r)}$;
end

It is a good practice to run the algorithm from multiple initial values $\boldsymbol{\theta}^{(0)}$ over $\boldsymbol{\Theta}$ to gain confidence that one has found a global maximiser of the log-likelihood—that is to say, an MLE.

6.2 The Quasi-EM Algorithm

A technique to reduce the computational challenge presented by the maximisation of the M step is to change the upper limit of the integral in (6.6). Given the quantity $\int_{t_i}^{T} \mu(t - t_i)\, \mathrm{d}t$ appearing in (6.6), we see that if the upper limit of the integral is set to ∞, then the integral is independent of the exact jump times t_i for every i (and is in fact equal to the branching ratio n defined in Sect. 3.3).

For certain choices of excitation function, such as the exponentially decaying one, this approximation yields exact solutions for the M step of the EM algorithm. However, closed-form solutions may not always exist for arbitrary μ (see, e.g. [68]).

In general, the idea is to seek an approximation of the form

$$\int_0^T \lambda^*(t)\, \mathrm{d}t = \lambda T + \sum_{t_i < T} \int_{t_i}^{T} \mu(t - t_i)\, \mathrm{d}t \approx \lambda T + \widehat{c} \cdot n(T)$$

for some constant $\widehat{c}$. Viewed in this way, taking $\widehat{c}$ to be equal to $\int_{t_i}^{\infty} \mu(t - t_i)\,\mathrm{d}t \equiv \int_0^{\infty} \mu(t)\,\mathrm{d}t \equiv n$—the expected number of descendants over all time for each immigrant from the immigration–birth representation—is simply a particular choice. In what follows, we shall adopt this choice of $\widehat{c}$.

By replicating the expectation and the maximisation steps described in Sect. 6.1, one arrives at an approximate inference procedure. However, we note that the algorithm is clearly not an EM algorithm, hence our use of the terminology quasi-EM algorithm.

6.3 A Worked Example

An illustrative example that showcases how the EM algorithm and its quasi-EM counterpart can be applied when $\mu(t) = \alpha \mathrm{e}^{-\beta t}$ for $\alpha, \beta > 0$. In this situation, $\boldsymbol{\theta} = (\lambda, \alpha, \beta)$, and, conceptually, $\boldsymbol{\Theta} = \mathbb{R}_+ \times \mathbb{R}_+ \times \mathbb{R}_+$. One could additionally impose the requirement that $n \equiv \frac{\alpha}{\beta} < 1$ (to ensure non-explosiveness/stationarity of the Hawkes process), but we do not explicitly do so here. The example involves estimating the (known) parameters of a Hawkes process model using three approaches:

1. MLE: direct (numerical) maximisation of the likelihood;
2. EM: using the EM algorithm outlined in Sect. 6.1; and
3. quasi-EM: applying the method outline in Sect. 6.2.

The estimates found are then compared to the true parameter values used to simulate the set of event times.

The Hawkes process for this example has the conditional intensity

$$\lambda^*(t) = \lambda + \sum_{t_i < t} \alpha\, \mathrm{e}^{-\beta(t - t_i)}\,,$$

the time interval is $[0, T] = [0, 600]$, and the parameter values used are as follows:

$$\lambda = 1.20, \quad \alpha = 1.70, \quad \beta = 2.00\,.$$

The data used here are simulated by using Ogata's modified thinning algorithm outlined in Sect. 4.3 (see also [54] or [19, p. 271]).

Note that we have chosen $\alpha < \beta$, which is a requirement for stationarity of the Hawkes process. Additionally, $\beta^{-1} \ll T$ here (so that the approximation employed for the quasi-EM algorithm is anticipated to be reasonable).

6.3.1 The EM Algorithm

The complete data log-likelihood for a Hawkes process with exponentially decaying excitation function is given by

$$\ell(\boldsymbol{\theta}, \boldsymbol{Z}) = -\lambda T + \ln\lambda \sum_{i=1}^{n(T)} z_{i0} - \sum_{i=1}^{n(T)} \int_{t_i}^{T} \alpha \mathrm{e}^{-\beta(t-t_i)}\, \mathrm{d}t + \sum_{i=1}^{n(T)} \sum_{0<j<i} z_{ij} (\ln\alpha - \beta(t_i - t_j)) \,. \tag{6.7}$$

E Step
Calculate $\boldsymbol{Z}^{(r)}$ from (6.4) using (6.5).

M Step
The M step is carried out by maximising the conditional expected complete data log-likelihood

$$\ell(\boldsymbol{\theta}, \boldsymbol{Z}^{(r)}) = -\lambda T + \ln\lambda \sum_{i=1}^{n(T)} z_{i0}^{(r)} - \frac{\alpha}{\beta} \sum_{i=1}^{n(T)} \left(1 - \mathrm{e}^{-\beta(T-t_i)}\right) + \sum_{i=1}^{n(T)} \sum_{j=1}^{i-1} z_{ij}^{(r)} \cdot (\ln\alpha - \beta(t_i - t_j)) \,,$$

over $\boldsymbol{\theta} \in \boldsymbol{\Theta}$.

To this end, we remark that, as long as $n(T) \geqslant 2$ and none of the elements of $\boldsymbol{\theta}^{(r)}$ are zero, the conditional probabilities are non-zero, and therefore the maximiser can have no coordinate equal to zero. Moreover, the functional form is separable in λ (implying that this parameter can be maximised completely independently of (α, β)). The maximum cannot be achieved for any of the parameters tending to ∞, and therefore any maximum must truly belong to the interior of $\boldsymbol{\Theta}$.

We identify stationary points by taking the partial derivative of $\ell(\boldsymbol{\theta}, \boldsymbol{Z}^{(r)})$ with respect to λ, α, and β, respectively, and set each derivative equal to zero. This yields the following system of equations, where $\gamma_i^{(r+1)} := \mathrm{e}^{-\beta^{(r+1)}(T-t_i)}$:

$$-T + \frac{1}{\lambda^{(r+1)}} \sum_{i=1}^{n(T)} z_{i0}^{(r)} = 0 \,,$$

$$\frac{1}{\beta^{(r+1)}} \sum_{i=1}^{n(T)} (\gamma_i^{(r+1)} - 1) + \frac{1}{\alpha^{(r+1)}} \sum_{i=1}^{n(T)} \sum_{j=1}^{i-1} z_{ij}^{(r)} = 0 \,, \text{ and}$$

$$-\frac{\alpha^{(r+1)}}{(\beta^{(r+1)})^2}\sum_{i=1}^{n(T)}\big(\gamma_i^{(r+1)}-1\big)-\frac{\alpha^{(r+1)}}{\beta^{(r+1)}}\sum_{i=1}^{n(T)}(T-t_i)\gamma_i^{(r+1)}$$
$$-\sum_{i=1}^{n(T)}\sum_{j=1}^{i-1}z_{ij}^{(r)}(t_i-t_j)=0\,.$$

Rearranging yields the system of equations

$$\lambda^{(r+1)}=\frac{1}{T}\sum_{i=1}^{n(T)}z_{i0}^{(r)},$$

$$\alpha^{(r+1)}=\beta^{(r+1)}\frac{\sum_{i=1}^{n(T)}\sum_{j=1}^{i-1}z_{ij}^{(r)}}{\sum_{i=1}^{n(T)}(1-\gamma_i^{(r+1)})}, \tag{6.8}$$

$$\beta^{(r+1)}=\alpha^{(r+1)}\frac{\big[\frac{1}{\beta^{(r+1)}}\sum_{i=1}^{n(T)}(1-\gamma_i^{(r+1)})-\sum_{i=1}^{n(T)}(T-t_i)\cdot\gamma_i^{(r+1)}\big]}{\sum_{i=1}^{n(T)}\sum_{j=1}^{i-1}(t_i-t_j)z_{ij}^{(r)}}. \tag{6.9}$$

Note that we do have a closed-form solution for $\lambda^{(r+1)}$. However, in this particular case, analytic solutions do not exist for $\alpha^{(r+1)}$ and $\beta^{(r+1)}$. Observe that (6.8) and (6.9) are *simultaneous* in $(\alpha^{(r+1)}, \beta^{(r+1)})$. Thus, we can substitute $\alpha^{(r+1)}$ from (6.8) into (6.9) and use a standard root-finding algorithm to solve for $\beta^{(r+1)}$. Once this has been obtained, a solution for $\alpha^{(r+1)}$ can then be found using (6.8).

6.3.2 *Quasi-EM Algorithm*

The complete data log-likelihood function can be simplified by making the approximation described in Sect. 6.2. That is, let the upper bounds of the integrals in (6.7) be ∞ rather than the fixed horizon T. This means that for each event time t_i, the integral ranges over $[t_i, \infty)$. The resulting approximate complete data log-likelihood is given by

$$\ell^{\text{Quasi}}(\boldsymbol{\theta}, \boldsymbol{Z})=-\lambda T+\ln\lambda\sum_{i=1}^{n(T)}z_{i0}-\frac{\alpha}{\beta}n(T)$$
$$+\sum_{i=1}^{n(T)}\sum_{0<j<i}(z_{ij}\ln\alpha-\beta(t_i-t_j))\,.$$

By taking the conditional expectation of ℓ^{Quasi} with respect to the (random) $\boldsymbol{Z}$, given the current estimate $\boldsymbol{\theta}^{(r)}$ and the observed point process, and maximising this resulting conditional expectation yields

$$\lambda^{(r+1)} = \frac{\sum_{i=1}^{n(T)} z_{i0}^{(r)}}{T},$$

$$\alpha^{(r+1)} = \frac{\left[\sum_{i=1}^{n(T)} \sum_{j=1}^{i-1} z_{ij}^{(r)}\right]^2}{n(T) \sum_{i=1}^{n(T)} \sum_{j=1}^{i-1} (t_i - t_j) z_{ij}^{(r)}}, \text{ and}$$

$$\beta^{(r+1)} = \frac{\sum_{i=1}^{n(T)} \sum_{j=1}^{i-1} z_{ij}^{(r)}}{\sum_{i=1}^{n(T)} \sum_{j=1}^{i-1} (t_i - t_j) z_{ij}^{(r)}}.$$

For this particular Hawkes process model, the approximation results in analytic solutions at the M step for all three parameters.

6.3.3 Results

The simulation consists of 5,646 event times over the interval [0, 600]. Table 6.1 presents parameter estimates found for our simulated data. Figure 6.1 shows the likelihood of exact EM and the corresponding likelihood values obtained from performing the quasi-EM algorithm. Figures 6.2, 6.3, 6.4 show the profile likelihoods of exact EM against the quasi-EM algorithm.

The reported estimates computed through the MLE and the EM are identical (to three significant figures) and are close to the true values. The quasi-EM algorithm finds estimates which are larger than the MLE values, but which are also close to the true parameter values. We remark that the quasi-EM approach recovers estimates which are quite close to the MLE values in this case.

Table 6.1 Calibrated parameters using MLE, EM, and quasi-EM

Parameter	λ	α	β
Ground truth	1.20	1.70	2.00
MLE	1.383	1.659	1.944
EM	1.383	1.659	1.944
Quasi-EM	1.396	1.673	1.965

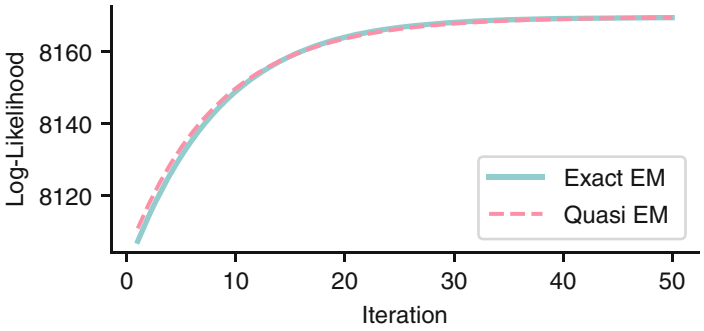

Fig. 6.1 Log-likelihood of EM and quasi-EM algorithms

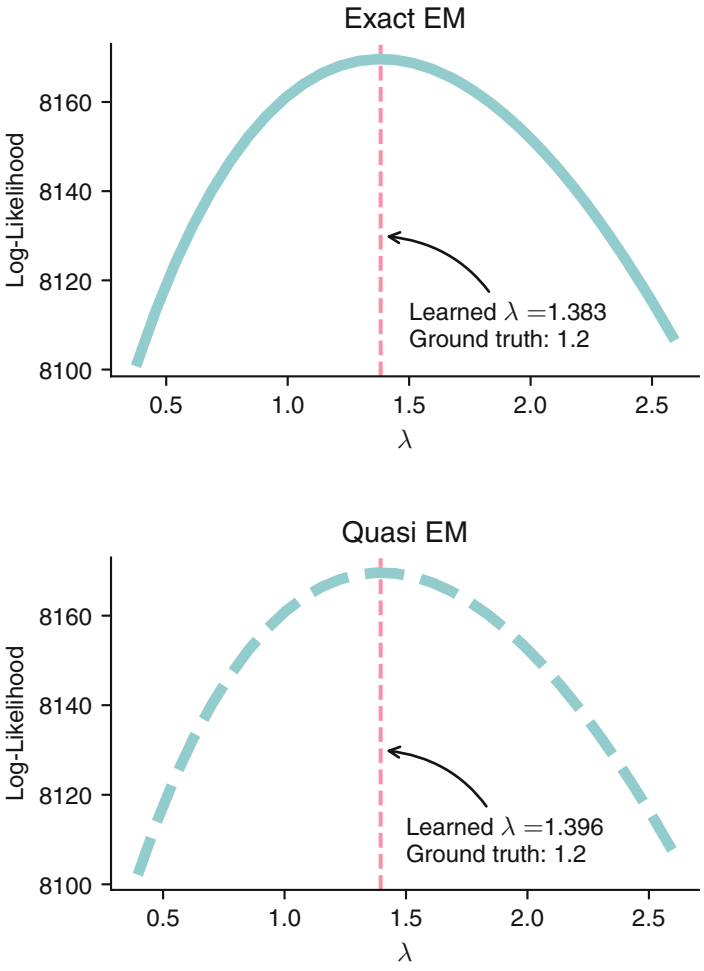

Fig. 6.2 Profile log-likelihood of EM and quasi-EM algorithms for the background rate λ

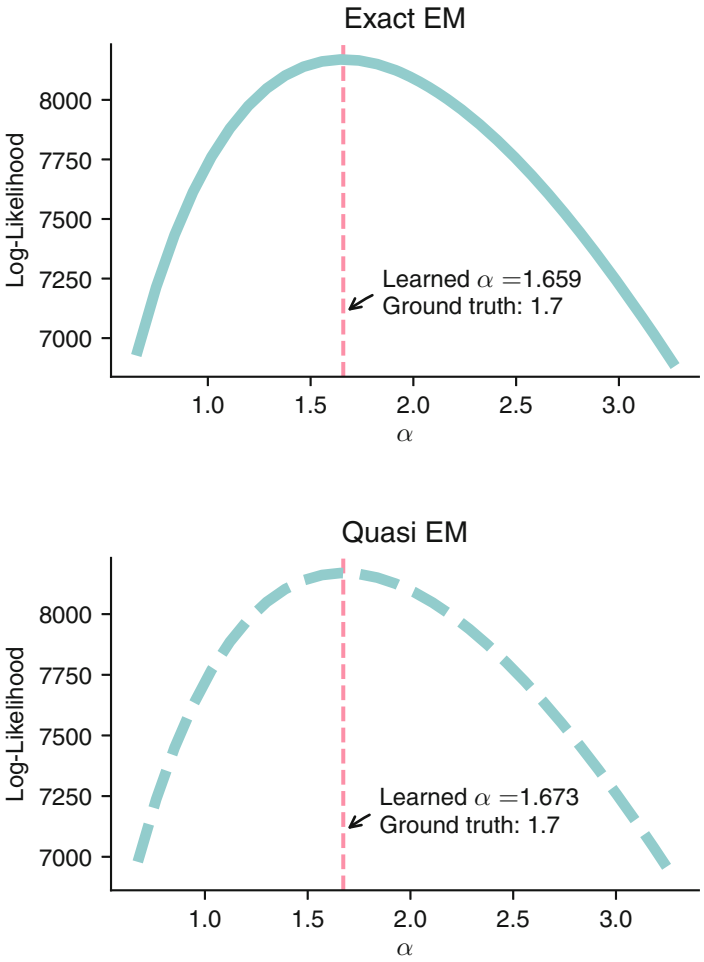

Fig. 6.3 Profile log-likelihood of EM and quasi-EM algorithms for the decay rate α

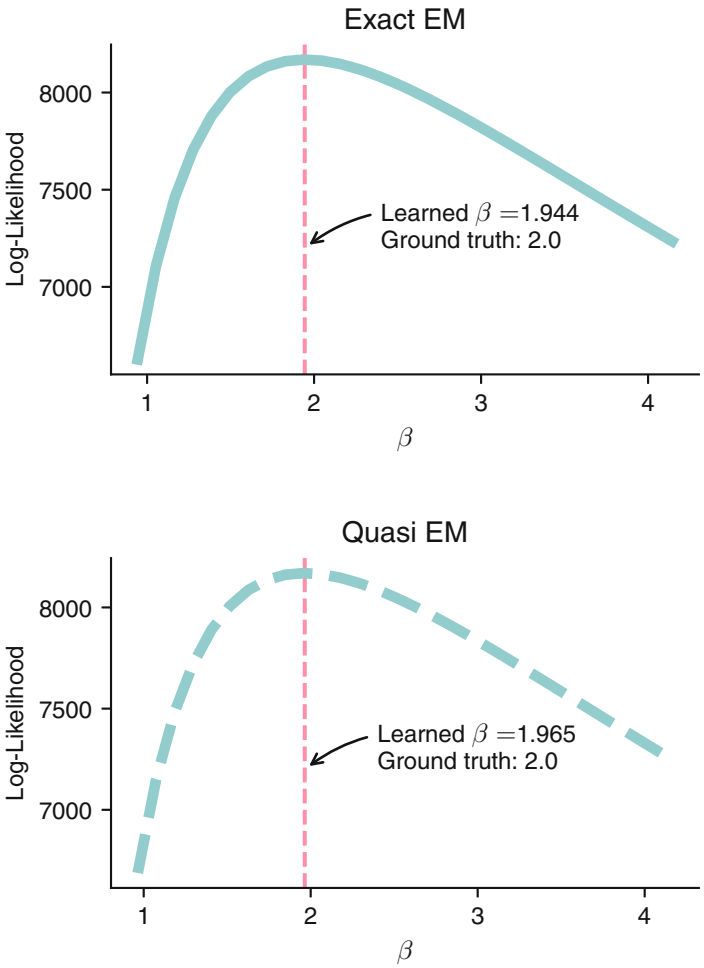

Fig. 6.4 Profile log-likelihood of EM and quasi-EM algorithms for the decay rate β

Chapter 7
Moment Matching and Interval Censored Inference

In this chapter, we focus on the problem of drawing inferences from Hawkes processes using the GMM. Differently from evaluating the maximum likelihood estimates as explained in Chap. 5, the GMM is a method for constructing estimators that uses assumptions regarding the specific moments of the random variables instead of assumptions with regard to the entirety of the distribution. These assumptions are known as moment conditions. We will first introduce the GMM and then detail how it can be used for Hawkes processes with exponential excitation function. Furthermore, we explain how to use the GMM to infer parameters of a class of generalised Hawkes processes with exponential excitation function but this time with random jump sizes.

7.1 The Generalised Method of Moments

The GMM estimation technique is a direct extension of the method of moments (MM) technique. The idea is to determine estimates of the parameters by setting sample moments to be as close as possible to their population counterparts. For the MM, there are exactly as many moment equations q as there are parameters to be estimated p and these equations match the empirical with theoretical moments. GMM generalises the MM by allowing the number of moment conditions to be either greater than or equal to the number of parameters. To describe the GMM procedure, let $\boldsymbol{\theta} \in \mathbb{R}^p$ denote the $p \times 1$ parameter vector that we wish to estimate. Suppose that we have an observed sample $\{x_i : i = 1, 2, \ldots, n\}$ where n is the sample size. Let $\boldsymbol{g}_i(\boldsymbol{\theta}) = \boldsymbol{g}(x_i, \boldsymbol{\theta})$ be a $q \times 1$ vector of functions of the data and parameters. The GMM estimator is based on a model where, for the true parameter value $\boldsymbol{\theta}_*$, the conditions

$$\mathbb{E}[\boldsymbol{g}_i(\boldsymbol{\theta}_*)] = \mathbf{0} \tag{7.1}$$

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

P. J. Laub et al., *The Elements of Hawkes Processes*,
https://doi.org/10.1007/978-3-030-84639-8_7

are satisfied. Let

$$\widehat{\boldsymbol{g}}(\boldsymbol{\theta}) := \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{g}_i(\boldsymbol{\theta}) \tag{7.2}$$

be a $q \times 1$ vector denoting the empirical moment conditions of $\boldsymbol{g}_i(\boldsymbol{\theta})$, which is the difference between an empirical and the corresponding theoretical moment. In practice, rather than solving (7.1), we seek the solution of $\widehat{\boldsymbol{g}}(\boldsymbol{\theta}) = \mathbf{0}$. We have the following cases:

(a) Underidentified: $q < p$: In this case, there are more parameters than moment conditions, and it is not possible to find a unique solution to $\widehat{\boldsymbol{g}}(\boldsymbol{\theta}) = \mathbf{0}$.
(b) Exactly identified (**MM**): $q = p$: In this case, there exists at least one solution to $\widehat{\boldsymbol{g}}(\boldsymbol{\theta}) = \mathbf{0}$. The MM estimator $\widehat{\boldsymbol{\theta}}$ can be identified by minimising the criterion function that is equivalent to the squared sum $\widehat{\boldsymbol{g}}(\boldsymbol{\theta})'\widehat{\boldsymbol{g}}(\boldsymbol{\theta})$.
(c) Overidentified (**GMM**): $q > p$: In this case, there is no unique solution to $\widehat{\boldsymbol{g}}(\boldsymbol{\theta}) = \mathbf{0}$. Let W denote any $q \times q$ positive semi-definite matrix. Define

$$J(\boldsymbol{\theta}) := \widehat{\boldsymbol{g}}(\boldsymbol{\theta})' W \widehat{\boldsymbol{g}}(\boldsymbol{\theta}). \tag{7.3}$$

Then the GMM estimator is given by

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}).$$

That is, $\widehat{\boldsymbol{\theta}}$ is the parameter that minimises $\widehat{\boldsymbol{g}}(\boldsymbol{\theta})' W \widehat{\boldsymbol{g}}(\boldsymbol{\theta})$.

If we have case (a), then there is no solution. We now proceed to examine methods (b) and (c), respectively, in the case of Hawkes processes.

7.1.1 Method of Moments

If we have p moment conditions and p parameters, we can use the MM estimator to try to find a unique solution. Consider the following empirical moment conditions:

$$\widehat{\boldsymbol{g}}(\theta) = \begin{pmatrix} \widehat{g}_1(\theta_1, \ldots, \theta_p) \\ \vdots \\ \widehat{g}_p(\theta_1, \ldots, \theta_p) \end{pmatrix} = \begin{pmatrix} \mu_1 - w_1(\boldsymbol{\theta}) \\ \vdots \\ \mu_p - w_p(\boldsymbol{\theta}) \end{pmatrix} = \mathbf{0}, \tag{7.4}$$

where $\mu_1, \ldots, \mu_p$ denote the p empirical moments and $w_1, \ldots, w_p$ be the p theoretical moments. The MM estimator can be obtained by inverting (7.4) to solve:

$$\widehat{\boldsymbol{\theta}} = [\widehat{\theta}_1, \ldots, \widehat{\theta}_p]' = \widehat{\boldsymbol{\theta}}(\mu_1, \ldots, \mu_p).$$

This can be done by minimising $\widehat{g}(\boldsymbol{\theta})' W \widehat{g}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$.

7.1.2 Generalised Method of Moments

In the case of overidentified models with $q > p$, the GMM is a two stage estimator [31], described as follows:

1. Determine the moment conditions and select the moments we need.
2. Choose the initial weight matrix $W_1 = I$, with an initial starting point $\boldsymbol{\theta}_0 = \overline{\boldsymbol{\theta}}$.
3. Using the predetermined W_1, use the well known Newton–Raphson root finding algorithm to evaluate

$$\arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

with $J(\boldsymbol{\theta})$ defined in (7.3). In order to do so, let us examine the derivatives of $L(\boldsymbol{\theta})$. We then have

$$\nabla J = \frac{\partial J}{\partial \boldsymbol{\theta}} = \begin{pmatrix} 2\widehat{\boldsymbol{g}}' W \frac{\partial \widehat{\boldsymbol{g}}}{\partial \theta_1} \\ 2\widehat{\boldsymbol{g}}' W \frac{\partial \widehat{\boldsymbol{g}}}{\partial \theta_2} \\ \vdots \\ 2\widehat{\boldsymbol{g}}' W \frac{\partial \widehat{\boldsymbol{g}}}{\partial \theta_p} \end{pmatrix}_{p\times 1}.$$

The second derivative of J is given by

$$H = \frac{\partial^2 J}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'} = \begin{pmatrix} \frac{\partial^2 J}{\partial \theta_1 \partial \theta_1} & \frac{\partial^2 J}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 J}{\partial \theta_1 \partial \theta_p} \\ \vdots & \ddots & \cdots & \vdots \\ \vdots & \cdots & \ddots & \vdots \\ \frac{\partial^2 J}{\partial \theta_p \partial \theta_1} & \frac{\partial^2 J}{\partial \theta_p \partial \theta_2} & \cdots & \frac{\partial^2 J}{\partial \theta_p \partial \theta_p} \end{pmatrix}_{p\times p} =: (H_{ij}),$$

where

$$H_{ij} = \frac{\partial^2 J}{\partial \theta_i \theta_j} = 2\left(\left(\frac{\partial \widehat{\boldsymbol{g}}}{\partial \theta_i} \right) W \frac{\partial \widehat{\boldsymbol{g}}}{\partial \theta_j} + \left(\frac{\partial^2 \widehat{\boldsymbol{g}}}{\partial \theta_i \partial \theta_j} \right)^T W \widehat{\boldsymbol{g}} \right),$$

and we have

$$\widehat{\boldsymbol{\theta}}_1 = \boldsymbol{\theta}_0 - \nabla J(\boldsymbol{\theta}_0) H^{-1}(\boldsymbol{\theta}_0).$$

4. Based on θ_1, the weight matrix is given by

$$W_2 = S^{-1}(\widehat{\boldsymbol{\theta}}_1),$$

where

$$S = \widehat{\boldsymbol{g}}(\boldsymbol{\theta}_1)\,\widehat{\boldsymbol{g}}(\boldsymbol{\theta}_1)'.$$

5. Replace W_1 by W_2 and $\boldsymbol{\theta}_0$ by $\boldsymbol{\theta}_1$.
6. If the stopping criterion is reached, exit and return $\boldsymbol{\theta}$; otherwise, go back to Step 3.

7.2 Application to Hawkes Processes

Here, we derive the long term stationary expectation of the Hawkes processes, its variance, and covariance, which will be used to aid our inference strategy. In particular, having these three theoretical expressions will enable the estimation of up to three parameters of Hawkes processes by the GMM methodology. We note that although we employ the asymptotic quantities when $t \to \infty$ as a means to *avoid* modelling the initial points of the counting process (which are usually not known), our method can indeed produce the exact expectation and higher order moments should be the initial conditions to be known or provided.

A problem frequently encountered in Hawkes processes modelling is the fact that event times are usually not known. The only available information is the number of aggregated events over a given interval. While event times are censored, we show in this chapter that it is possible to use just the aggregated counts to infer the parameters of Hawkes processes. This phenomenon is known as *interval censoring* and is illustrated by Fig. 7.1. Censoring is a phenomenon in which the value of an observation is only partially known. In our case, the times at which events happen are *censored*. Now that we have introduced the notion of interval censoring, we turn to Hawkes processes and first derive the infinitesimal generators to perform inference on interval censored data.

Recall that the Hawkes conditional intensity (see (3.2) and repeated here) is given by

$$\lambda^*(t) = \lambda + \sum_{t_i < t} \alpha \mathrm{e}^{-\beta(t-t_i)}. \tag{7.5}$$

However, in this chapter, we follow the presentation of [18] and use a slightly different formulation of the conditional intensity function:

$$\lambda^*(t) = \lambda + (\lambda_0 - \lambda)\mathrm{e}^{-\beta t} + \sum_{t_i < t} \alpha \mathrm{e}^{-\beta(t-t_i)}. \tag{7.6}$$

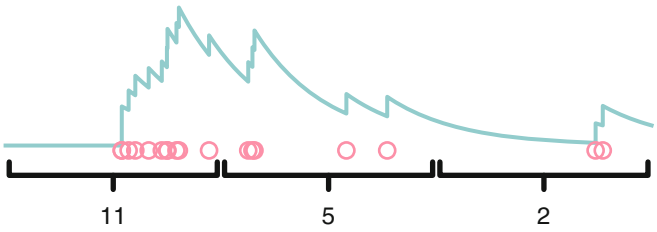

Fig. 7.1 The event times and intensity of a Hawkes process. The event times, depicted here as circles, are *not observable* (hidden). However, the aggregated number of events is *observable*. They are 11, 5, and 2 events for each interval. The times at which event occurs are censored. This is known as the interval censoring phenomenon

From (7.6), we observe that the impact of the second term on $\lambda^*(t)$ dies out exponentially as time passes. That is, as $t \to \infty$ the impact of the initial value for the conditional intensity λ_0 vanishes to zero, leaving us with

$$\lambda^*(t) \approx \lambda + \sum_{t_i < t} \alpha \mathrm{e}^{-\beta(t-t_i)}.$$

Furthermore, we define

$$\kappa = \beta - \alpha.$$

Our aim in this section is to find an explicit expression of the following quantity:

$$\mathbb{E}\left[f(t, \lambda^*(t), N(t))|\mathcal{H}(t)\right] \tag{7.7}$$

so that we are able to elicit moments such as the mean of $N(t)$ by taking $f(\cdot, \cdot, n) = n$ or the second moment of $N(t)$, by taking $f(\cdot, \cdot, n) = n^2$, say.

Theorem 7.1 *Let N be a Hawkes process with the associated conditional intensity function in (7.6). Let the operator $\mathcal{T}$ be defined as*

$$\mathcal{T} f(t, \lambda^*, n) := -\beta(\lambda^* - \lambda)\frac{\mathrm{d}f}{\mathrm{d}\lambda^*} + \lambda^* \left(f(t, \lambda^* + \alpha, n+1) - f(t, \lambda^*, n)\right).$$

Then we have the following for $0 \leqslant s < t$:

$$\begin{aligned} &\mathbb{E}[f(t, \lambda^*(t), N(t)) - f(s, \lambda^*(s), N(s))|\mathcal{H}(s)] \\ &= \mathbb{E}\left[\int_s^t \mathcal{T} f(u, \lambda^*, n)\, \mathrm{d}u \mid \mathcal{H}(s)\right]. \end{aligned} \tag{7.8}$$

◇

Proof For a proof, see [21].

The subsequent expressions are an immediate consequence of Theorem 7.1 and we see that for a given Hawkes process, the following quantities $\mathbb{E}[N^2(t) \mid \lambda^*(0)]$, $\mathbb{E}[\lambda^*(t)N(t) \mid \lambda^*(0)]$, and $\mathbb{E}[(\lambda^*(t))^2 \mid \lambda^*(0)]$ satisfy the set of ordinary differential equations (ODEs):

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[N^2(t) \mid \lambda^*(0)] = 2\mathbb{E}[\lambda^*(t)N(t) \mid \lambda^*(0)] + \mathbb{E}[\lambda^*(t) \mid \lambda^*(0)] \tag{7.9}$$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[\lambda^*(t)N(t) \mid \lambda^*(0)] &= \lambda\beta\mathbb{E}[N(t) \mid \lambda^*(0)] - \kappa\mathbb{E}[\lambda^*(t)N(t) \mid \lambda^*(0)] \\ &\quad + \mathbb{E}[(\lambda^*(t))^2 \mid \lambda^*(0)] + \alpha\mathbb{E}[\lambda^*(t) \mid \lambda^*(0)]\end{aligned} \tag{7.10}$$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[(\lambda^*(t))^2 \mid \lambda^*(0)] &= (\alpha^2 + 2\lambda\beta)\mathbb{E}[\lambda^*(t) \mid \lambda^*(0)] \\ &\quad - 2\kappa\mathbb{E}[\lambda^2(t) \mid \lambda^*(0)]\end{aligned}$$

with the initial conditions $N^2(0) = 0$, $\lambda^*(0) = \lambda_0^*$, and $\lambda^*(0)N(0) = 0$. To see why (7.9) is true, first set $f(t, \lambda, n) = n^2$ and $u(t) \coloneqq \mathbb{E}[N^2(t)|\lambda^*(0)]$. Then apply Theorem 7.1 and differentiate $u(t)$ with respect to t to arrive at the ODE with the initial condition $u_0 = N^2(0) = 0$. The other expressions can be derived in a similar manner. Figure 7.2 illustrates the dependencies between expressions in that the expression $\mathbb{E}[\lambda(t) \mid \lambda^*(0)]$ is needed for the computations of $\mathbb{E}[\lambda^2(t) \mid \lambda^*(0)]$ via (ii), which in turn is needed for $\mathbb{E}[\lambda(t)N(t) \mid \lambda^*(0)]$ through (iii). Finally, this is used

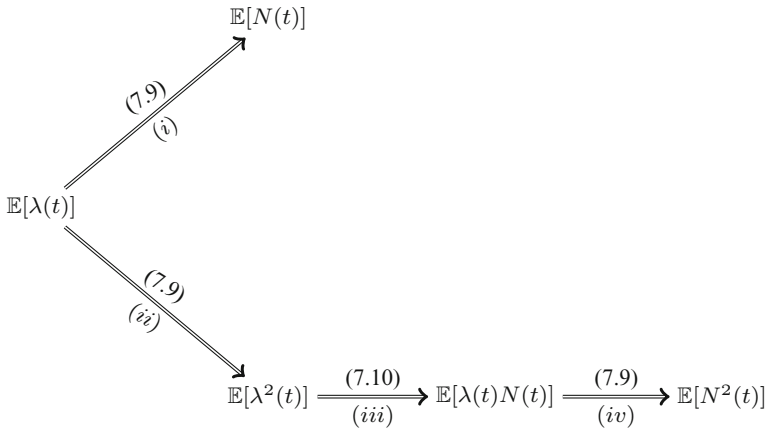

Fig. 7.2 Relationships between moment conditions. Remark that the expression $\mathbb{E}[\lambda(t)]$ is instrumental for the computations of $\mathbb{E}[\lambda^2(t)]$ via (ii), which in turn is needed for $\mathbb{E}[\lambda(t)N(t)]$ through (iii). Finally, this is used through (iv) to compute $\mathbb{E}[N^2(t)]$. The condition of $\cdot \mid \lambda^*(0)$ is suppressed for brevity

through (iv) to compute $\mathbb{E}[N^2(t) \mid \lambda^*(0)]$. These expressions will be given later in Sects. 7.2.1 and 7.2.2.

In the following, we derive some expressions that involve $\lambda^*(t)$ as well as $N(t)$ that are needed for the computations of the long term mean, variance, and covariance of the number of jumps in an interval.

7.2.1 Moments Involving $\lambda^*(t)$

Lemma 7.2 *The expectation of $\lambda^*(t)$ conditional on $\lambda^*(0)$ is given by*

$$m_1(t;\lambda_0) \coloneqq \mathbb{E}[\lambda^*(t)|\lambda^*(0)] = \frac{\lambda\beta}{\kappa} + \left(\lambda^*(0) - \frac{\lambda\beta}{\kappa}\right)\mathrm{e}^{-\kappa t}.$$

Proof Apply Theorem 7.1 and set $f(t,\lambda,n) = \lambda$ and $u(t) \coloneqq \mathbb{E}[\lambda^*(t)]$. Differentiating with respect to t we arrive at the ODE with the initial condition $u_0 = \lambda$:

$$\frac{\mathrm{d}u(t)}{\mathrm{d}t} + \kappa u(t) = \lambda\beta, \qquad u_0 = \lambda.$$

Solving this ODE and rearranging, we get

$$u_t = u_0\mathrm{e}^{-\kappa t} + \frac{\lambda\beta}{\kappa}\left(1 - \mathrm{e}^{-\kappa t}\right) = \left(u_0 - \frac{\lambda\beta}{\kappa}\right)\mathrm{e}^{-\kappa t} + \frac{\lambda\beta}{\kappa}$$

and a little algebra yields the result.

Lemma 7.3 *For a fixed t, the second moment of $\lambda^*(t)$ conditioned on $\lambda^*(0)$ is given by*

$$\begin{aligned} m_2(t;\lambda_0) &\coloneqq \mathbb{E}[(\lambda(t))^2|\lambda^*(0)] \\ &= (\lambda^*(0))^2\mathrm{e}^{-2\kappa t} + \frac{2\lambda\beta + \alpha^2}{\kappa} \cdot \left(\lambda^*(0) - \left(\frac{\lambda\beta}{\kappa}\right)\right)(\mathrm{e}^{-\kappa t} - \mathrm{e}^{-2\kappa t}) \\ &+ \frac{1}{2\kappa}\left([2\lambda\beta + \alpha^2] \cdot \frac{\lambda\beta}{\kappa}\right)(1 - \mathrm{e}^{-2\kappa t}). \end{aligned}$$

Corollary 7.1 *Under the condition that $\kappa > 0$, the asymptotic first and moments of the intensity level λ_t are given by*

$$m_1 \coloneqq \lim_{t\to\infty} \mathbb{E}[\lambda^*(t)|\lambda^*(0)] = \frac{\lambda\beta}{\kappa}$$

$$m_2 \coloneqq \lim_{t\to\infty} \mathbb{E}[(\lambda(t))^2|\lambda^*(0)] = \frac{1}{2\kappa}\left([2\lambda\beta + \alpha^2]\cdot\frac{\lambda\beta}{\kappa}\right).$$

7.2.2 Moments Involving $N(t)$ and $\lambda^*(t)$

Lemma 7.4 *The expectation of $N(t)$ conditional on $N(0) = 0$ and $\lambda^*(0)$ is given as*

$$w_1(t;\lambda^*(0)) \coloneqq \mathbb{E}\left[N(t)\,|\,\lambda^*(0)\right] = m_1 t + (\lambda^*(0) - m_1)\frac{1}{\kappa}(1 - \mathrm{e}^{-\kappa t}).$$

Lemma 7.5

$$\begin{aligned}
&\mathbb{E}[N(t)\lambda^*(t)|\lambda^*(0)] \\
&= \frac{\lambda\beta}{\kappa^2}\left[(\kappa t - 1) + \mathrm{e}^{-\kappa t}\right] + \frac{\lambda\beta(\lambda^*(0) - m_1)}{\kappa}\left[\frac{1}{\kappa} - t\mathrm{e}^{-\kappa t} - \frac{\mathrm{e}^{-\kappa t}}{\kappa}\right] \\
&+ \frac{(\lambda^*(0))^2}{\kappa}\left[\mathrm{e}^{-\kappa t} - \mathrm{e}^{-2\kappa t}\right] \\
&+ \frac{(2\lambda\beta + \alpha^2)\cdot(\lambda^*(0) - \frac{\lambda\beta}{\kappa})}{\kappa}\left[t\mathrm{e}^{-\kappa t} + \frac{1}{\kappa}\left(\mathrm{e}^{-2\kappa t} - \mathrm{e}^{-\kappa t}\right)\right] \\
&\qquad + \frac{\left[2\lambda\beta + \alpha^2\right]\cdot\frac{\lambda\beta}{\kappa}}{2\kappa^2}\left[1 + \mathrm{e}^{-2\kappa t} - 2\mathrm{e}^{-\kappa t}\right] \\
&+ \frac{\alpha\cdot m_1}{k}\left[1 - \mathrm{e}^{-\kappa t}\right] + \alpha(\lambda_0 - \mu_1)\cdot t\mathrm{e}^{-\kappa t}.
\end{aligned}$$

With these expressions at our disposal, we derive the following three quantities that will be useful in the inference or Hawkes process using the GMM.

Proposition 7.6 *The long term expectation of the number of jumps over an interval τ is given by*

$$w_1^\tau \coloneqq \lim_{t\to\infty} \mathbb{E}\left[N_{t+\tau} - N_t\,|\,\lambda^*(0)\right] = \frac{\lambda\beta}{\kappa}\tau. \tag{7.11}$$

Proposition 7.7 *The long term variance of the number of jumps over an interval τ is given by*

$$w_2^\tau \coloneqq \lim_{t\to\infty} \mathbb{E}\left[(N_{t+\tau} - N_t)^2\,|\,\lambda^*(0)\right] - \left(\mathbb{E}\left[N_{t+\tau} - N_t\,|\,\lambda^*(0)\right]\right)^2$$

$$= \frac{\lambda\beta}{\kappa}\left(\tau\frac{\beta^2}{\kappa^2} + \left(1 - \frac{\beta^2}{\kappa^2}\right)\frac{1 - \mathrm{e}^{-\kappa\tau}}{\kappa}\right). \tag{7.12}$$

Proposition 7.8 *The long term covariance of the number of jumps over an interval τ and lag δ is given by*

$$\begin{aligned} w_3^\tau &:= \lim_{t\to\infty} \mathbb{E}\left[(N(t+\tau) - N(t))(N(t+2\tau+\delta) - N(t+\tau+\delta)) \,|\, \lambda^*(0)\right] \\ &\quad - \mathbb{E}\left[(N(t+\tau) - N(t)) \,|\, \lambda^*(0)\right]\mathbb{E}\left[(N(t+2\tau+\delta) \,|\, \lambda^*(0)\right] \\ &= \frac{\lambda\beta\alpha(2\beta-\alpha)(\mathrm{e}^{-\kappa\tau}-1)^2}{2\kappa^4}\mathrm{e}^{-\kappa\delta}. \end{aligned} \tag{7.13}$$

Proof The proofs of Propositions 7.6, 7.7, and 7.8 can be found in [18]. The dependencies on moment quantities needed to derive these results are depicted in Fig. 7.2 whose expressions are given above.

7.3 Numerical Results and Discussion

7.3.1 GMM for Hawkes Model

Here we illustrate the estimation procedure by applying the techniques presented earlier to obtain an estimate for the parameters for Hawkes process model. We randomly generate using the parameters $\theta = (\lambda = 0.8, \alpha = 0.2, \beta = 1.0)$ with $\beta > \alpha$. We simulate a Hawkes process for an interval of length $T = 4000$ using the thinning algorithm explained in Sect. 4.3. In addition, we carry out the inference using the maximum likelihood estimation procedure to aid comparison to the moment matching methods. First observe that we have $q = p = 3$. The $\widehat{\boldsymbol{g}}(\boldsymbol{\theta})$ expression in (7.2) now takes the form

$$\widehat{\boldsymbol{g}}(\boldsymbol{\theta}) = \begin{pmatrix} \mu_1 - w_1^\tau \\ \mu_2 - w_2^\tau \\ \mu_3 - w_3^\tau, \end{pmatrix}$$

where μ_1, μ_2, and μ_3 denote the empirical mean, variance, and covariance of the number of events within an interval with a predetermined specification of τ and δ given in (7.14) to (7.16), respectively. The quantities w_1^τ, w_2^τ, and w_3^τ given in (7.11) to (7.13), respectively, are functions of $\boldsymbol{\theta}$. Since we are working under asymptotic expressions, wherein $t \to \infty$, we discard a number of events at the beginning of our simulated Hawkes processes. Let K_i denote the number of events falling in bin i after discarding a predetermined number of events. Also, let $n_\sharp$ be the number of bins remaining. In this case, we have

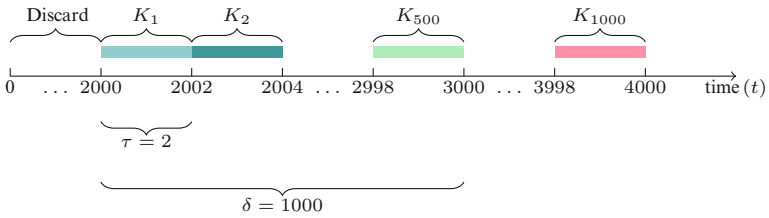

Fig. 7.3 A time line with $T = 4000$, $\tau = 2$, $\delta = 1000$, thus giving $n_\sharp = 1000$ intervals and $n_* := n_\sharp - \Delta = 500$ where $\Delta = \lceil \delta/\tau \rceil = 500$. For example K_1 denotes the number of events falling in bin 1 of length $\tau = 2$

Table 7.1 A simulation study comparing the MLE and the moment matching strategies when $q = p = 3$. For each parameter, the mean absolute error (MAE) and its corresponding standard deviation (SD) are reported. The two step iteration algorithm is averaged over 100 times. The variability of the estimates is due to the different starting points of the optimisation algorithm

	MLE			Moment matching		
	λ	α	β	λ	α	β
MAE	0.061	0.050	0.089	0.132	0.066	0.247
SD	0.039	0.152	0.014	0.175	0.041	0.171

$$\mu_1 = \frac{1}{n_\sharp} \sum_{i=1}^{n_\sharp} K_i \tag{7.14}$$

$$\mu_2 = \frac{1}{n_\sharp} \sum_{i=1}^{n_\sharp} K_i^2 - \mu_1^2 \tag{7.15}$$

$$\mu_3 = \frac{1}{n_*} \sum_{i=1}^{n_*} \left(K_i \times K_{i+\Delta} \right) - \left(\frac{1}{n_*} \sum_{i=1}^{n_*} K_i \right) \times \left(\frac{1}{n_*} \sum_{i=1}^{n_*} K_{i+\Delta} \right), \tag{7.16}$$

where $n_* := n_\sharp - \Delta$ and $\Delta = \lceil \delta/\tau \rceil$. Figure 7.3 gives an illustration of the empirical part of the moment matching strategy.

In practice, we could use the MM methodology as explained in Sect. 7.1.1, but for illustrative purposes, we apply the two step iteration algorithm explicated in Sect. 7.1.1 over 100 times, which yields the results presented in terms of mean absolute error (MAE) and its standard deviation (SD) in Table 7.1. The MLE leads to an MAE of 0.061, 0.050, and 0.089 for λ, α, and β, respectively. The GMM inference seems to work well under these parameterisations as compared to the MLEs.

Next, it is valuable for pedagogical reasons to show how the GMM works for the case $q > p$. With the same parameter settings as before, we fix $\lambda = 0.8$ and proceed

Table 7.2 A simulation study comparing the MLE and the moment matching strategies when $q = 3$, $p = 2$. Herein, we fixed $\lambda = 0.8$ but estimate two parameters α and β. For each parameter, the MAE and its corresponding SD are reported. The two step iteration algorithm is averaged over 100 replications

	MLE		Moment matching	
	α	β	α	β
MAE	0.023	0.150	0.044	0.263
SD	0.002	0.029	0.021	0.110

to estimate α and β. In this case, we have 3 equations ($q = 3$) with w_1^τ, w_2^τ, and w_3^τ given by (7.11) to (7.13), but only 2 ($p = 2$) parameters to estimate, i.e., $\theta = (\alpha, \beta)$. Table 7.2 shows the results in terms of MAE and SD over 100 replications.

One incontrovertible advantage of using the moment matching inference procedure is its speed—it is seen to be orders of magnitude faster compared to drawing inferences using the maximum likelihood estimation. Overall, the moment matching strategy performs reasonably well under these circumstances.

7.4 Inference for Generalised Hawkes

In this section, we extend the usual conditional intensity function to the following form:

$$\lambda^*(t) = \lambda + (\lambda_0 - \lambda)\mathrm{e}^{-\beta t} + \sum_{t_i < t} Y_i \mathrm{e}^{-\beta(t-t_i)}, \tag{7.17}$$

where $\{Y_i\}_{i=1,2,\ldots}$ are the sizes of jumps, which are modelled as a sequence of IID positive random variables with distribution function $G(y)$, $y > 0$ occurring at the corresponding event times $\{t_i\}_{i=1,2,\ldots}$. In this model, the jump sizes are random, as opposed to fixed at α, as in (7.5). Figure 7.4 illustrates a sample path of the conditional intensity function that is given in (7.17). As we can see, the jump sizes during event times $t_1, \ldots, t_6$ have differing values (heights) because they are drawn from a distribution with positive support.

To simplify notation, the first and second moments of Y_i are denoted by

$$\mu_{1_G} := \int_0^\infty y \, \mathrm{d}G(y), \quad \mu_{2_G} := \int_0^\infty y^2 \, \mathrm{d}G(y), \quad \widetilde{\kappa} := \beta - \mu_{1_G}.$$

The corresponding long term expectation, variance, and covariance in this setting are given as follows where the proof for these propositions follows closely that of the proof of Lemma 7.2:

Proposition 7.9 *The long term expectation of the number of jumps over an interval τ is given by*

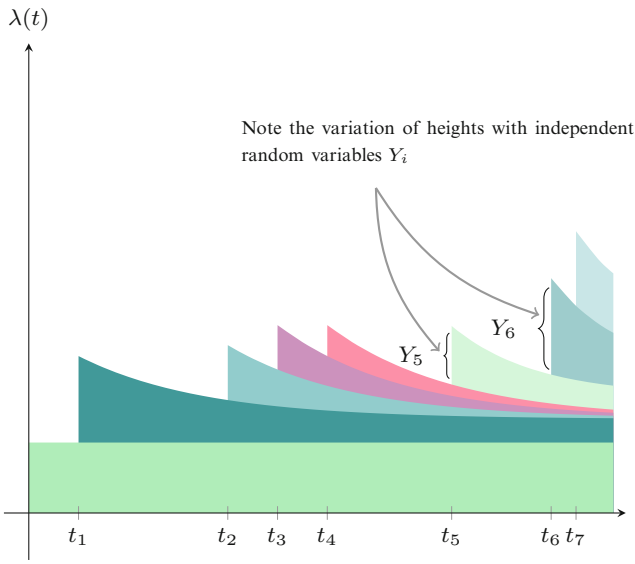

Fig. 7.4 A sample path of generalised Hawkes processes generated with conditional intensity function as in (7.17). Note that the different height of Y is due to the fact that it is drawn from a distribution with density function $G(y),\ y > 0$

$$\widetilde{w}_1^{\tau} := \lim_{t\to\infty} \mathbb{E}\left[N_{t+\tau} - N_t \mid \lambda^*(0)\right] = \frac{\lambda\beta}{\widetilde{\kappa}}\tau. \tag{7.18}$$

Proposition 7.10 *The long term variance of the number of jumps over an interval τ is given by*

$$\begin{aligned}\widetilde{w}_2^{\tau} &:= \lim_{t\to\infty} \mathbb{E}\left[(N_{t+\tau} - N_t)^2 \mid \lambda_0\right] - \left(\mathbb{E}\left[N_{t+\tau} - N_t \mid \lambda^*(0)\right]\right)^2 \\ &= \frac{\lambda\beta}{\widetilde{\kappa}}\left(\tau\frac{\beta^2}{\widetilde{\kappa}^2} + \left(1 - \frac{\beta^2}{\widetilde{\kappa}^2}\right)\frac{1 - \mathrm{e}^{-\widetilde{\kappa}\tau}}{\widetilde{\kappa}}\right).\end{aligned} \tag{7.19}$$

Proposition 7.11 *The long term covariance of the number of jumps over an interval τ and lag δ is given by*

$$\begin{aligned}\widetilde{w}_3^{\tau} &:= \lim_{t\to\infty} \mathbb{E}\left[(N(t+\tau) - N(t))(N(t+2\tau+\delta) - N(t+\tau+\delta)) \mid \lambda^*(0)\right] \\ &\quad - \mathbb{E}\left[(N(t+\tau) - N(t)) \mid \lambda_0\right]\mathbb{E}\left[(N(t+2\tau+\delta) \mid \lambda^*(0)\right] \\ &= \frac{\lambda\beta\mu_{1_G}(2\beta - \mu_{1_G})(\mathrm{e}^{-\widetilde{\kappa}\tau} - 1)^2}{2\widetilde{\kappa}^4}\mathrm{e}^{-\widetilde{\kappa}\beta}.\end{aligned} \tag{7.20}$$

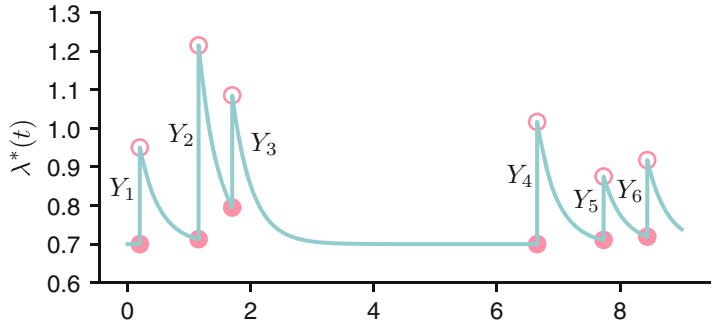

Fig. 7.5 A sample path of generalised Hawkes processes generated with conditional intensity function as in (7.17) with Y_i taking IID exponential random variables, parameterised by its mean

Table 7.3 GMM estimation for generalised Hawkes process. The jump sizes are distributed as $Y_i \sim \mathsf{Exp}(1)$. For each parameter, the MAE and its corresponding SD are reported. The two step iteration algorithm is averaged over 100 replications

	Moment matching		
	λ	b	β
MAE	0.185	0.069	0.227
SD	0.149	0.093	0.169

By a similar token, we carry out parameter inference for the parameters of generalised Hawkes process model. Here, we let the random jump sizes Y_i be exponential distributed parameterised by its mean $1/b$, with $\theta = (\lambda = 0.8,\ b = 1,\ \beta = 1.1)$ with $\beta > \mu_{1_G}$. A sample path of the conditional intensity function is given in Fig. 7.5. Applying the same calculations as before, we proceed to estimate λ, b, and β. In this case, we have 3 parameters ($p = 3$) to be estimated using the 3 equations ($q = 3$) $\widetilde{w}_1^\tau$, $\widetilde{w}_2^\tau$, and $\widetilde{w}_3^\tau$ given in (7.18) to (7.20), respectively. We remark that $\widetilde{w}_1^\tau$, $\widetilde{w}_2^\tau$, and $\widetilde{w}_3^\tau$ coincide with w_1^τ, w_2^τ, and w_3^τ, respectively, when $\widetilde{\kappa} = \kappa$, i.e. when $\mu_{1_G} = \alpha$. Table 7.3 presents the results of the GMM procedure in terms of the MAE and SD. We see that under these specifications, the moment matching strategy performs reasonably well.

Chapter 8
Bayesian Methods

In this chapter, we detail one approach for drawing inferences based on the Bayesian framework for Hawkes processes, in particular using the MCMC methodology. MCMC has been developed for the past half a decade or so and has been used widely in physics as well as in statistics and probability. MCMC methods play an important role in Bayesian statistics, especially when parameter estimation cannot be made directly, owing to the complexity of the Bayesian model; for example, when there is no closed form solution to the posterior distribution of a target parameter which we wish to estimate. MCMC allows one to sample random values from the posterior distribution and these values are subsequently used to estimate quantities of interest, such as the posterior means of model parameters. MCMC methods are typically easy and quick to implement. They also provide an alternative approach to the analysis of Bayesian models even when an analytic solution is possible.

8.1 A Primer on Bayesian Inference and MCMC

A classical model treats its unknown parameters as constants that need to be estimated, whereas a Bayesian model regards the same parameters as random variables, each of them having a prior distribution. It is assumed that the readers have a basic understanding of Bayesian methods, and hence the discussion will focus on Bayesian inference and results. The following is adapted from Chapter 6 of [41]. Here, we detail a particular MCMC algorithm, namely the Metropolis–Hastings (MH) algorithm [50]. Suppose that we wish to simulate samples from an arbitrary probability density function

$$f(x) = \frac{1}{Z}\, p(x),$$

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

P. J. Laub et al., *The Elements of Hawkes Processes*,
https://doi.org/10.1007/978-3-030-84639-8_8

where $p(x)$ is a known function and the normalising constant Z is typically unknown. Define $q(y \mid x)$ to be a *proposal* distribution which is a transition density describing the law of moving to state y from state x. Then, the MH algorithm can be summarised as in the following manner. To sample from a density $f(x)$ known up to a normalising constant, initialise with some X_0 for which $p(X_0) > 0$. Given a number of steps to take K, i.e. for $k = 0, 1, \ldots, K$, execute the following steps:

1. Given the current state X_k, we generate $Y \sim q(y \mid X_k)$.
2. Generate $U \sim \mathsf{Unif}(0, 1)$ and deliver

$$X_{k+1} = \begin{cases} Y, & \text{if } U \leqslant \mathcal{A}(X_k, Y) \\ X_k, & \text{otherwise,} \end{cases}$$

where

$$\mathcal{A}(x, y) = \min\left(\frac{f(y)q(x \mid y)}{f(x)q(y \mid x)}, 1\right).$$

The probability $\mathcal{A}(x, y)$ is known as the acceptance probability. Thus, we have the so-called MH Markov chain $X_0, X_1, \ldots, X_K$, and X_K is approximately distributed according to $f(x)$ when K is large. The discussions of convergence and how to choose K will not be discussed in the chapter, and we refer the reader to [60] for details regarding these issues. To estimate an expectation $\mathbb{E}[h(X)]$, with $X \sim f$, one can use the following estimator:

$$\frac{1}{K}\sum_{k=1}^{K} H(X_k).$$

The original Metropolis algorithm was suggested for symmetric proposal distributions, that is, for $q(y \mid x) = q(x \mid y)$. This was then generalised to allow for nonsymmetric proposals, as we have just discussed, by [32].

8.2 Bayesian Inference for Random Hawkes Processes

We closely follow the presentation of [61] but use a slightly generalised formulation of the conditional intensity function, which is given by

$$\lambda_t = a + (\lambda_0 - a)\mathrm{e}^{-\beta t} + \sum_i Y_i \mathrm{e}^{-\beta(t - t_i)}, \tag{8.1}$$

where $a > 0$, $\lambda_0 > a$, and $\{Y_i\}_{i \geqslant 1}$ is a sequence of positive IID random variables. The quantities $\{Y_i\}_{i \geqslant 1}$ are known as *contagion parameters* or simply *levels of excitation*. We further let the levels of excitation Y be of the following form:

$$\text{For each } i: \quad Y_i \sim \mathsf{Exp}(b^{-1}), \;\; f_Y(x) = \frac{1}{b}\mathrm{e}^{-\frac{1}{b}x}, \; x > 0.$$

Note that this can be extended to other positive random variables. If $\{Y_i\}_{i\geqslant 1}$ are indeed constants during all event times, say α, then we recover the classical Hawkes process [35]. Note that in the classical Hawkes process case, parameter estimation has been worked out recently in [61]. Given a set of event times $\{t_i\}_{i=1}^{n(T)}$, where $n(T)$ denotes the number of jump times, we are interested in estimating the parameters (a, b, β). To do this, we use the standard MH algorithm. Here, we fix $\lambda_0 > a$, which is assumed to be known.

8.2.1 The Likelihood

With the conditional intensity function given in (8.1), the likelihood can be written as

$$\mathcal{L}\left(\boldsymbol{T} \mid a, b, \beta, \boldsymbol{Y}\right) = \Big[\prod_{i=1}^{n(T)} \lambda(t_i)\Big]\mathrm{e}^{-\int_0^T \lambda_t \,\mathrm{d}t}$$

with T being the observation period, and $\boldsymbol{T} = \{t_1, \ldots, t_{n(T)}\}$, and $\boldsymbol{Y} = \{Y_1, \ldots, Y_{n(T)}\}$.

8.2.2 The Priors

We place the following priors for the parameters of our model:

$$a \sim \mathsf{Unif}(0, \lambda_0) \tag{8.2}$$

$$\beta \sim \mathsf{Unif}(0, 2) \tag{8.3}$$

$$b \mid \beta \sim \mathsf{Unif}(0, \beta). \tag{8.4}$$

8.2.3 The Posteriors

Differently from classical Hawkes, the levels of excitation differ across event times. Hence, we need to model the sequence $p(\boldsymbol{Y})$. The sequence $\boldsymbol{Y}$ is conditionally independent, given b, i.e.

$$Y_i \perp\!\!\!\perp Y_j \mid b, \;\; \text{for } i \neq j.$$

Using the fact that posterior $\propto$ likelihood $\times$ prior, we may write

$$\begin{aligned} p(a, b, \beta, \boldsymbol{Y} \mid \boldsymbol{T}) &\propto \mathcal{L}(\boldsymbol{T} \mid a, b, \beta, \boldsymbol{Y}) \times p(\boldsymbol{Y} \mid b) \cdot p(b \mid \beta) \cdot p(a) \cdot p(\beta) \\ &\propto \prod_{i=0}^{n(T)} \lambda_{t_i} \mathrm{e}^{-\int_0^T \lambda_t \, \mathrm{d}t} \times \prod_{i=0}^{n(T)} \frac{1}{b} \mathrm{e}^{-\frac{Y_i}{b}} \cdot \frac{1}{\beta} \cdot \frac{1}{\lambda_0} \cdot \frac{1}{2} \\ &\propto \prod_{i=0}^{n(T)} \lambda_{t_i} \mathrm{e}^{-\int_0^T \lambda_t \, \mathrm{d}t} \times \frac{1}{b^{n(T)}} \mathrm{e}^{-\frac{1}{b} \sum_{i=1}^{n(T)} Y_i} \frac{1}{\beta}, \end{aligned} \tag{8.5}$$

where $a \in (0, \lambda_0)$, $\beta \in (0, 2)$, $b \in (0, \beta)$.

The log-posterior takes the form

$$\begin{aligned} &\ln p(a, b, \beta, \boldsymbol{Y} \mid \boldsymbol{T}) \\ &= -n(T) \ln b - \frac{1}{b} \sum_{i=0}^{n(T)} Y_i + \sum_{i=0}^{n(T)} \ln(\lambda(t_i)) - \int_0^T \lambda_t \, \mathrm{d}t - \ln \beta + \text{const.}, \end{aligned}$$

where $a \in (0, \lambda_0)$, $\beta \in (0, 2)$, $b \in (0, \beta)$, and

$$\int_0^T \lambda_t \, \mathrm{d}t = aT + \frac{1}{\beta}(\lambda_0 - a)(1 - \mathrm{e}^{-\beta T}) + \frac{1}{\beta} \sum_{i=1}^{n(T)} Y_i (1 - \mathrm{e}^{-\beta(T - t_i)}).$$

8.2.4 Markov Chain Monte Carlo

The posterior derived in Eq. (8.5) is usually intractable in that it may not be of a recognisable form for us to sample from.

To see the mechanics of MH algorithm applied specifically to the Hawkes process setting, consider the following parameterisation as an illustration. Let θ_A be the parameter of interest, i.e. we would like to draw inferences on θ_A.

Suppose that $f(\theta_A)$ can only be evaluated but not directly sampled; then, we resort to the use of an MH algorithm to update θ_A. For the MH step, the candidate θ'_A is drawn from $q(\theta'_A \mid \theta_A^{(k)})$, which indicates that the current step can depend on the past draw of θ_A.

The MH step samples from $q(\theta'_A \mid \theta_A^{(k)})$, which implies that we draw $\theta_A^{(k+1)} \sim q(\theta'_A \mid \theta_A^{(k)})$ and that the criterion to accept or reject the proposal candidate is based on the acceptance probability, denoted by $\mathcal{A}(\theta_A^{(k)}, \theta'_A)$:

$$\min \left(\frac{p(\theta'_A) \, q(\theta_A^{(k)} \mid \theta'_A)}{p(\theta_A^{(k)}) \, q(\theta'_A \mid \theta_A^{(k)})}, 1 \right). \tag{8.6}$$

The MH algorithm is as follows: given $\theta_A^{(0)}$, for $k = 0, 1, ..., K$,

- sample $\theta_A^{(k+1)} \sim q(\theta'_A \mid \theta_A^{(k)})$ and *accept* or *reject* $\theta_A^{(k+1)}$ based on equation (8.6).

For a more comprehensive discussion of the subject of MCMC and other variants, we refer the reader to Chapter 6 in [41].

8.2.5 The Proposals

We choose the following symmetric proposals $q(\cdot)$ distributions for a, b, and β and an asymmetric one for Y_i for every i:

$$a' \mid a \sim \mathsf{Normal}(a, \sigma_a^2),\ b' \mid b \sim \mathsf{Normal}(b, \sigma_b^2),$$
$$\beta' \mid \beta \sim \mathsf{Normal}(\beta, \sigma_\beta^2), Y_i' \mid b \sim \mathsf{Exp}(b^{-1}).$$

8.2.6 The Acceptance Ratios

From the proposal distributions, we see that

$$\frac{q(a \mid a', \boldsymbol{Y}, b, \beta, \boldsymbol{T})}{q(a' \mid a, \boldsymbol{Y}, b, \beta, \boldsymbol{T})} = \frac{q(a \mid a')}{q(a' \mid a)} = \frac{\frac{1}{\sqrt{2\pi}\sigma_a} \exp\left(-\frac{1}{2\sigma_a^2}(a - a')^2\right)}{\frac{1}{\sqrt{2\pi}\sigma_a} \exp\left(-\frac{1}{2\sigma_a^2}(a' - a)^2\right)} = 1,$$

$$\frac{q(\boldsymbol{Y} \mid a, \boldsymbol{Y}', b, \beta, \boldsymbol{T})}{q(\boldsymbol{Y}' \mid a, \boldsymbol{Y}, b, \beta, \boldsymbol{T})} = \frac{q(\boldsymbol{Y} \mid b)}{q(\boldsymbol{Y}' \mid b)} = \frac{\prod_{i=1}^{n(T)} \frac{1}{b} \mathrm{e}^{-\frac{Y_i}{b}}}{\prod_{i=1}^{n(T)} \frac{1}{b} \mathrm{e}^{-\frac{Y_i'}{b}}}$$

$$= \exp\left(-\frac{1}{b} \sum_{i=1}^{n(T)} \left(Y_i - Y_i'\right)\right),$$

$$\frac{q(b \mid a, \boldsymbol{Y}, b', \beta, \boldsymbol{T})}{q(b' \mid a, \boldsymbol{Y}, b, \beta, \boldsymbol{T})} = 1,$$

$$\frac{q(\beta \mid a, \boldsymbol{Y}, b, \beta', \boldsymbol{T})}{q(\beta' \mid a, \boldsymbol{Y}, b, \beta, \boldsymbol{T})} = 1.$$

Hence, the acceptance ratios (omitting the minimum with 1 for the moment) are given as follows:

$$\mathcal{A}_a = \frac{p(a', \boldsymbol{Y}, b, \beta|\boldsymbol{T})}{p(a, \boldsymbol{Y}, b, \beta|\boldsymbol{T})} \frac{q(a|a', \boldsymbol{Y}, b, \beta, \boldsymbol{T})}{q(a'|a, \boldsymbol{Y}, b, \beta, \boldsymbol{T})} = \frac{p(a', \boldsymbol{Y}, b, \beta|\boldsymbol{T})}{p(a, \boldsymbol{Y}, b, \beta|\boldsymbol{T})},$$

$$\begin{aligned}\mathcal{A}_{\boldsymbol{Y}} &= \frac{p(a, \boldsymbol{Y}', b, \beta \mid \boldsymbol{T})}{p(a, \boldsymbol{Y}, b, \beta|\boldsymbol{T})} \frac{q(\boldsymbol{Y}|a, \boldsymbol{Y}', b, \beta, \boldsymbol{T})}{q(\boldsymbol{Y}' \mid a, \boldsymbol{Y}, b, \beta, \boldsymbol{T})} \\ &= \frac{p(a, \boldsymbol{Y}', b, \beta|\boldsymbol{T})}{p(a, \boldsymbol{Y}, b, \beta|\boldsymbol{T})} \mathrm{e}^{-\frac{1}{b}\sum_{i=1}^{n(T)}(Y_i' - Y_i)},\end{aligned}$$

and similar calculations can be obtained for $\mathcal{A}_b$ and $\mathcal{A}_\beta$. The criterion to accept or reject the proposal candidate for the respective candidate θ is based on the acceptance probabilities $\min(\mathcal{A}_\theta, 1)$. The algorithm for carrying out the estimation for Hawkes processes is presented in Algorithm 6.

Algorithm 6: Inference for one-dimensional Hawkes process using Metropolis–Hastings algorithm

Input: Jump times $\boldsymbol{t} = \{t_0, t_1, \ldots t_{n(T)}\}$
Result: Estimated parameters $\widehat{a}, \widehat{b}, \widehat{\beta}$
Initialise $a, b, \beta, \{Y_i\}$, set $k \leftarrow 0$, choose a burn-in BI
while $k <$ MAX-ITER **do**
  `// Sample` Y_i
  $Y_i^{\text{new}} \leftarrow \mathsf{Exp}((b^{(k)})^{-1})$
  $$Y_i^{(k+1)} \leftarrow \begin{cases} Y_i^{\text{new}} & \text{with probability } \min(\mathcal{A}_{Y_i}, 1) \\ Y_i^{(k)} & \text{otherwise} \end{cases}$$
  `// Sample` a
  $a^{\text{new}} \leftarrow \mathsf{Normal}(a^{(k)}, \sigma_a^2)$
  $$a^{(k+1)} \leftarrow \begin{cases} a^{\text{new}} & \text{with probability } \min(\mathcal{A}_a, 1) \\ a^{(k)} & \text{otherwise} \end{cases}$$
  `// Sample` b
  $b^{\text{new}} \sim \mathsf{Normal}(b^{(k)}, \sigma_b^2)$
  $$b^{(k+1)} \leftarrow \begin{cases} b^{\text{new}} & \text{with probability } \min(\mathcal{A}_b, 1) \\ b^{(k)} & \text{otherwise} \end{cases}$$
  `// Sample` β
  $\beta^{\text{new}} \sim \mathsf{Exp}((b^{(k)})^{-1})$
  $$\beta^{(k+1)} \leftarrow \begin{cases} \beta^{\text{new}} & \text{with probability } \min(\mathcal{A}_\beta, 1) \\ \beta^{(k)} & \text{otherwise} \end{cases}$$
end
$\widehat{a} = \text{mean}\{a^{(\text{BI})}, a^{(\text{BI}+1)}, \ldots a^{(\text{MAX-ITER})}\}$
$\widehat{b} = \text{mean}\{b^{(\text{BI})}, b^{(\text{BI}+1)}, \ldots b^{(\text{MAX-ITER})}\}$
$\widehat{\beta} = \text{mean}\{\beta^{(\text{BI})}, \beta^{(\text{BI}+1)}, \ldots \beta^{(\text{MAX-ITER})}\}$
Output: $\widehat{a}, \widehat{b}, \widehat{\beta}$.

8.3 Experiments

Here, we illustrate the MH algorithm estimation procedure to obtain an estimate for the parameters for the Hawkes process model. Using the thinning algorithm explained in Sect. 4.3, we simulate a Hawkes process for an interval of length $T = 1000$ using the parameters $a = 1.20$, $b = 0.70$, and $\beta = 2.00$. We remark that our priors in (8.2) to (8.4) were chosen so as to impose that the candidate is rejected if it does not respect the stationary conditions for random Hawkes processes, i.e. $a > 0$, $\beta > 0$, $0 < Y < \beta$, $0 < b < \beta$. Figure 8.1 shows the histogram of the posterior means for a, b, and β, respectively. From the samples generated using the MH algorithm, the posterior means returns $\widehat{a} = 1.17$, $\widehat{b} = 0.67$, and $\widehat{\beta} = 1.85$. The acceptance rates were found to be 62.8% for a, 55.5% for b, and 48.3% for β. These estimates confirm that the MH algorithm method used on Hawkes processes has commensurate accuracy to the ground truth and these inference procedures operate sensibly.

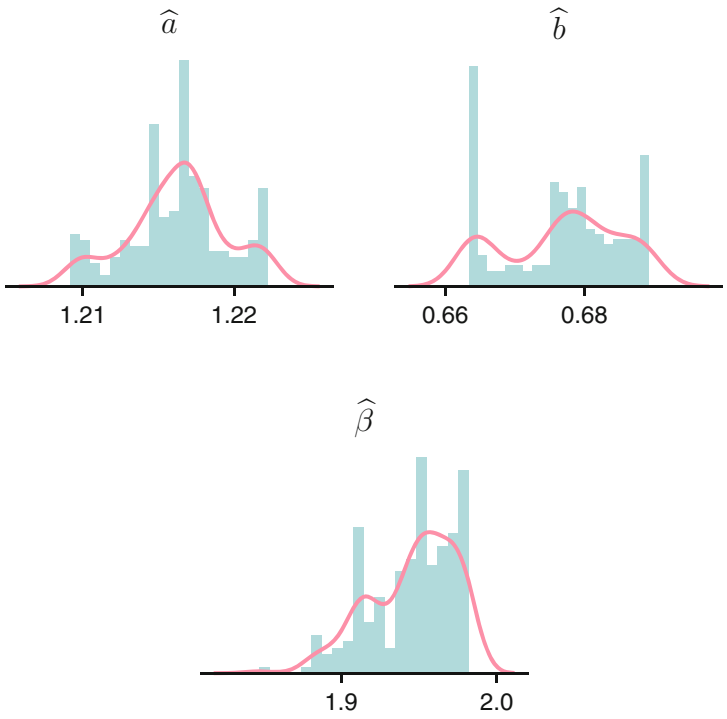

Fig. 8.1 Frequency histograms for simulated a, b, and β. Note that in these graphs, the 'peaks' point towards the true values of $a = 1.20$, $b = 0.70$, and $\beta = 2.00$, respectively

Chapter 9
Goodness of Fit

This section outlines approaches to determining the appropriateness of a Hawkes processes model for point data, which is a critical link in their application.

9.1 Transformation to a Poisson Process

Assessing the goodness of fit for some point data to a Hawkes model is an important practical consideration. In performing this assessment the point process' compensator is essential, as is the random time change theorem (here adapted from [12]):

Theorem 9.1 (Random Time Change Theorem) *Say* $\{t_1, t_2, \ldots, t_k\}$ *is a realisation over time* $[0, T]$ *from a point process with conditional intensity function* $\lambda^*(\cdot)$. *If* $\lambda^*(\cdot)$ *is positive over* $[0, T]$ *and* $\Lambda(T) < \infty$ *almost surely, then the transformed points* $\{\Lambda(t_1), \Lambda(t_2), \ldots, \Lambda(t_k)\}$ *form a Poisson process with unit rate.*

The random time change theorem is fundamental to the model fitting procedure called *(point process) residual analysis*. Original work [27] on residual analysis goes back to [51, 59], and [66]. Daley and Vere-Jones's Proposition 7.4.IV [19] rewords and extends the theorem as follows.

Theorem 9.2 (Residual Analysis) *Consider an unbounded, increasing sequence of time points* $\{t_1, t_2, \ldots\}$ *in the half-line* $(0, \infty)$, *and a monotonic, continuous compensator* $\Lambda(\cdot)$ *such that* $\lim_{t\to\infty} \Lambda(t) = \infty$ *almost surely. The transformed sequence* $\{t_1^*, t_2^*, \ldots\} = \{\Lambda(t_1), \Lambda(t_2), \ldots\}$, *whose counting process is denoted* $N^*(t)$, *is a realisation of a unit rate Poisson process if and only if the original sequence* $\{t_1, t_2, \ldots\}$ *is a realisation from the point process defined by* $\Lambda(\cdot)$.

Hence, equipped with a closed form of the compensator from (5.6), the quality of the statistical inference can be ascertained using standard fitness tests for Poisson

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

P. J. Laub et al., *The Elements of Hawkes Processes*,
https://doi.org/10.1007/978-3-030-84639-8_9

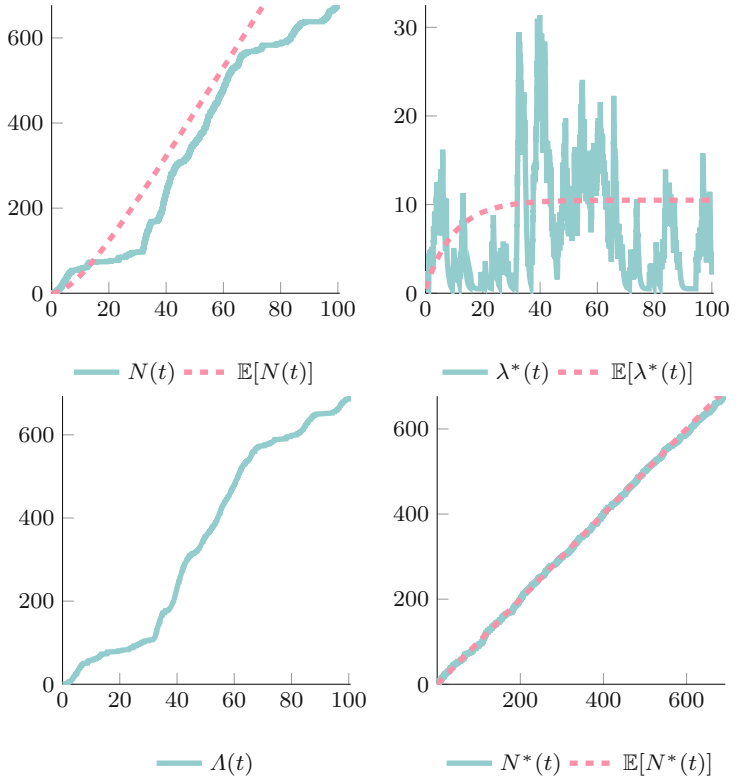

Fig. 9.1 An example of using the random time change theorem to transform a Hawkes process into a unit rate Poisson process. (**a**) A Hawkes process $N(t)$ with $(\lambda, \alpha, \beta) = (0.5, 2, 2.1)$, with the associated (**b**) conditional intensity function and (**c**) compensator. (**d**) The transformed process $N^*(t)$, where $t_i^* = \Lambda(t_i)$

processes. Figure 9.1 shows a realisation of a Hawkes process and the corresponding transformed process. In Fig. 9.1 $\Lambda(t)$ appears identical to $N(t)$. They are actually slightly different ($\Lambda(\cdot)$ is continuous); however, the similarity is expected due to Doob–Meyer decomposition of the compensator.

9.2 Tests for Poisson Process

9.2.1 Basic Tests

There are many procedures for testing whether a series of points form a Poisson process (see [17] for an extensive treatment). As a first test, one can run a hypothesis test to check $\sum_i \mathbb{I}_{\{t_i^* < t\}} \sim \mathsf{Poi}(t)$. If this initial test succeeds, then the interarrival times,

$$\{\tau_1, \tau_2, \tau_3, \dots\} = \{t_1^*, t_2^* - t_1^*, t_3^* - t_2^*, \dots\},$$

should be tested to ensure $\tau_i \overset{\text{IID}}{\sim} \mathsf{Exp}(1)$. A qualitative approach is to create a quantile–quantile (Q–Q) plot for τ_i using the exponential distribution (see for example Fig. 9.2a). Otherwise a quantitative alternative is to run Kolmogorov–Smirnov (or perhaps Anderson–Darling) tests.

9.2.2 Test for Independence

The next test, after confirming there is reason to believe that the τ_i are exponentially distributed, is to check their independence. This can be done by looking for autocorrelation in the τ_i sequence. Obviously zero autocorrelation does not imply independence, but a non-zero amount would certainly imply a non-Poisson model. A visual examination can be conducted by plotting the points (U_{i+1}, U_i). If there are noticeable patterns, then the τ_i are autocorrelated. Otherwise the points should look evenly scattered; see for example Fig. 9.2b. Quantitative extensions exist; for example see Section 3.3.3 of [40], or serial correlation tests in [41].

9.2.3 Lewis Test

A statistical test with more power is the Lewis test as described by [39]. Firstly, it relies on the fact that if $\{t_1^*, t_2^*, \dots, t_N^*\}$ are arrival times for a unit rate Poisson

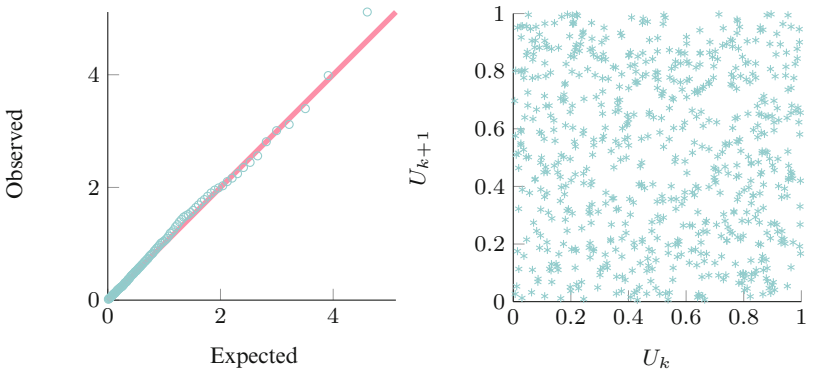

Fig. 9.2 (**a**) Q–Q testing for IID $\mathsf{Exp}(1)$ interarrival times. (**b**) A qualitative autocorrelation test. The U_k values are defined as $U_k = F(t_k^* - t_{k-1}^*) = 1 - \mathrm{e}^{-(t_k^* - t_{k-1}^*)}$

process, then $\{t_1^*/t_N^*, t_2^*/t_N^*, \ldots, t_{N-1}^*/t_N^*\}$ are distributed as the order statistics of a uniform [0, 1] random sample. This observation is called conditional uniformity and forms the basis for a test itself. Lewis' test relies on applying Durbin's modification (introduced in [24] with a widely applicable treatment by [43]).

9.2.4 Brownian Motion Approximation Test

An approximate test for Poissonity can be constructed by using the Brownian motion approximation to the Poisson process. This is to say, the observed times are transformed to be (approximately) Brownian motion, and then the known properties of Brownian motion sample paths can be used to accept or reject the original sample.

The motivation for this line of enquiry comes from Algorithm 7.4.V of [19], which is described as an 'approximate Kolmogorov–Smirnov-type test'. Unfortunately, a typographical error causes the algorithm (as printed) to produce incorrect answers for various significance levels. An alternative test based on the Brownian motion approximation is proposed here.

Say that $n(t)$ is a Poisson process of rate T. Define $M(t) = (n(t) - tT)/\sqrt{T}$ for $t \in [0, 1]$. Donsker's invariance principle implies that, as $T \to \infty$, $(M(t) : t \in [0, 1])$ converges in distribution to standard Brownian motion $(B(t) : t \in [0, 1])$. Figure 9.3 shows example realisations of $M(t)$ for various T that, at least qualitatively, are reasonable approximations to standard Brownian motion.

An alternative test is to utilise the first arcsine law for Brownian motion, which states that the random time $M^* \in [0, 1]$, given by

$$M^* = \arg \max_{s \in [0,1]} B(s)\,,$$

is arcsine distributed (that is, $M^* \sim \mathsf{Beta}(1/2, 1/2)$).

Therefore, the test takes a sequence of arrivals observed over $[0, T]$ and:

1. Transforms the arrivals to $\{t_1^*/T, t_2^*/T, \ldots, t_k^*/T\}$ that should be a Poisson process with rate T over [0, 1]
2. Constructs the Brownian motion approximation $M(t)$ as above and finds the maximiser M^*
3. Accepts the 'unit-rate Poisson process' hypothesis if M^* lies within the $(\alpha/2, 1-\alpha/2)$ quantiles of the $\mathsf{Beta}(1/2, 1/2)$ distribution; otherwise, it is rejected.

As a final note, many other tests can be performed based on other properties of Brownian motion. For example, the test could be based simply on noting that $M(1) \sim \mathsf{Normal}(0, 1)$, and thus accepts if $M(1) \in [Z_{\alpha/2}, Z_{1-\alpha/2}]$ and rejects otherwise, where Z_γ is the γ-quantile of the standard normal distribution.

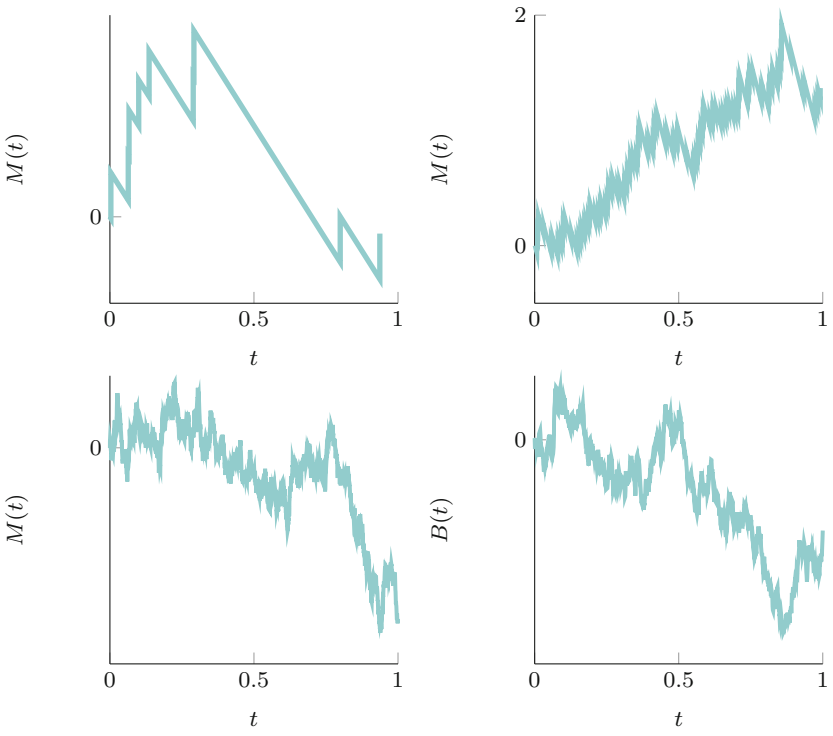

Fig. 9.3 Realisations of Poisson process approximations to Brownian motion. Plots (**a**)–(**c**) use the observed windows of T =10, 100, and 10,000, respectively. Plot (**d**) is a direct simulation of Brownian motion for comparison

9.3 Goodness of Fit for Mutually Exciting Hawkes Processes

The random time change theorem applies to the mutually exciting Hawkes process with only a minor modification, [8, 13].

Theorem 9.3 (Multidimensional Random Time Change Theorem) *Say we observe m point processes with the joint conditional intensity function* $\boldsymbol{\lambda}^*(\cdot)$ *over* $[0, T]$. *Label the points from* N_k *as* $\{(t_1^{(k)}, \ldots, t_{N_k(T)}^{(k)})\}$ *for* $k = 1, \ldots, m$. *If* $\boldsymbol{\lambda}^*(\cdot)$ *is positive over* $[0, T]$ *and* $\boldsymbol{\Lambda}(T) < \infty$ *almost surely, then for each* $k = 1, \ldots, m$ *the transformed points*

$$\{\Lambda_k(t_1^{(k)}), \ldots, \Lambda_k(t_{N_k(T)}^{(k)})\}$$

form a Poisson process with unit rate.

9.4 Exponentially Decaying Kernels

When performing a goodness of fit analysis using the random time change theorem, we evaluate the Hawkes process compensator at the time of each arrival. In general, this is an $\mathcal{O}(n(T)^2)$ operation, though when the excitation function decays exponentially this can become $\mathcal{O}(n(T))$.

Remember that $\Lambda(t) \coloneqq \int_0^t \lambda^*(s)\,\mathrm{d}s$ and that we can break up the time axis like

$$[0, t_{n(T)}] = [0, t_1] \cup (t_1, t_2] \cup \cdots \cup (t_{n(T)-1}, t_{n(T)}].$$

Using this, we can combine the fact that $\Lambda(t_{i+1}) - \Lambda(t_i) = \int_{t_i}^{t_{i+1}} \lambda^*(s)\,\mathrm{d}s$ with the Markovian form of the intensity (3.8) to get

$$\Lambda(t_{i+1}) - \Lambda(t_i) = \lambda(t_{i+1} - t_i) + (\lambda^*(t_i) + \alpha - \lambda)\frac{1}{\beta}(1 - \mathrm{e}^{-\beta(t_{i+1}-t_i)})\,. \tag{9.1}$$

Iterating i over all the arrivals gives us a $\mathcal{O}(n(T))$ method to perform the random time change.

Similarly, in the mutually exciting case, we can integrate (5.14) to find

$$\boldsymbol{\Lambda}(t_{i+1}) - \boldsymbol{\Lambda}(t_i) = \boldsymbol{\lambda}(t_{i+1} - t_i) + (\boldsymbol{\lambda}^*(t_i) + \boldsymbol{\alpha}_{d_i} - \boldsymbol{\lambda})\frac{1}{\boldsymbol{\beta}}(1 - \mathrm{e}^{-\boldsymbol{\beta}(t_{i+1}-t_i)})\,. \tag{9.2}$$

Part III
Case Studies

Chapter 10
Code Preliminaries

Before starting the data analysis, let us create some basic Python code to implement some of the fundamental Hawkes process functions discussed in earlier chapters. As variable names in Python can include Unicode characters (since version 3), we use variable names which are as close as possible to their equivalent mathematical notation. The full version of the code is available online as the Python package `hawkesbook`.

Before the following code, we assume that the common Python libraries have been imported, e.g.

```
import numpy as np
import numpy.random as rnd
from scipy.optimize import fsolve, minimize
```

10.1 Intensity Functions and Compensators

Translating mathematics to computer code is challenging as computers require very explicit instructions. Mathematical concepts often have many tedious details embedded in them, and we make notations which suppress these typically irrelevant details for the sake of clarity. The notation for the conditional intensity function, $\lambda^*(\cdot)$, is a prime example, as that little superscript asterisk quietly does a lot of heavy lifting!

Let us start with a general Hawkes process which is parameterised by the background rate λ and the excitation function $\mu(\cdot)$. For some calculations, we also need to evaluate $M(t) := \int_0^t \mu(s)\,\mathrm{d}s$ so we let $\boldsymbol{\theta} = (\lambda, \mu, M)$ for the following functions.

Remember, the full conditional intensity function is conditional on the history $\mathcal{H}(t)$ of all arrivals up to time t. For our purposes, the history $\mathcal{H}(t)$ can be

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

P. J. Laub et al., *The Elements of Hawkes Processes*,
https://doi.org/10.1007/978-3-030-84639-8_10

represented as the vector of all arrivals before time t. So our Python function for $\lambda^*(t)$ will look like `intensity(t, H_t, θ)`, taking the time t, the history $\mathcal{H}(t)$, and the vector $\boldsymbol{\theta}$ of parameters specifying our point process.

The general Hawkes intensity is:

```
def hawkes_intensity(t, H_t, θ): # see (3.2)
    λ, μ, _ = θ
    λˣ = λ
    for t_i in H_t:
        λˣ += μ(t - t_i)
    return λˣ
```

The line `λ, μ, _ = θ` is a Pythonism which just unpacks the $\boldsymbol{\theta}$ vector and creates variables for each of the three elements. Giving the third element the underscore variable name signifies that we do not care about this variable in this function (i.e. we do not need M to calculate the intensity).

For example, to evaluate the intensity of a Hawkes process with a power law decay $\mu(t) = k/(c+t)^p$, we could write:

```
# Assuming λ, k, c, p, t, H_t are defined earlier...
μ = lambda t: k / (c + t)**p
θ = (λ, μ, None)
λˣ = hawkes_intensity(t, H_t, θ)
```

Here, the line `μ = lambda t: k / (c + t)**p` just constructs a small function to calculate power law decay for the specific k, c, and p parameters.

Next, we can create a function to evaluate the compensator function $\Lambda(t) \coloneqq \int_0^t \lambda^*(s)\,\mathrm{d}s$. While M was not required to evaluate the intensity, it is required for the compensator while μ is not needed:

```
def hawkes_compensator(t, H_t, θ): # see (5.6)
    if t <= 0: return 0
    λ, _, M = θ

    Λ = λ * t
    for t_i in H_t:
        Λ += M(t - t_i)
    return Λ
```

Let us create the equivalent functions for the exponentially decaying excitation $\mu(t) = \alpha \mathrm{e}^{-\beta t}$. These processes are parameterised by the background intensity λ and the parameters α and β for the $\mu(\cdot)$, so $\boldsymbol{\theta} = (\lambda, \alpha, \beta)$. The resulting intensity function is:

```
def exp_hawkes_intensity(t, H_t, θ): # see (3.2)
    λ, α, β = θ
    λˣ = λ
    for t_i in H_t:
        λˣ += α * np.exp(-β * (t - t_i))
    return λˣ
```

The corresponding compensator is:

```
def exp_hawkes_compensator(t, H_t, θ): # see (5.6)
    if t <= 0: return 0
    λ, α, β = θ
    Λ = λ * t
    for t_i in H_t:
        Λ += (α/β) * (1 - np.exp(-β * (t - t_i)))
    return Λ
```

As noted in Sect. 9.4, the compensators for exponentially decaying Hawkes processes can be efficiently calculated by leveraging the Markov nature of the $(N(\cdot), \lambda^*(\cdot))$ process. To evaluate $\{\Lambda(T_i)\}_{i=1,\ldots,N(t)}\}$, e.g. during a goodness of fit analysis according to the random time change theorem, we can use:

```
def exp_hawkes_compensators(H_t, θ): # see (9.1)
    λ, α, β = θ

    Λ = 0
    λˣ_prev = λ
    t_prev = 0

    Λs = np.empty(len(H_t), dtype=np.float64)
    for i, t_i in enumerate(H_t):
        Λ += λ * (t_i - t_prev) + (
                 (λˣ_prev - λ)/β *
                 (1 - np.exp(-β * (t_i - t_prev))))
        Λs[i] = Λ

        λˣ_prev = λ + (λˣ_prev - λ) * (
                 np.exp(-β * (t_i - t_prev))) + α
        t_prev = t_i
    return Λs
```

The `enumerate` function is Python's way of looping through a vector and getting each item along with its corresponding index.

Finally, we can create versions of these methods for the power law decay kernel (3.4), where the $\boldsymbol{\theta}$ vector contains (λ, k, c, p):

```
def power_hawkes_intensity(t, ℋ_t, θ): # see (3.5)
    λ, k, c, p = θ
    λˣ = λ
    for t_i in ℋ_t:
        λˣ += k / (c + (t-t_i))**p
    return λˣ

def power_hawkes_compensator(t, ℋ_t, θ):
    λ, k, c, p = θ
    Λ = λ * t
    for t_i in ℋ_t:
        Λ += ((k * (c * (c + (t-t_i)))**-p *
               (-c**p * (c + (t-t_i)) + c * (c + (t-t_i))**p)) /
               (p - 1))
    return Λ

def power_hawkes_compensators(ℋ_t, θ):
    Λs = np.empty(len(ℋ_t), dtype=np.float64)
    for i in range(len(ℋ_t)):
        t_i = ℋ_t[i]
        ℋ_i = ℋ_t[:i]
        Λs[i] = power_hawkes_compensator(t_i, ℋ_i, θ)
    return Λs
```

10.2 Log-Likelihoods and MLE

Likelihoods relate a set of observations $\mathcal{H}(T)$ over a time period $t \in [0, T]$ to a particular parameter vector $\boldsymbol{\theta}$. Calculating the log-likelihood for a point process is relatively simple, if we already have some way to calculate its intensity function $\lambda^*(\cdot)$ and compensator $\Lambda(\cdot)$:

```
def log_likelihood(ℋ_T, T, θ, λˣ, Λ): # see (5.2)
    ℓ = 0.0
    for i, t_i in enumerate(ℋ_T):
        ℋ_i = ℋ_T[:i]
        λˣ_i = λˣ(t_i, ℋ_i, θ)
        ℓ += np.log(λˣ_i)
    ℓ -= Λ(T, ℋ_T, θ)
    return ℓ
```

This general log-likelihood function has the unimpressive $\mathcal{O}(n(T)^2)$ complexity. And without a functional form for $\mu(\cdot)$, we cannot use this log-likelihood to get an MLE. If we specialise to the case of power law decay Hawkes processes, then the log-likelihood becomes:

```
def power_log_likelihood(ℋ_T, T, θ):
    ℓ = 0.0
    for i, t_i in enumerate(ℋ_T):
        ℋ_i = ℋ_T[:i]
        λˣ_i = power_hawkes_intensity(t_i, ℋ_i, θ)
        ℓ += np.log(λˣ_i)
    ℓ -= power_hawkes_compensator(T, ℋ_T, θ)
    return ℓ
```

This still has $\mathcal{O}(n(T)^2)$ complexity (this is unavoidable for nearly all choices of $\mu(\cdot)$). For exponentially decaying Hawkes, Ozaki's efficient $\mathcal{O}(n(T))$ log-likelihood function is relatively simple to implement:

```
def exp_log_likelihood(ℋ_T, T, θ): # see (5.8)
    λ,α,β = θ
    t = ℋ_T
    N_T = len(ℋ_T)

    A = np.zeros(N_T)
    A[0] = 0
    for i in range(1, N_T):
        A[i] = np.exp(-β*(t[i] - t[i-1]))*(1 + A[i-1])

    ℓ = -λ*T
    for i, t_i in enumerate(ℋ_T):
        ℓ += np.log(λ + α*A[i]) - \
                (α/β)*(1 - np.exp(-β*(T-t_i)))
    return ℓ
```

Log-likelihoods are not inherently interesting, they are just a means to an end, and that end is to fit some data! With the log-likelihood for the exponential Hawkes process, we can calculate its MLE. Python's `SciPy` library has a `minimize` function which can numerically minimise functions given some constraints and an initial guess of the optimal solution. We will use this function to fit an exponential Hawkes process.

```
def exp_mle(ℋ_T, T, θ_start=(1, 2, 3)):
    eps = 1e-5
    θ_bounds = ((eps,None),(eps,None),(eps,None))
    loss = lambda θ: -exp_log_likelihood(t, T, θ)
    θ_mle = minimize(loss, θ_start, bounds = θ_bounds).x
    return θ_mle
```

The initial `θ_start` value here is chosen arbitrarily. We could give the optimiser a helping hand by using the result of the moment matching algorithm as the starting point for MLE. The `loss = lambda θ: -exp_log_likelihood(t, T, θ)` definition ensures that we maximise the likelihood by minimising the negative log-likelihood. The `bounds = θ_bounds` parameter tells the optimiser that λ, α, and β

are strictly positive (to be precise, it forces each $\theta_i \geq 10^{-5}$). Note, this does not restrict $\boldsymbol{\theta}$ to non-explosive Hawkes processes with $\alpha < \beta$.

The MLE method for the power law style Hawkes is quite similar to the exponential case:

```
def power_mle(t, T, θ_start=(1, 1, 2, 3)):
    eps = 1e-5
    θ_bounds = ((eps,None),(eps,None),(eps,None),
        (1+eps, 100))
    loss = lambda θ: -power_log_likelihood(t, T, θ)
    θ_mle = minimize(loss, θ_start, bounds = θ_bounds).x
    return θ_mle
```

Here the only major difference is the $\boldsymbol{\theta} = (\lambda, k, c, p)$ have different bounds. We need to enforce that $p > 1$, and we noticed the stability of the fits improved slightly if we also forced p to not become too large either.

10.3 Simulation

The most general simulation method for point processes is the inverse compensator method outlined in Sect. 4.1. This method simulates a unit rate Poisson process $\{t_1^*, t_2^*, \dots\}$, then creates arrival times $\{t_1, t_2, \dots\}$ by solving the inverse compensator equations (4.1)

$$t_1^* = \Lambda(t_1), \quad t_{i+1}^* - t_i^* = \Lambda(t_{i+1}) - \Lambda(t_i).$$

Generating unit rate Poisson processes is quite simple, as $t_{i+1}^* - t_i^* \sim \mathsf{Exp}(1)$, thus $t_{i+1}^* - t_i^* \overset{\mathcal{D}}{=} -\ln(U)$ where $U \sim \mathsf{Unif}(0, 1)$. The harder task (at least for Python!) is to solve these series of equations.

Python, and most other languages, phrases these kind of problems as root-finding problems. So, instead of asking Python to find t_i which solves this set of equalities, we pass it the expressions

$$\Lambda(t_1) - t_1^*, \quad \text{and} \quad \Lambda(t_{i+1}) - \Lambda(t_i) - (t_{i+1}^* - t_i^*)$$

and ask it to find the t_i which set these expressions to zero.

```
def simulate_inverse_compensator(θ, Λ, N): # see Section 4.1
    ℋ = np.empty(N, dtype=np.float64)

    tˣ_1 = -np.log(rnd.rand())
    exp_1 = lambda t_1: Λ(t_1, ℋ[:0], θ)-tˣ_1
```

```
    t_1_guess = 1.0
    t_1 = fsolve(exp_1, t_1_guess)[0]

    ℋ[0] = t_1
    t_prev = t_1
    for i in range(1, N):
        Δtˣ_i = -np.log(rnd.rand())

        Λ_i = Λ(t_prev, ℋ, θ)
        exp_i = lambda t_next: Λ(t_next, ℋ[:i], θ)-Λ_i-Δtˣ_i

        t_next_guess = t_prev + 1.0
        t_next = fsolve(exp_i, t_next_guess)[0]

        ℋ[i] = t_next
        t_prev = t_next
    return ℋ
```

This method can simulate any type of point process, though its generality comes at the cost of slow and unreliable performance. The numerical `fsolve` method appears to find suboptimal solutions frequently, so this specific inverse compensator method is not recommended for general use.

To get faster simulation algorithms, we must consider Hawkes processes with a specific $\mu(\cdot)$, and the most efficient simulation algorithms are for the exponential Hawkes (due to its joint Markovian nature). Perhaps the most elegant algorithm for exponential Hawkes is the exact method which uses the composition method:

```
def exp_simulate_by_composition(θ, N): # see (4.2)
    λ, α, β = θ
    λˣ_k = λ
    t_k = 0

    ℋ = np.empty(N, dtype=np.float64)
    for k in range(N):
        U_1, U_2 = rnd.rand(2)
        T_1 = t_k - np.log(U_1) / λ
        T_2 = t_k - np.log(1 + β / (λˣ_k+α-λ)*np.log(U_2))/β
        t_prev = t_k
        t_k = min(T_1, T_2)
        ℋ[k] = t_k
        λˣ_k = λ + (λˣ_k + α - λ) * (
                  np.exp(-β*(t_k - t_prev)))
    return ℋ
```

This method simulates a fixed number of arrivals, though if the for loop were replaced by `while t_k < T` it will keep simulating until it passes time T.

While the composition method is specific to the exponentially decaying Hawkes, the thinning method works for a large class of point processes. The thinning simulation method for the exponentially decaying Hawkes is given next:

```
def exp_simulate_by_thinning(θ, T): # see (4.3)
    λ, α, β = θ  # Line 1

    λˣ = λ
    times = []
    t = 0
    while True:
        M = λˣ
        Δt = rnd.exponential() / M
        t += Δt
        if t > T:
            break
        λˣ = λ + (λˣ - λ) * np.exp(-β * Δt) # Line 2
        u = M * rnd.rand()
        if u > λˣ:
            continue # This potential arrival is 'thinned' out
        times.append(t)
        λˣ += α # Line 3
    return np.array(times)
```

To modify this method to simulate power law Hawkes, change Line 1 to `λ, k, c, p = θ`, Line 3 to `λˣ += k / (c ** p)`, and Line 2 to:

```
        λˣ = power_hawkes_intensity(t, np.array(times), θ)
```

10.4 Fitting

Fitting a Hawkes process with a large number of events can be slow using likelihood-based methods. A GMM fit is faster and can be used as a starting value for the other fitting methods. For any moment-based fitting procedure, we need to calculate the empirical moments of our data, and the theoretical moments given a $\boldsymbol{\theta}$ parameter. The empirical moments of the binned observations can be found using:

```
def empirical_moments(t, T, τ, lag): # see Chapter 7
    bins = np.arange(0, T, τ)
    N = len(bins) - 1
    count = np.zeros(N)

    for i in range(N):
        count[i] = np.sum((bins[i] <= t) & (t < bins[i+1]))

    empMean = np.mean(count)
    empVar = np.std(count)**2
    empAutoCov = np.mean((count[:-lag] - empMean) *
                 (count[lag:] - empMean))
```

```
    return np.array([empMean, empVar, empAutoCov]).reshape(3,1)
```

For the simple exponential Hawkes process, the theoretical moments for binned observations are:

```
def exp_moments(θ, τ, lag): # see (7.11), (7.12), and (7.13).
    λ,α,β = θ
    κ = β - α;
    δ = lag*τ

  mean = (λ*β/κ)*τ
  var = (λ*β/κ)*(τ*(β/κ) + (1-β/κ)*((1-np.exp(-κ*τ))/κ))
  autoCov = (λ*β*α*(2*β-α)*(np.exp(-κ*τ)-1)**2/(2*κ**4)) \
          *np.exp(-κ *δ)
  return np.array([mean, var, autoCov]).reshape(3,1)
```

From there, the GMM loss function is:

```
def exp_gmm_loss(θ, τ, lag, empMoments, W): # see (7.3)
    moments = exp_moments(θ, τ, lag)
    g = empMoments - moments
    return (g.T).dot(W).dot(g).item()
```

Putting it all together, the GMM fitting procedure can be written as:

```
def exp_gmm(t, T, τ=5, lag=5, iters=2, θ_start=(1, 2, 3)):
    empMoments = empirical_moments(t, T, τ, lag)

    W = np.eye(3)
    bounds = ((0, None), (0, None), (0, None))
    θ = minimize(exp_gmm_loss, x0=θ_start,
            args=(τ, lag, empMoments, W),
            bounds=bounds).x

    for i in range(iters):
        moments = exp_moments(θ, τ, lag)

        g = empMoments - moments
        S = g.dot(g.T)
        W = np.linalg.inv(S)
        W /= max(W) # Avoid overflow of the loss function

        θ = minimize(exp_gmm_loss, x0=θ,
                args=(τ, lag, empMoments, W),
                bounds=bounds).x
    return θ
```

The GMM method is a very fast fitting procedure but it is fast because it is very crude. By contrast, the EM method can be much more accurate. This method iterates between an 'E' step and an 'M' step. In the first step, we calculate some useful expectations which are sometimes called *responsibilities*:

```
def em_responsibilities(t, θ):
    λ,α,β = θ

    N = len(t)
    resp = np.zeros((N,N))
    resp[:,0] = λ
    resp[0,0] = 1
    for i in range(1, N):
        resp[i, 1:i+1] = α*np.exp(-β*(t[i] - t[:i]))
        resp[i, :i+1] = resp[i,:i+1]/np.sum(resp[i, :i+1])
    return resp
```

The entire EM algorithm can then be written as:

```
def exp_em(t, T, iters=100, θ_start=(1, 2, 3)):
    N = len(t)
    θ = θ_start.copy()
    llIterations = np.zeros(iters)
    for i in range(iters):
        # E step: calculate responsibilities
        resp = em_responsibilities(t, θ)

        # M steps: update |λ|, then |α|, then |β|
        λ,α,β = θ
        λ = np.sum(resp[:,0])/T
        numer = np.sum(resp[:,1:])
        denom = np.sum(1 - np.exp(-β*(T - t)))
        α = β*numer/denom

        # M step: Update |β|
        numer = np.sum(1 - np.exp(-β*(T - t)))/β  \
              - np.sum((T - t)*np.exp(-β*(T - t)))
        denom = np.sum([np.sum((t[i] - t[:i])*resp[i,1:i+1]) \
                        for i in range(1,N)])
        β = α*numer/denom
        θ = (λ, α, β)
        llIterations[i] = exp_log_likelihood(t, T, θ)
    return θ, llIterations
```

10.5 Mutually Exciting Hawkes Processes

The prospect of implementing the mutually exciting generalisation of the Hawkes process is not as daunting as it may first appear. In the general case, we parameterise the process by $\boldsymbol{\theta} = (\boldsymbol{\lambda}, \boldsymbol{\mu})$. The $\boldsymbol{\lambda} \in \mathbb{R}^m$ are the vector of background rates for each process, and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_m)$ is a vector of m functions like $\mu_k : \mathbb{R} \to \mathbb{R}^m_+$, which describe the excitatory effect of an arrival on each of the m conditional intensities. Given a $\boldsymbol{\theta}$ and a history $\mathcal{H}(t)$, a vector $\boldsymbol{\lambda}^*(\cdot)$ containing the intensity of each process can be calculated:

```
def mutual_hawkes_intensity(t, ℋ_t, θ): # see (3.14)
    λ, μ = θ
    λˣ = λ
    for (t_i, d_i) in ℋ_t:
        λˣ += μ[d_i](t - t_i)
    return λˣ
```

In the simpler case where the excitation functions decay exponentially we have $\boldsymbol{\theta} = (\boldsymbol{\lambda}, A, \boldsymbol{\beta})$. Remember, after an arrival to process k, the joint intensity $\boldsymbol{\lambda}^*$ jumps up by the amount given in $\boldsymbol{\alpha}_k$, the kth row of the matrix A.

```
def mutual_exp_hawkes_intensity(t, times, ids, θ):
    λ, α, β = θ
    λˣ = λ.copy()
    for (t_i, d_i) in zip(times, ids):
        λˣ += α[d_i] * np.exp(-β*(t - t_i))
    return λˣ
```

Along the same lines, the compensators $\boldsymbol{\Lambda}(\cdot)$ are calculated by:

```
def mutual_exp_hawkes_compensator(t, times, ids, θ):
    λ, α, β = θ
    Λ = λ * t
    for (t_i, d_i) in zip(times, ids):
        Λ += (α[d_i]/β) * (1 - exp(-β*(t - t_i)))
    return Λ
```

The efficient calculation of $\{\boldsymbol{\Lambda}(t_i)\}_{i=1,\ldots,N(t)}$ for the mutually exciting Hawkes is given below.

```
def mutual_exp_hawkes_compensators(times, ids, θ): # see (9.2)
    λ, α, β = θ
    m = len(λ)

    Λ = np.zeros(m)
    λˣ_prev = λ
```

```
    t_prev = 0

    Λs = np.zeros((len(times), m), dtype=np.float64)

    for i in range(len(times)):
        t_i = times[i]
        d_i = ids[i]

        Λ += λ * (t_i - t_prev) +
                 (λˣ_prev - λ)/β *
                 (1 - np.exp(-β*(t_i - t_prev)))
        Λs[i,:] = Λ

        λˣ_prev = λ + (λˣ_prev - λ) *
            np.exp(-β*(t_i - t_prev)) + α[d_i,:]
        t_prev = t_i

    return Λs
```

The random time-changed points for the kth component,

$$\left\{\Lambda_k(t_1^{(k)}), \ldots, \Lambda_k(t_{N_k(T)}^{(k)})\right\}$$

can be extracted by:

```
# Assuming times, ids, θ, and k are defined earlier...
Λs = mutual_exp_hawkes_compensators(times, ids, θ)
timeShifted_k = Λs[ids == k]
```

Simulating the mutually exciting Hawkes using thinning is a natural extension to the one-dimensional case:

```
def mutual_exp_simulate_by_thinning(θ, T):
    λ, α, β = θ
    m = len(λ)

    λˣ = λ
    times = []

    t = 0

    while True:
        M = np.sum(λˣ)
        Δt = rnd.exponential() / M
        t += Δt
        if t > T:
            break

        λˣ = λ + (λˣ - λ)*np.exp(-β * Δt)
```

```
        u = M * rnd.rand()
        if u > np.sum(λˣ):
            continue # No arrivals (they are 'thinned' out)

        cumulativeλˣ = 0

        for i in range(m):
            cumulativeλˣ += λˣ[i]
            if u < cumulativeλˣ:
                times.append((t, i))
                λˣ += α[i]
                break

    return times
```

The log-likelihood for the general mutually exciting Hawkes case is analogous to the one-dimensional case above:

```
def mutual_log_likelihood(ℋ_T, T, θ, λˣ, Λ): # see (5.10)
    m = len(θ)
    ℓ = 0
    for (t_i, d_i) in ℋ_T:
        # Get the history of arrivals before time t_i
        ℋ_i = [(t_s, d_s) for (t_s, d_s) in ℋ_T if t_s < t_i]
        λˣ_i = λˣ(t_i, ℋ_i, θ)
        ℓ += log(λˣ_i[d_i])

    ℓ -= np.sum(Λ(T, ℋ_T, θ))
    return ℓ
```

The special case for exponentially decaying Hawkes, utilising the (5.12) relation, is:

```
def mutual_exp_log_likelihood(times, ids, T, θ):
    λ, α, β = θ

    if np.min(λ) <= 0 or np.min(α) < 0 or np.min(β) <= 0:
        return -np.inf

    ℓ = 0
    λˣ = θ[0]

    t_prev = 0
    for t_i, d_i in zip(times, ids):
        λˣ = λ + (λˣ - λ)*np.exp(-β * (t_i - t_prev))
        ℓ += log(λˣ[d_i])

        λˣ += α[d_i,:]
        t_prev = t_i
```

```
    Λs_T = mutual_exp_hawkes_compensator(T, times, ids, θ)
    ℓ -= np.sum(Λs_T)

    return ℓ
```

The only final hurdle before we can run MLE is the fact that `Scipy`'s optimiser only accepts a 1-dimensional vector argument, whereas $\boldsymbol{\theta} = (\boldsymbol{\lambda}, A, \boldsymbol{\beta})$ is a combination of a vector, a matrix, and a vector. The following two functions just help to flatten the $\boldsymbol{\theta}$ into one long vector and the reverse operation:

```
def flatten_theta(θ):
    return np.hstack([θ[0], np.hstack(θ[1]), θ[2]])

def unflatten_theta(θ_flat, m):
    λ = θ_flat[:m]
    α = θ_flat[m:(m + m**2)].reshape((m,m))
    β = θ_flat[(m + m**2):]

    return (λ, α, β)
```

Combining all these pieces, the mutually exciting Hawkes with exponential decay can be fit using MLE with:

```
def mutual_exp_mle(t, ids, T, θ_start):
    m = len(θ_start[0])
    θ_start_flat = flatten_theta(θ_start)

    def loss(θ_flat):
        θ = unflatten_theta(θ_flat, m)
        return -mutual_exp_log_likelihood(t, ids, T, θ)

    res = minimize(loss, θ_start_flat,
            method = 'Nelder-Mead')

    θ_mle = unflatten_theta(res.x, m)
    logLike = -res.fun

    return θ_mle, logLike
```

Chapter 11
Seismology

In this section, we consider the most traditional application of Hawkes processes, which is modelling the arrival of earthquakes. In honour of the many early Hawkes researchers from Japan, we will take a look at records of earthquakes near the Japanese islands (Fig. 11.1). There is a natural motivation for using Hawkes processes to model earthquake arrivals. This is the idea that a large ('parent') earthquake is often followed by a few smaller earthquakes or aftershocks ('children').

11.1 Data Preparation and Exploration

We start by importing some packages, including the code from the previous chapter which is packaged together with the name `hawkesbook`.

```
import hawkesbook as hawkes
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy import stats
from statsmodels.graphics.gofplots import qqplot
```

Then we load and preprocess the data. Each earthquake arrival is given as a datetime, so we convert this into the (fractional) number of days after the start of our observation period (1/1/1973).

```
quakes = pd.read_csv("japanese-earthquakes.csv")
timeToQuake = quakes.index pd.Timestamp("1/1/1973")
ts = np.array(timeToQuake.total_seconds()/60/60/24)
```

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

P. J. Laub et al., *The Elements of Hawkes Processes*,
https://doi.org/10.1007/978-3-030-84639-8_11

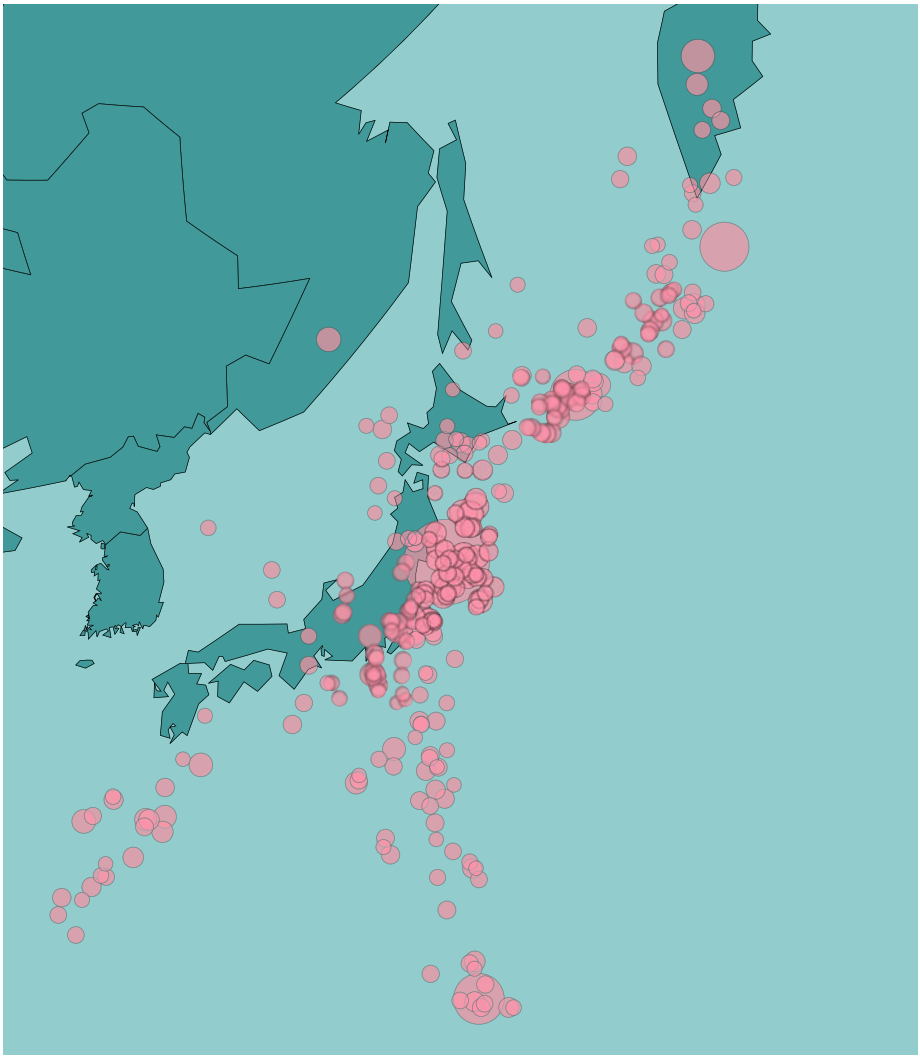

Fig. 11.1 The locations of 300 earthquakes from the dataset of 48,888 earthquakes in the Japan region between 1st January 1973 and 31 December 2020. The data was downloaded from the IRIS Earthquake Browser, which collates various sources (e.g. including the USGS)

To start, we can generate some plots to familiarise ourselves with the data. We can look at how many earthquakes occurred in each year of the dataset:

```
quakes.Year.hist()
```

The number of earthquakes each year shows us that the process (or our measurement of it) is not stationary (Fig. 11.2). In fact, the huge spike in the number

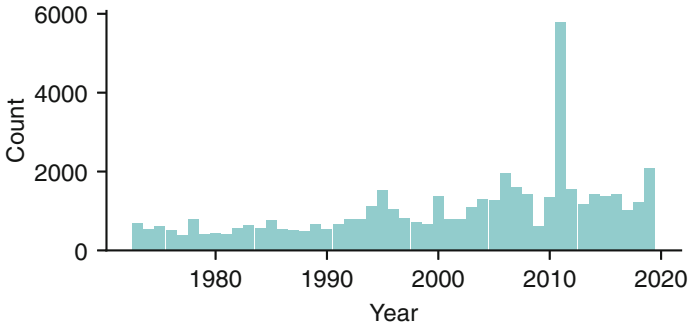

Fig. 11.2 Number of earthquakes recorded each year

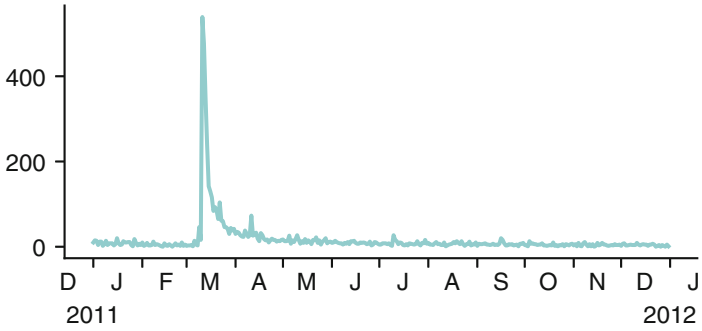

Fig. 11.3 Number of earthquakes recorded each day in 2011

of earthquakes in 2011 may be indicative of a self-exciting process (assuming it is not merely an error in the data collection). Plotting the number of earthquakes each day of 2011 shows a massive spike in activity in March which takes about a month to die off (Fig. 11.3).

```
allDates = pd.date_range("2011/1/1", "2011/12/31")

numByDate = pd.Series(index=allDates)
for i, date in enumerate(allDates):
    numByDate[i] = np.sum(quakes.index.date == date)

plt.plot(numByDate)
```

The decay in the number of earthquakes after this major shock in March looks quite similar to the exponential decay of a Hawkes intensity process. Then again, the same plot for 1990 shows a year where earthquakes arrived at quite regular intervals (Fig. 11.4).

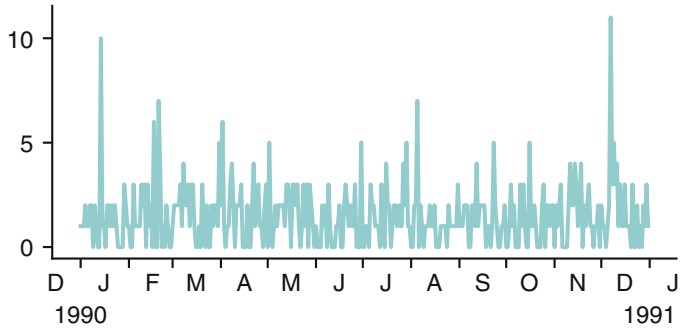

Fig. 11.4 Number of earthquakes recorded each day in 1990

11.2 Poisson Process

It is worthwhile to first check that the simplest model of arrivals, the Poisson process, does not work. Fitting a Poisson process is as simple as calculating the interarrival times and taking their average.

```
iat = np.diff(np.insert(ts, 0, 0))
poissonRate = 1/np.mean(iat)
```

For this data, the Poisson model says we expect earthquakes to arrive independently at a rate of ≈ 2.79 per day. We can compare the empirical CDF of the interarrival times against the $\mathsf{Exp}(2.79)$ CDF:

```
empCDF = np.arange(len(iat))/float(len(iat))
plt.plot(np.sort(iat), empCDF)

x = np.linspace(0, np.max(iat), 1_000)
trueCDF = stats.expon(scale=1/poissonRate).cdf(x)
plt.plot(x, trueCDF)
```

The CDFs show some differences between the observed and expected values (Fig. 11.5). A Q–Q plot highlights these differences, and can easily be produced with the 'qqplot' function from `statsmodels` (or the 'probplot' function in `scipy`):

```
qqplot(iat, stats.expon(scale=1/poissonRate))
```

The Q–Q plot shows that the Poisson assumption is not a great fit for this dataset (Fig. 11.6). This outcome is unsurprising. We would not expect a complex geological process to be explained by a single parameter model over a 50 year time horizon!

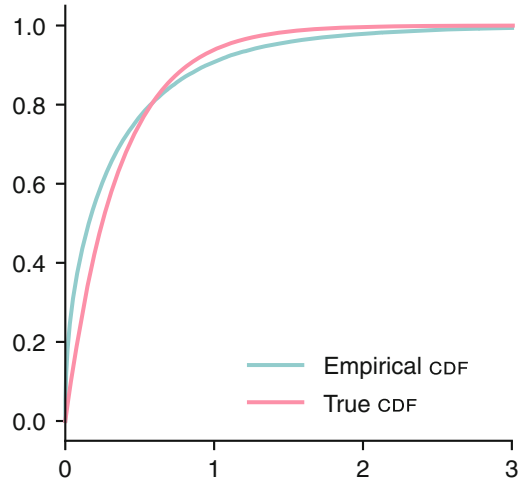

Fig. 11.5 Comparing the empirical CDF of the interarrival times to the Exp(2.79) CDF implied by a Poisson process model for earthquake arrivals

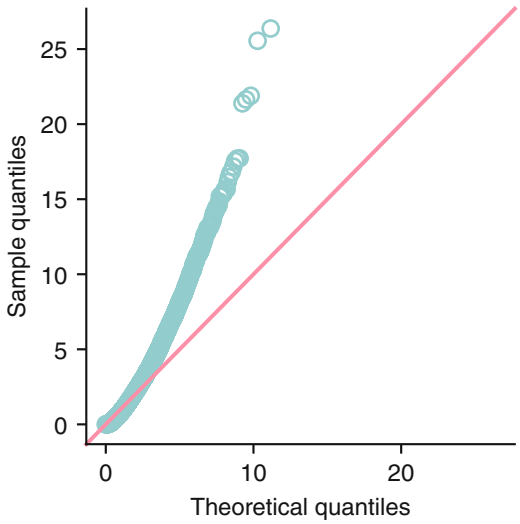

Fig. 11.6 Q–Q plot for a Poisson model for earthquake arrivals

11.3 Hawkes Process with Exponential Decay

We can fit the exponential Hawkes model to this data using, for example, the MLE method:

```
obsPeriod = pd.Timestamp("31/12/2020")-pd.Timestamp("1/1/1973")
T = obsPeriod.days
θ_exp_mle, llIterations = hawkes.exp_mle(ts, T)
```

The resulting fit is $\widehat{\boldsymbol{\theta}}_{\text{Exp}}^{\text{MLE}} = (\widehat{\lambda}, \widehat{\alpha}, \widehat{\beta}) \approx (0.850, 1.01, 1.45)$. As the log-likelihood for the exponential Hawkes model, implemented in

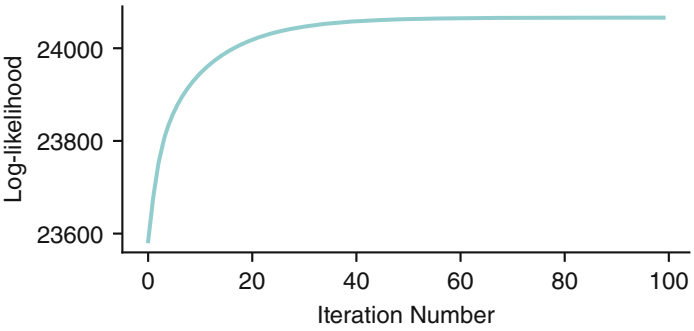

Fig. 11.7 The evolution of the log-likelihood for the Hawkes process fits produced in each iteration of the EM algorithm

`hawkes.exp_log_likelihood(ℋ_T, T, θ)`, uses Ozaki's efficient $\mathcal{O}(n(T))$ algorithm, this MLE fit is quite fast. On a late 2020 Mac Mini it finishes in about 50 ms.

The MLE can sometimes be a bit unreliable as the numerical optimiser may become stuck in a local minima. We can use the EM algorithm as a backup:

```
θ_em, llIterations = hawkes.exp_em(ts, T, iters=100)
```

The EM fit takes much longer than the MLE, finishing after about 10 minutes. As the EM algorithm proceeds, it updates the fitted $\boldsymbol{\theta}$ to increase the likelihood at each iteration. We can plot the log-likelihood at each of these iterations to see if the algorithm has basically converged (Fig. 11.7).

It seems 100 iterations are enough to let the EM process converge. The resulting fit is $\widehat{\boldsymbol{\theta}}^{\text{EM}}_{\text{Exp}} \approx (0.853, 1.02, 1.46)$ which is basically the same as the MLE. If the EM algorithm it left to run even longer, we find that after about 250 iterations the first 5 significant figures of the $\widehat{\boldsymbol{\theta}}^{\text{EM}}_{\text{Exp}}$ do not appear to change.

```
ll_mle = hawkes.exp_log_likelihood(ts, T, θ_exp_mle)
ll_em = hawkes.exp_log_likelihood(ts, T, θ_exp_em)
θ_exp = θ_exp_mle if ll_mle > ll_em else θ_exp_em
ll_exp = max(ll_mle, ll_em)
```

As the MLE finds a slightly higher maximum value ($\ell = 24066.14$ compared to $\ell = 24066.10$), we keep the MLE fit $\widehat{\boldsymbol{\theta}}_{\text{Exp}} = \widehat{\boldsymbol{\theta}}^{\text{MLE}}_{\text{Exp}}$.

To create a Q–Q plot for the Hawkes process fit, we use our knowledge about compensators from (4.1). That is, if we calculate the compensator at each arrival time, then this forms a unit-rate Poisson process, and then a Q–Q plot is simple to make.

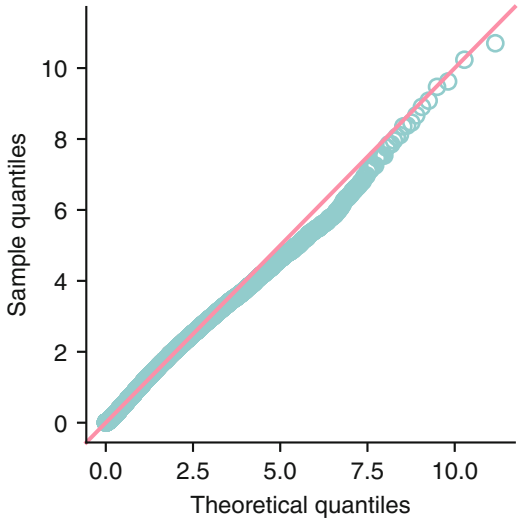

Fig. 11.8 Q–Q plot for an exponentially-decaying Hawkes process model for earthquake arrivals

```
tsShifted = hawkes.exp_hawkes_compensators(ts, θ_exp)
iat = np.diff(np.insert(tsShifted, 0, 0))
qqplot(iat, dist=stats.expon)
```

This Q–Q plot shows that the Hawkes model is quite a promising model for this data (Fig. 11.8).

We use the *parametric bootstrap method* [26, Section 10.4] to find confidence intervals for the fit. That is, we take the fitted value $\widehat{\boldsymbol{\theta}}_{\text{Exp}}$, simulate a large number of Hawkes processes with these values, and then fit each of the simulated datasets using MLE. The sample quantiles of these fitted values will function as confidence intervals for our original fit.

```
θ_bootstrap = []

for seed in range(10_000):
    rnd.seed(seed)
    tsSim = hawkes.exp_simulate_by_thinning(θ_exp, T)
    θ_mle = hawkes.exp_mle(tsSim, T)
    θ_bootstrap.append(θ_mle)

θ_bootstrap = np.vstack(θ_bootstrap)
print(np.quantile(θ_bootstrap, [0.025, 0.975], axis=0))
```

The resulting 95% confidence intervals are $\widehat{\lambda} \in (0.820, 0.882)$, $\widehat{\alpha} \in (0.977, 1.04)$, and $\beta \in (1.40, 1.51)$. The histograms of the bootstrapped values are plotted in Fig. 11.9.

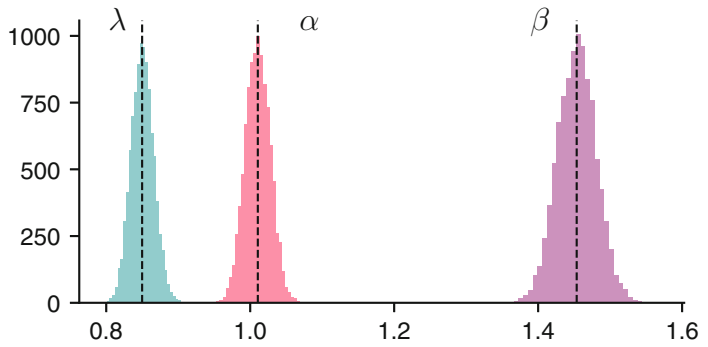

Fig. 11.9 Histograms of the 10,000 bootstrapped values for (from left to right) each parameter λ, α, and β using the parametric bootstrap method

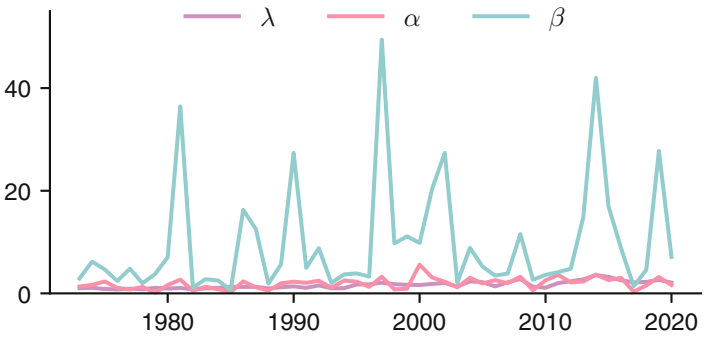

Fig. 11.10 The fitted $\boldsymbol{\theta} = (\lambda, \alpha, \beta)$ after running the EM algorithm on each year of data separately

To assess the long term stability of the exponentially-decaying Hawkes process, we fit a separate Hawkes model for each year of earthquakes in the observation period (Fig. 11.10).

```
years = sorted(set(quakes.Year))
numYears = len(years)

θ_yearly = []

for i, year in enumerate(years):
    quakesYear = quakes[quakes.Year == year]

    nyd = pd.Timestamp(f"01/01/{year}")
    timeToQuake = quakesYear.index - nyd
    ts_i = timeToQuake.total_seconds()/60/60/24

    obsPeriod = pd.Timestamp(f"1/1/{year+1}") - nyd
    T_i = obsPeriod.days
```

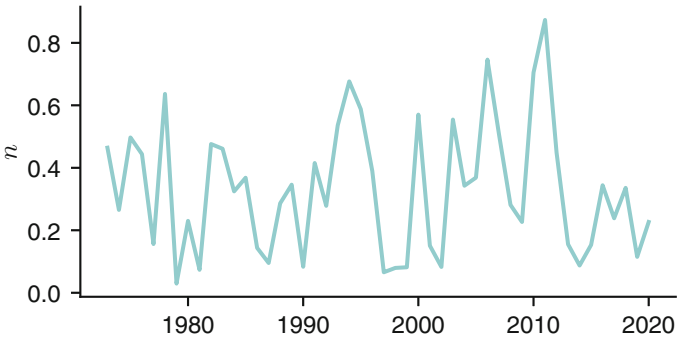

Fig. 11.11 The fitted branching ratio $n = \alpha/\beta$ found by running the EM algorithm on each year separately

```
    θ_yearly.append(hawkes.exp_mle(ts_i, T_i))

θ_yearly = np.vstack(θ_yearly)
plt.plot(θ_yearly)
```

The background rate λ and the intensity jump size α are quite stable over time. The β parameter is not stable. This is quite strange, as when β is large we have a process which becomes somewhat like a Poisson process, yet when it is low it is has a longer memory and is more self-exciting. Figure 11.11 shows the fitted branching ratio $n = \alpha/\beta$ for each year; the erratic nature of the yearly fits is also visible there.

This may simply be the noise which is caused by using smaller sample sizes in our fits. Or this extra variability may represent a more complicated process which is not readily fit by the simple Hawkes for the varying time scales.

11.4 Hawkes Process with Power Law Decay

The other most popular form of the Hawkes process is the one with a power law decay kernel. We fit the entire dataset (not just a year's worth) using this power law Hawkes.

```
θ_pl = hawkes.power_mle(ts, T)
ll_pl = hawkes.power_log_likelihood(ts, T, θ_pl)
```

As the power law log-likelihood is an $\mathcal{O}(n(T)^2)$ operation, this MLE fit is far slower than the exponential decay. On the same computer, it took just over one hour to fit this model. Extending the log-likelihood computation to run as much in parallel as possible decreased this time to about 20 min.

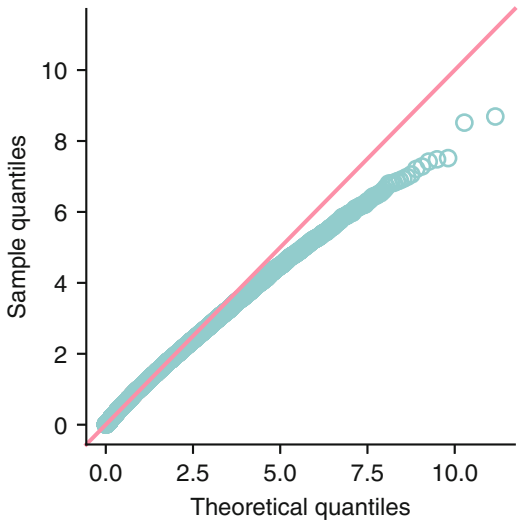

Fig. 11.12 Q–Q plot for a power law decaying Hawkes process model for earthquake arrivals

The result is $\boldsymbol{\theta}_{\text{PL}} = (\widehat{\lambda}, \widehat{k}, \widehat{c}, \widehat{p}) \approx (0.298, 0.101, 0.0436, 1.148)$. The log-likelihood is $\ell = 25730.1$, which indicates that the power law fit is better than the exponentially-decaying Hawkes which had $\ell = 24066.1$. It is slightly inappropriate to compare the log-likelihoods directly as the power law Hawkes has one more parameter than the exponentially decaying form. Calculating the Bayesian information criterion (BIC) can be done by:

```
BIC_exp = 3 * np.log(len(ts)) - 2 * ll_exp
BIC_pl = 4 * np.log(len(ts)) - 2 * ll_pl
```

The BIC for the exponential model $-48{,}099$ is larger than the power law BIC of $-51{,}417$. Thus the power law version is the better model even after penalising the larger number of parameters. Paradoxically, the Q–Q plot for the power law fit in Fig. 11.12 appears to be worse than the exponentially decaying version in Fig. 11.8.

11.5 Discussion

Earthquakes are more than just an instant in time; they are events with a location, and a magnitude. If we wished to consider a more accurate model of earthquakes arrivals, we would incorporate this extra information into the Hawkes model. The *ETAS* model does exactly that. The conditional intensity function of the epidemic-type aftershock sequence (ETAS) model is like $\lambda^*(t, x, y, m)$ as it depends on time t, the location (latitude and longitude) x and y, and the magnitude m. More specifically, it decomposes to

$$\lambda^*(t, x, y, m) = \nu(m)\lambda(t, x, y \mid \mathcal{H}_t),$$

where $\nu(m)$ controls the magnitude of new arrivals, and $\lambda(t, x, y \mid \mathcal{H}_t)$ is the conditional intensity function of a spatial Hawkes process (i.e. a marked Hawkes process, where the marks are the spatial coordinates of each arrival). The `ETAS` R package would be a place to start for performing inference with this model [38].

Interestingly, another model characteristic to consider is the opposite of self-exciting, called *self-inhibiting processes* or *stress-release processes*. In these processes, each arrival will decrease the intensity for future arrivals. Somewhat paradoxically, these have also been used to model earthquake arrivals. The logic is that tectonic plates move against each other and build up a level of stress/tension over time, which is released by an earthquake. This would result in earthquake arrivals which are more regular than the Poisson process. Perhaps, some combination of these two ideas is in action; e.g. small-magnitude earthquakes behave like this stress-release model, whereas large-magnitude earthquakes cause aftershocks and thus are more Hawkes-like in nature. As we are not particularly knowledgeable about earthquakes, we dare not speculate too much on the matter.

Chapter 12
Finance

Finance is one domain where mutually exciting Hawkes processes are particularly interesting. The global financial crisis was a painful lesson in the unobserved interdependencies of our financial systems. In the literature, there is the term 'financial contagion' to describe the effect of a financial event in one firm or asset impacting other firms or assets like a virus. The mutually exciting Hawkes model is obviously a useful model for any financial risk manager.

12.1 Data Preparation and Exploration

To demonstrate the methodology, we will look at data on the trades in some popular technology stocks. The particular stocks and times we considered do not fit perfectly into the mutually exciting Hawkes model; real data is not always particularly obliging to our modelling desires. However, the steps in this analysis are informative and can be used as a guide for other financial (or non-financial) datasets.

The stocks we considered are Amazon (AMZN.OQ), Apple (AAPL.OQ), Facebook (FB.OQ), Microsoft (MSFT.OQ), and Twitter (TWTR.K). We have all trades of these stocks for the 20th April 2021 between 1–2 pm New York time (ET), which corresponds to the time of Apple's recent 'Spring Loaded' event.

```
# Assuming we have already imported the same packages
# as in the previous chapter...
names = ["amazon", "apple", "facebook", "microsoft", "twitter"]
stocks = [pd.read_csv(name + ".csv") for name in names]
m = len(stocks)
```

There are 44,070 trades in this dataset, 2275 for Amazon, 16,417 for Apple, 5254 for Facebook, 8739 for Microsoft, and 11,385 for Twitter. Figure 12.1 shows the price for Apple's stock during the period. Though this particular hour was chosen

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

P. J. Laub et al., *The Elements of Hawkes Processes*,
https://doi.org/10.1007/978-3-030-84639-8_12

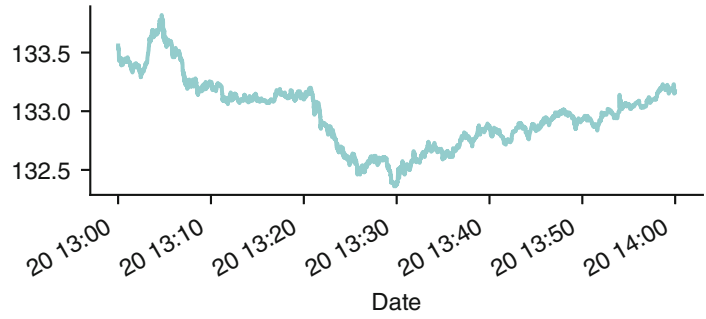

Fig. 12.1 Stock price for Apple (AAPL.OQ) between 1–2 pm on the 20th April 2021

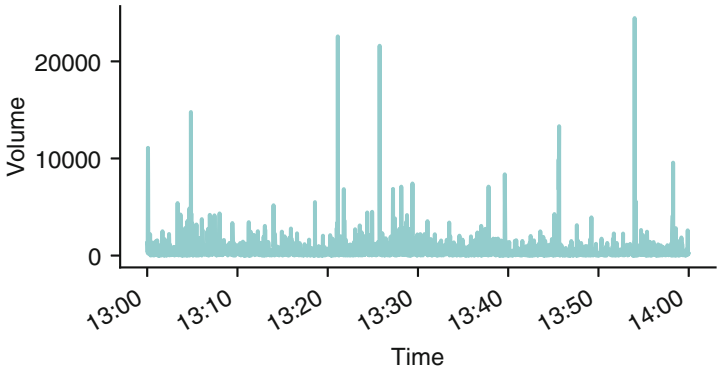

Fig. 12.2 Volume of trades of Apple (AAPL.OQ) each second between 1–2 pm on the 20th April 2021

because of Apple's new devices being announced, their stock price did not have any notable swings (nor did their competitors).

The volume of shares traded each second, as in Fig. 12.2, appears quite bursty.

Similarly, the number of trades which occurred each second (Fig. 12.3) shows a steady background-rate of trades with a few periods of high activity superimposed. This is the data which we will fit.

To start the fitting, we convert the date-times of each trade (accurate to the millisecond) into simple floating-point numbers.

```
firstTime = np.min([stock.Date.iloc[0] for stock in stocks])
for k in range(m):
    timeDiff = stocks[k].index - firstTime
    stocks[k] = timeDiff.total_seconds().to_numpy()
```

Stock exchanges are just computer systems, and no computer operates in continuous time. Exchanges look at discrete intervals, say every millisecond, and if any valid trades are pending it will process them all together. As such, many of

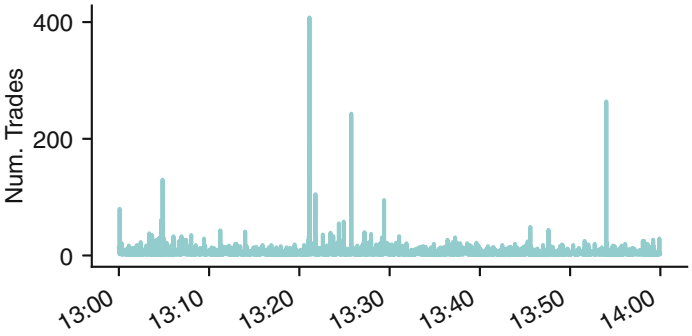

Fig. 12.3 Number of trades of Apple (AAPL.OQ) each second between 1–2 pm on the 20th April 2021

the trades in our data occurred at the same time, and this is not acceptable for the Hawkes model. To address this, we modify the times by adding some millisecond-level noise to separate these events. The next code adds some Unif(0, 1) ms noise to the timestamps, and reorder the trades accordingly:

```
rg = rnd.default_rng(1)

for k in range(m):
    dt = np.diff(stocks[k])
    tickSize = np.min(dt[dt > 0])
    stocks[i] = sort(stocks[k] +
            tickSize * rg.uniform(size=len(stocks[k])))

T = np.max([stocks[k][-1] for k in range(m)]) + 1
```

We will take these modified times of trades as raw data—the points of a hypothetical point process.

12.2 Independent Poisson Processes

Before proceeding, it is worthwhile to see how the Poisson model would fit this data.

```
for stock in stocks:
    iat = np.diff(np.insert(stock, 0, 0))
    qqplot(iat, dist=stats.expon(scale=np.mean(iat)))
```

Figures 12.4 and 12.5 show a couple of the resulting Poisson fits.

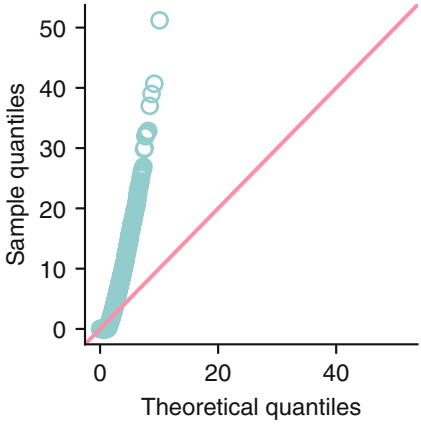

Fig. 12.4 Q–Q plot for a Poisson model for the time of trades in Apple stocks

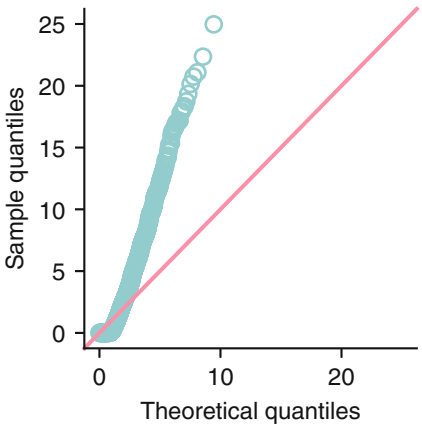

Fig. 12.5 Q–Q plot for a Poisson model for the time of trades in Microsoft stocks

Clearly, the Poisson model for these arrivals is a very poor model. To quantitatively assess the fit, we can calculate the joint likelihood of the combined independent Poisson processes.

```
indepPoissonLogLike = 0

for stock in stocks:
    iat = np.diff(np.insert(stock, 0, 0))
    poissonRate = 1/np.mean(iat)

    indepPoissonLogLike += len(stock)*np.log(poissonRate)
        - poissonRate * np.sum(iat)

numObs = sum(len(stock) for stock in stocks)
BIC_pois = m * np.log(numObs) - 2*indepPoissonLogLike
```

The log-likelihood for the independent Poisson process model was 2670, and the BIC is −5287.

12.3 Independent Hawkes Processes

Let us try to fit an exponential Hawkes model instead.

```
θ_ems = []
indepHawkesLogLike = 0

for stock in stocks:
    θ, logLikelIters = hawkes.exp_em(stock, T, iters=20)
    θ_ems.append(θ)
    indepHawkesLogLike += logLikelIters[-1]

BIC_exp = 3 * m * np.log(numObs) - 2 * indepHawkesLogLike
```

We can see in Fig. 12.6 that the log-likelihoods appear to have converged for each Hawkes fit, so our 20 EM iterations are sufficient.

The resulting log-likelihood for the joint model of independent Hawkes processes is 155,303 and the BIC is −310,446. Numerically, this is a huge improvement over the independent Poisson model. To assess the fits visually, we produce the Q–Q plots of the time-shifted data.

```
for k in range(m):
    timeShifted = hawkes.exp_hawkes_compensators(
        stocks[k], θ_ems[k])
    iat = np.diff(np.insert(timeShifted, 0, 0))
    qqplot(iat, dist=stats.expon)
```

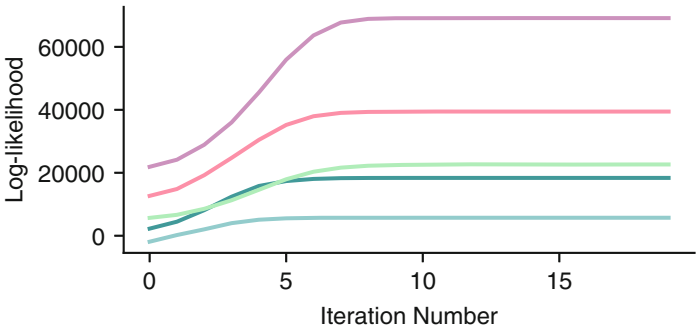

Fig. 12.6 Log-likelihoods across the EM iterations for each stock

The Q–Q plots in Figs. 12.7 and 12.8 show that the Hawkes fit improved upon the Poisson, especially for the periods of heavy trading (the kink in the bottom-left of the plots has disappeared). Neither model does a particularly convincing job of modelling the larger interarrival times (the top-right of the plots) corresponding to the periods of less trading activity.

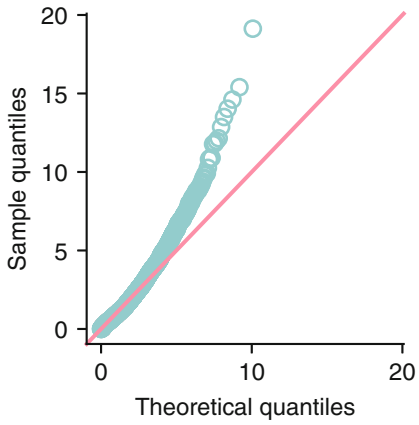

Fig. 12.7 Q–Q plot for a Hawkes model for the time of trades in Apple stocks

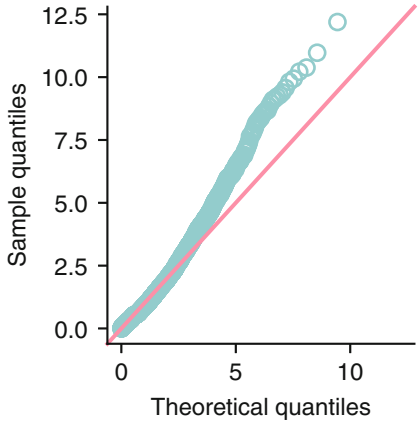

Fig. 12.8 Q–Q plot for a Hawkes model for the time of trades in Microsoft stocks

12.4 Mutually Exciting Hawkes Processes

To create a mutually exciting Hawkes model, we first need to combine the data. Each point in the mutually exciting model is a pair (t_i, d_i), where t_i is the time of

the arrival and d_i is the number of the process which it arrived to. In this case, d_i will just be a number to distinguish which stock was traded, like $\{0 = \text{Amazon}, 1 = \text{Apple}, \ldots, 5 = \text{Twitter}\}$.

```
allTrades = []

for k in range(m):
    for t_i in stocks[k]:
        allTrades.append((t_i, k))

allTrades = sorted(allTrades)

tradeTimes = np.array([t_i for (t_i, d_i) in allTrades])
tradeIds = np.array([d_i for (t_i, d_i) in allTrades])
```

Next, to perform MLE by numerically maximising the likelihood, we must select a starting point for the optimiser. This step is crucial. Numerically optimising a function can be an extremely fiddly business, especially with a multidimensional argument! Even though the class of mutually exciting Hawkes contains the independent Hawkes model which we fitted above, with a bad starting value the MLE fit for the mutually exciting Hawkes can be *worse* than the independent Hawkes model. As we already have the EM-fitted $\boldsymbol{\theta}$'s for each independent Hawkes, we can use this independent model as a starting point for the MLE.

```
λ_indep = np.array([θ_em[0] for θ_em in θ_ems])
α_indep = np.array([θ_em[1] for θ_em in θ_ems])
β_indep = np.array([θ_em[2] for θ_em in θ_ems])

θ_mutual_start = (λ_indep, np.diag(α_indep), β_indep)
```

This should avoid the problem noted above, where the optimiser finds a worse fit than the independent Hawkes model. However, when the interaction terms are all zero, this starting point behaves like a local maxima and the optimiser can have a hard time leaving this point. So we modify this starting point slightly by taking 1% of the $\boldsymbol{\alpha}$ matrix's diagonal and spreading it over the empty parts of the matrix.

```
m = len(λ)
α_dep = 0.99*np.diag(α_indep) + 0.01*α_indep*np.ones((m,m))
θ_mutual_start = (λ_indep, α_dep, β_indep)
```

With this, we can run the MLE.

```
θ_mutual_mle, mutLogLike = hawkes.mutual_exp_mle(
    tradeTimes, tradeIds, T, θ_mutual_start)
```

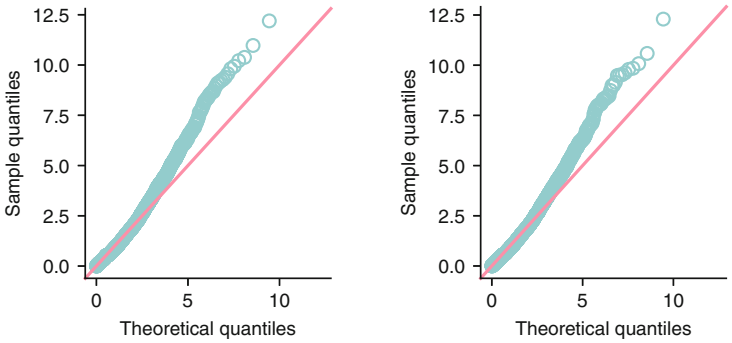

Fig. 12.9 Q–Q plots for the time of trades in Microsoft stocks. The left subplot is the result of an independent Hawkes model, while the right is from a mutually exciting Hawkes model

The resulting log-likelihood is 156,796 which is a little higher than the independent Hawkes log-likelihood 155,303. We can compute the BIC for each model:

```
BIC_uni = 3 * m * np.log(len(tradeTimes))
            - 2 * indepHawkesLogLike
BIC_multi = (m + m**2 + m) * np.log(len(tradeTimes))
            - 2 * mutualHawkesLogLike
```

As the mutually exciting BIC of $-313{,}218$ is smaller than the independent Hawkes BIC of $-310{,}446$, we could conclude that the mutually exciting model is the better model. That is, the extra parameters introduced by the mutual model produce a worthwhile increase in the likelihood to justify the extra complexity.

While the BIC favours the mutually exciting model, the Q–Q plots (e.g. Fig. 12.9) are nearly indiscernible difference between the two Hawkes models.

```
Λs = hawkes.mutual_exp_hawkes_compensators(
        tradeTimes, tradeIds, θ_mutual_mle)

for k in range(m):
    timeShifted = Λs[tradeIds == k,k]
    iat = np.diff(np.insert(timeShifted, 0, 0))
    qqplot(iat, dist=stats.expon)
```

However, we can still investigate the resulting fit for interesting conclusions. To assess the stationary of the resulting fit, we calculate the spectral radius of $\boldsymbol{\alpha}/\boldsymbol{\beta}$:

```
λ, α, β = θ_mutual_mle
spectralRadius = np.max(np.abs(np.linalg.eigvals(α / β)))
```

The radius is 0.67, and as this is less than 1 the resulting process is stable. Furthermore the $\boldsymbol{\alpha}$ values are given by:

$$\boldsymbol{\alpha} = \begin{pmatrix} 1403.63 & 26.98 & 17.05 & 23.51 & 16.38 \\ 4.49 & 2148.71 & 9.04 & 22.42 & 5.43 \\ 8.67 & 32.86 & 1480.87 & 20.92 & 8.19 \\ 8.32 & 48.62 & 14.41 & 991.39 & 11.06 \\ 1.98 & 7.93 & 4.8 & 4.38 & 1055.18 \end{pmatrix}.$$

From this $\boldsymbol{\alpha}$ matrix, one can see which company's trades have more or less of an impact on the trades of the other companies.

12.5 Discussion and Literature Review

This section reviews primarily the work of Aït-Sahalia, et al. [1] and Filimonov and Sornette [28]. It assumes the reader is familiar with mathematical finance and the use of stochastic differential equations. We refer the reader to [64] for further references.

12.5.1 Financial Contagion

A major domain for self- and mutually exciting processes is financial analysis. Frequently it is seen that large movements in a major stock market propagate in foreign markets as a process called *financial contagion*. Examples of this phenomenon are clearly visible in historical series of asset prices.

The 'Hawkes diffusion model' introduced by [1] is an attempt to extend previous models of stock prices to include financial contagion. Modern models for stock prices are typically built upon the model popularised by [9] where the log returns on the stock follow geometric Brownian motion. While this seminal paper was lauded by the economics community, the model inadequately captured the 'fat tails' of the return distribution and so was not commonly used by traders [33]. Merton [49] attempted to incorporate heavy tails by including a Poisson jump process to model booms and crashes in the stock returns; this model is often called Merton diffusion model. The Hawkes diffusion model extends this model by replacing the Poisson jump process with a mutually exciting Hawkes process, so that crashes can self-excite and propagate in a market and between global markets.

The basic Hawkes diffusion model describes the log returns of m assets $\{X_1(\cdot), \ldots, X_m(\cdot)\}$ where each asset $i = 1, \ldots, m$ has associated expected return $\mu_i \in \mathbb{R}$, constant volatility $\sigma_i \in \mathbb{R}^+$, and standard Brownian motion $(W_i^X(t) : t \geqslant 0)$. The Brownian motions have constant correlation coefficients $\{\varrho_{i,j} : i, j =$

$1, \ldots, m\}$. Jumps are added by a self- and mutually exciting Hawkes process (as per Definition 3.3 with some selection of constants $\alpha_{\cdot,\cdot}$ and $\beta_{\cdot,\cdot}$) with stochastic jump sizes $(Z_i(t) : t \geqslant 0)$. The asset dynamics are then assumed to satisfy

$$\mathrm{d}X_i(t) = \mu_i \,\mathrm{d}t + \sigma_i \,\mathrm{d}W_i^X(t) + Z_i(t)\,\mathrm{d}N_i(t)\,.$$

The general Hawkes diffusion model replaces the constant volatilities with stochastic volatilities $\{V_1(\cdot), \ldots, V_m(\cdot)\}$ specified by the Heston model. Each asset $i = 1, \ldots, m$ has a: long term mean volatility $\theta_i > 0$, rate of returning to this mean $\kappa_i > 0$, volatility of the volatility $\nu_i > 0$, and standard Brownian motion $(W_i^V(t) : t \geqslant 0)$. Correlation between the $W_\cdot^X(\cdot)$'s is optional, yet the effect would be dominated by the jump component. Then the full dynamics are captured by

$$\mathrm{d}X_i(t) = \mu_i \,\mathrm{d}t + \sqrt{V_i(t)}\,\mathrm{d}W_i^X(t) + Z_i(t)\,\mathrm{d}N_i(t)\,,$$

$$\mathrm{d}V_i(t) = \kappa_i(\theta_i - V_i(t))\,\mathrm{d}t + \nu_i\sqrt{V_i(t)}\,\mathrm{d}W_i^V(t)\,.$$

However the added realism of the Hawkes diffusion model comes at a high price. The constant volatility model requires $5m+3m^2$ parameters to be fit (assuming $Z_i(\cdot)$ is characterised by two parameters) and the stochastic volatility extension requires an extra $3m$ parameters (assuming $\forall i, j = 1, \ldots, m$ that $\mathbb{E}[W_i(\cdot)^V W_j(\cdot)^V] = 0$). In [1] hypothesis tests reject the Merton diffusion model in favour of the Hawkes diffusion model; however, there are no tests for overfitting the data (for example, Akaike or Bayesian information criterion comparisons). Remember that John Von Neumann (reputedly) claimed that 'with four parameters I can fit an elephant' [25].

For computational necessity, the authors made a number of simplifying assumptions to reduce the number of parameters to fit (such as the background intensity of crashes is the same for all markets). Even so, the Hawkes diffusion model was only able to be fitted for pairs of markets ($m = 2$) instead of for the globe as a whole. Since the model was calibrated to daily returns of market indices, historical data was easily available (for example, from Google or Yahoo! finance); care had to be taken to convert timezones and handle the different market opening and closing times. The parameter estimation method used by [1] was the generalised method of moments; however, the theoretical moments derived satisfy long and convoluted equations.

12.5.2 Mid-Price Changes and High-Frequency Trading

A simpler system to model is a single stock's price over time, though there are many different prices to consider. For each stock one could use: the last transaction price, the best ask price, the best bid price, or the mid-price (defined as the average of best ask and best bid prices). The last transaction price includes inherent microstructure

noise (for example, the bid–ask bounce), and the best ask and bid prices fail to represent the actions of both buyers and sellers in the market.

Filimonov and Sornette [28] model the mid-price changes over time as a Hawkes process. In particular, they look at long term trends of the (estimated) branching ratio. In this context, n represents the proportion of price moves that are not due to external market information but simply reactions to other market participants. This ratio can be seen as the quantification of the principle of economic reflexivity. The authors conclude that the branching ratio has increased dramatically from 30% in 1998 to 70% in 2007.

Later that year [47] critiqued the test procedure used in this analysis. Filimonov and Sornette [28] had worked with a dataset with timestamps accurate to a second, and this often led to multiple arrivals nominally at the same time (which is an impossible event for simple point processes). Fake precision was achieved by adding $\mathsf{Unif}(0, 1)$ random fractions of seconds to all timestamps, a technique also used by [10]. Lorenzen found that this method added an element of smoothing to the data which gave it a better fit to the model than the actual millisecond precision data. The randomisation also introduced bias to the Hawkes process parameter estimates, particularly of α and β. Lorenzen formed a crude measure of high-frequency trading activity leading to an interesting correlation between this activity and n over the observed period.

The literature on Hawkes processes in finance has been growing and therefore, only a brief description is provided here. Among one of topics that were not touched on is the applications of Hawkes processes to market microstructure modelling, for example the paper by Bacry et al. [4].

Appendix A
Supplementary Material

A.1 Preliminary Background Concepts

In this section, we briefly recall a handful of elementary distributions that are used in this book.

Arguably the simplest random variable is the Bernoulli trial, representing the success or failure of a random experiment, a single measurement with binary outcome.

Definition A.1 (Bernoulli Distribution) A random variable X is said to have a *Bernoulli distribution* with parameter $p \in [0, 1]$ if $\mathbb{P}(X = 1) = p$ and $\mathbb{P}(X = 0) = 1 - p$. We write $X \sim \mathsf{Ber}(p)$ to denote this. ◇

Definition A.2 (Beta Distribution) A random variable X is said to have a *beta distribution* with positive parameters a and b if it has PDF

$$f(x) = \frac{1}{B(a, b)} x^{a-1}(1 - x)^{b-1}, \qquad 0 < x < 1,$$

with

$$B(a, b) = \int_0^1 x^{a-1}(1 - x)^{b-1} dx.$$

We write $X \sim \mathsf{Beta}(a, b)$ to denote this. ◇

Counting the total number of successes from a collection of independent Bernoulli trials gives us the binomial distribution.

Definition A.3 (Binomial Distribution) A random variable X is said to have a *Binomial distribution* with parameters $n \in \{1, 2, \dots\}$ and $p \in [0, 1]$ if

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

P. J. Laub et al., *The Elements of Hawkes Processes*,
https://doi.org/10.1007/978-3-030-84639-8

$$\mathbb{P}(X = x) = \binom{n}{x} p^x (1-p)^{n-x}\,, \qquad x = 0, 1, \ldots, n\,.$$

We write $X \sim \mathsf{Bin}(n, p)$ to denote this. ◇

Definition A.4 (Erlang Distribution) A random variable X is said to have an *Erlang distribution* with parameters $n \in \{1, 2, \ldots\}$ and $\lambda > 0$ if it has PDF

$$f(x) = \frac{\lambda^n x^{n-1}}{(n-1)!} \mathrm{e}^{-\lambda x}\,.$$

We write $X \sim \mathsf{Erl}(n, \lambda)$ to denote this. ◇

Definition A.5 (Exponential Distribution) A random variable X is said to have an *exponential distribution* with rate $\lambda > 0$ if it has PDF

$$f(x) = \lambda \mathrm{e}^{-\lambda x}\,, \qquad x \geqslant 0\,.$$

We write $X \sim \mathsf{Exp}(\lambda)$ to denote this. ◇

Definition A.6 (Memoryless Property) A non-negative random variable X is said to have the *memoryless property* if, for all $s, t > 0$,

$$\mathbb{P}(X > s + t \mid X > s) = \mathbb{P}(X > t)\,.$$

◇

Any exponentially distributed random variable possesses the memoryless property (as does any geometrically distributed random variable).

Definition A.7 (Normal Distribution) A random variable X is said to have a *normal distribution* with parameters $\mu \in \mathbb{R}$ and $\sigma^2 > 0$ if it has PDF

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-\frac{1}{2\sigma^2}(x-\mu)^2}\,, \qquad x \in \mathbb{R}.$$

We write $X \sim \mathsf{Normal}(\mu, \sigma^2)$ to denote this. ◇

A distribution that has made repeated appearance throughout is the Poisson Normal and Gamma distributions.

Definition A.8 (Poisson Distribution) A random variable X is said to have a *Poisson distribution* with rate $\lambda > 0$ if it has probability mass function

$$\mathbb{P}(X = x) = \mathrm{e}^{-\lambda} \frac{\lambda^x}{x!}\,, \qquad x = 0, 1, 2, \ldots$$

We write $X \sim \mathsf{Poi}(\lambda)$ to denote this. ◇

The Poisson distribution arises naturally as the limit of a large number of Bernoulli trials each with small probability, as follows.

Theorem A.1 (Poisson as Limit of Binomial Random Variables) *Fix* $\lambda > 0$ *and define* $X_n \sim \mathsf{Bin}(n, \min\{\lambda/n, 1\})$ *for* $n = 1, 2, \ldots$. *Then* X_n *converges in distribution to* $X \sim \mathsf{Poi}(\lambda)$ *as* $n \to \infty$. ◇

Proof This is easy to see using probability generating functions. For each binomial random variable, the corresponding probability generating function is given by

$$G_{X_n}(z) = \mathbb{E}[z^{X_n}] = (1 - p_n + p_n z)^n ,$$

where $p_n = \min\{\lambda/n, 1\}$. The limiting probability generating function is easily determined as

$$\lim_{n\to\infty} G_{X_n}(z) = \lim_{n\to\infty} (1 - \lambda(1 - z)/n)^n = \exp(-\lambda(1 - z)) .$$

This last expression is the probability generating function of a $\mathsf{Poi}(\lambda)$ random variable, and so convergence in distribution has been established.

This result illuminates an alternate name for the Poisson distribution: the 'law of small numbers'.

The Poisson distribution is closely connected to the exponential distribution through the *Poisson process*, which will be discussed shortly. We briefly recall the exponential distribution and a fundamental property it possesses.

Definition A.9 (Uniform Distribution) A random variable X is said to have a *uniform distribution* on the interval $[a, b]$ if it has PDF

$$f(x) = \frac{1}{b - a} , \qquad a < x < b.$$

We write $X \sim \mathsf{Unif}(a, b)$ to denote this. ◇

A.2 Additional Proof Details

In this appendix, we collect additional details elided from the proof of Theorem 3.2.

A.2.1 Supplementary to Theorem 3.2 (Part I)

$$
\begin{aligned}
R(\tau) &= \mathbb{E}\left[\frac{\mathrm{d}N(t)}{\mathrm{d}t}\left(\lambda + \int_{-\infty}^{t+\tau} \mu(t+\tau-s)\,\mathrm{d}N(s)\right)\right] - \overline{\lambda^*}^2 \\
&= \lambda\mathbb{E}\left[\frac{\mathrm{d}N(t)}{\mathrm{d}t}\right] + \mathbb{E}\left[\frac{\mathrm{d}N(t)}{\mathrm{d}t}\left(\int_{-\infty}^{t+\tau} \mu(t+\tau-s)\,\mathrm{d}N(s)\right)\right] - \overline{\lambda^*}^2 \\
&= \lambda\overline{\lambda^*} + \mathbb{E}\left[\frac{\mathrm{d}N(t)}{\mathrm{d}t}\int_{-\infty}^{t+\tau} \mu(t+\tau-s)\,\mathrm{d}N(s)\right] - \overline{\lambda^*}^2.
\end{aligned}
$$

Introduce a change of variable $v = s - t$ and multiply by $\frac{\mathrm{d}v}{\mathrm{d}v}$:

$$
\begin{aligned}
R(\tau) &= \lambda\overline{\lambda^*} + \mathbb{E}\left[\int_{-\infty}^{\tau} \mu(\tau-v)\frac{\mathrm{d}N(t)}{\mathrm{d}t}\frac{\mathrm{d}N(t+v)}{\mathrm{d}v}\,\mathrm{d}v\right] - \overline{\lambda^*}^2 \\
&= \lambda\overline{\lambda^*} + \int_{-\infty}^{\tau} \mu(\tau-v)\mathbb{E}\left[\frac{\mathrm{d}N(t)}{\mathrm{d}t}\frac{\mathrm{d}N(t+v)}{\mathrm{d}v}\right]\mathrm{d}v - \overline{\lambda^*}^2.
\end{aligned}
$$

The expectation is (a shifted) $R^{(c)}(v)$. Substitute that and (3.10) in

$$
\begin{aligned}
R(\tau) &= \lambda\overline{\lambda^*} + \int_{-\infty}^{\tau} \mu(\tau-v)\left(R^{(c)}(v) + \overline{\lambda^*}^2\right)\mathrm{d}v - \overline{\lambda^*}^2 \\
&= \lambda\overline{\lambda^*} + \int_{-\infty}^{\tau} \mu(\tau-v)\left(\overline{\lambda^*}\delta(v) + R(v)\right)\mathrm{d}v + \overline{\lambda^*}^2\int_{-\infty}^{\tau} \mu(\tau-v)\,\mathrm{d}v - \overline{\lambda^*}^2 \\
&= \lambda\overline{\lambda^*} + \overline{\lambda^*}\mu(\tau) + \int_{-\infty}^{\tau} \mu(\tau-v)R(v)\,\mathrm{d}v + n\overline{\lambda^*}^2 - \overline{\lambda^*}^2 \\
&= \overline{\lambda^*}\mu(\tau) + \int_{-\infty}^{\tau} \mu(\tau-v)R(v)\,\mathrm{d}v + \overline{\lambda^*}\left(\lambda - (1-n)\overline{\lambda^*}\right).
\end{aligned}
$$

Using (3.9) yields

$$
\lambda - (1-n)\overline{\lambda^*} = \lambda - (1-n)\frac{\lambda}{1-n} = 0\,.
$$

$$
\therefore\ R(\tau) = \overline{\lambda^*}\mu(\tau) + \int_{-\infty}^{\tau} \mu(\tau-v)R(v)\,\mathrm{d}v\,.
$$

A.2.2 Supplementary to Theorem 3.2 (Part II)

Split the right-hand side of the equation into three functions g_1, g_2, and g_3:

$$R(\tau) = \underbrace{\overline{\lambda^*}\mu(\tau)}_{g_1(\tau)} + \underbrace{\int_0^\infty \mu(\tau+v)R(v)\,\mathrm{d}v}_{g_2(\tau)} + \underbrace{\int_0^\tau \mu(\tau-v)R(v)\,\mathrm{d}v}_{g_3(\tau)}\,. \tag{A.1}$$

Taking the Laplace transform of each term gives

$$\mathscr{L}\left\{g_1(\tau)\right\}(s) = \int_0^s \mathrm{e}^{-s\tau}\overline{\lambda^*}\alpha\mathrm{e}^{-\beta\tau}\,\mathrm{d}\tau = \frac{\alpha}{s+\beta}\overline{\lambda^*},$$

$$\begin{aligned}
\mathscr{L}\left\{g_2(\tau)\right\}(s) &= \int_0^\infty \mathrm{e}^{-s\tau}\int_0^\infty \alpha\mathrm{e}^{-\beta(\tau+v)}R(v)\,\mathrm{d}v\,\mathrm{d}\tau \\
&= \alpha\int_0^\infty \mathrm{e}^{-\beta v}R(v)\int_0^\infty \mathrm{e}^{-\tau(s+\beta)}\,\mathrm{d}\tau\,\mathrm{d}v \\
&= \frac{\alpha}{s+\beta}\int_0^\infty \mathrm{e}^{-\beta v}R(v)\,\mathrm{d}v \\
&= \frac{\alpha}{s+\beta}\mathscr{L}\left\{R\right\}(\beta)\,,
\end{aligned}$$

and

$$\mathscr{L}\left\{g_3(\tau)\right\}(s) = \mathscr{L}\left\{\mu(\tau)\right\}(s)\mathscr{L}\left\{R(\tau)\right\}(s) = \frac{\alpha}{s+\beta}\mathscr{L}\left\{R(\tau)\right\}(s)\,.$$

Therefore, the Laplace transform of (A.1) is

$$\mathscr{L}\left\{R(\tau)\right\}(s) = \frac{\alpha}{s+\beta}\left(\overline{\lambda^*} + \mathscr{L}\left\{R(\tau)\right\}(\beta) + \mathscr{L}\left\{R(\tau)\right\}(s)\right)\,. \tag{A.2}$$

Substituting $s = \beta$ and rearranging give that

$$\mathscr{L}\left\{R(\tau)\right\}(\beta) = \frac{\alpha\overline{\lambda^*}}{2(\beta-\alpha)}\,.$$

So, substituting the value of $\mathscr{L}\left\{R(\tau)\right\}(\beta)$ into (A.2) means

$$\mathscr{L}\left\{R(\tau)\right\}(s) = \frac{\alpha}{s+\beta}\left(\overline{\lambda^*} + \frac{\alpha\overline{\lambda^*}}{2(\beta-\alpha)} + \mathscr{L}\left\{R(\tau)\right\}(s)\right)$$

$$\Rightarrow \mathscr{L}\left\{R(\tau)\right\}(s) = \frac{\frac{\alpha}{s+\beta}\left(\overline{\lambda^*} + \frac{\alpha\overline{\lambda^*}}{2(\beta-\alpha)}\right)}{1-\frac{\alpha}{s+\beta}} = \frac{\alpha\overline{\lambda^*}(2\beta-\alpha)}{2(\beta-\alpha)(s+\beta-\alpha)}\,.$$

References

1. Aït-Sahalia, Y., Cacho-Diaz, J., & Laeven, R. J. A. (2015). Modeling financial contagion using mutually exciting jump processes. *Journal of Financial Economics, 117*(3), 585–606.
2. Asmussen, S. (2003). *Applied Probability and Queues*. Applications of Mathematics: Stochastic Modelling and Applied Probability (2ond edn.). Springer.
3. Azizpour, S., Giesecke, K., & Schwenkler, G. (2018). Exploring the sources of default clustering. *Journal of Financial Economics, 129*(1), 154–183.
4. Bacry, E., Delattre, S., Hoffmann, M., & Muzy, J.-F. (2013). Modelling microstructure noise with mutually exciting point processes. *Quantitative Finance, 13*(1), 65–77.
5. Bartlett, M. S. (1963). The spectral analysis of point processes. *Journal of the Royal Statistical Society. Series B (Methodological), 25*(2), 264–296.
6. Bartlett, M. S. (1963). Statistical estimation of density functions. *Sankhyā: The Indian Journal of Statistics, Series A, 25*(3), 245–254.
7. Bartlett, M. S. (1964). The spectral analysis of two-dimensional point processes. *Biometrika, 51*(3/4), 299–311.
8. Bauwens, L., & Hautsch, N. (2009). *Modelling Financial High Frequency Data Using Point Processes* (pp. 953–979). Berlin, Heidelberg: Springer.
9. Black, F., & Scholes, M. (1973). The pricing of options and corporate liabilities. *The Journal of Political Economy, 81*(3), 637–654.
10. Bowsher, C. G. (2007). Modelling security market events in continuous time: intensity based, multivariate point process models. *Journal of Econometrics, 141*(2), 876–912.
11. Brémaud, P., & Massoulié, L. (1996). Stability of nonlinear Hawkes processes. *The Annals of Probability, 24*(3), 1563–1588.
12. Brown, E., Barbieri, R., Ventura, V., Kass, R., & Frank, L. (2002). The time-rescaling theorem and its application to neural spike train data analysis. *Neural Computation, 14*(2), 325–346.
13. Brown, T. C., & Nair, G. (1988). A simple proof of the multivariate random time change theorem for point processes. *Journal of Applied Probability*, 210–214.
14. Carstensen, L. (2010). *Hawkes processes and combinatorial transcriptional regulation*. Ph.D. thesis, University of Copenhagen.
15. Chatalbashev, V., Liang, Y., Officer, A., & Trichakis, N. (2007). Exciting times for trade arrivals. http://users.iems.northwestern.edu/~armbruster/2007msande444/report1a.pdf. Stanford University MS&E 444 group project submission, retrieved on 10 Feb 2015.
16. Cox, D. R. (1955). Some statistical methods connected with series of events. *Journal of the Royal Statistical Society. Series B (Methodological), 17*(2), 129–164.
17. Cox, D. R., & Lewis, P. A. W. (1966). *The Statistical Analysis of Series of Events*. Monographs on Applied Probability and Statistics. London: Chapman and Hall.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

P. J. Laub et al., *The Elements of Hawkes Processes*,
https://doi.org/10.1007/978-3-030-84639-8

18. Da Fonseca, J., & Zaatour, R. (2014). Hawkes process: Fast calibration, application to trade clustering, and diffusive limit. *Journal of Futures Markets, 34*(6), 548–579.
19. Daley, D. J., & Vere-Jones, D. (2003). *An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods*. Springer.
20. Dassios, A., & Zhao, H. (2013). Exact simulation of Hawkes process with exponentially decaying intensity. *Electronic Communications in Probability, 18*(62), 1–13.
21. Davis, M. H. A. (1984). Piecewise-deterministic Markov processes: a general class of nondiffusion stochastic models. *Journal of the Royal Statistical Society. Series B. Methodological, 46*(3), 353–388.
22. Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B, 39*(1), 1–38.
23. Devroye, L. (1986). *Non-Uniform Random Variate Generation*. New York: Springer-Verlag.
24. Durbin, J. (1961). Some methods of constructing exact tests. *Biometrika, 53*(3/4), 41–55.
25. Dyson, F. (2004). A meeting with Enrico Fermi. *Nature, 427*(6972), 297–297.
26. Efron, B., & Hastie, T. (2016). *Computer Age Statistical Inference*. Cambridge University Press.
27. Embrechts, P., Liniger, T., & Lin, L. (2011). Multivariate Hawkes processes: an application to financial data. *Journal of Applied Probability, 48A*, 367–378. Special volume: a Festschrift for Søren Asmussen.
28. Filimonov, V., & Sornette, D. (2012). Quantifying reflexivity in financial markets: toward a prediction of flash crashes. *Physical Review E, 85*(5), 056108.
29. Giesecke, K., & Tomecek, P. (2005). Dependent events and changes of time. https://stanford.box.com/s/zh52e1nkuxgvj23w1adurfm4bpvhpyj0. Working paper, retrieved on 10 Feb 2015.
30. Grimmett, G., & Stirzaker, D. (2001). *Probability and Random Processes*. Oxford University Press.
31. Hansen, L. P. (1982). Large sample properties of generalized method of moments estimators. *Econometrica, 50*(4), 1029–1054.
32. Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika, 57*(1), 97–109.
33. Haug, E. G. & Taleb, N. N. (2014). Why we have never used the Black–Scholes–Merton option pricing formula. *Wilmott Magazine, 71*.
34. Hautsch, N. (2011). *Econometrics of Financial High-Frequency Data*. Springer.
35. Hawkes, A. G. (1971). Spectra of some self-exciting and mutually exciting point processes. *Biometrika, 58*(1), 83–90.
36. Hawkes, A. G. (1971). Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society. Series B (Methodological), 33*(3), 438–443.
37. Hawkes, A. G., & Oakes, D. (1974). A cluster process representation of a self-exciting process. *Journal of Applied Probability, 11*(3), 493–503.
38. Jalilian, A. (2019). ETAS: an R package for fitting the space-time ETAS model to earthquake data. *Journal of Statistical Software, 88*(1), 1–39.
39. Kim, S.-H., & Whitt, W. (2015). The power of alternative Kolmogorov-Smirnov tests based on transformations of the data. *ACM Transactions on Modeling and Computer Simulation, 25*(4), 1–22.
40. Knuth, D. E. (2014). *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Addison-Wesley Professional.
41. Kroese, D., Taimre, T., & Botev, Z. I. (2011). *Handbook of Monte Carlo Methods*. Wiley.
42. Lewis, P. A. W. (1964). A branching Poisson process model for the analysis of computer failure patterns. *Journal of the Royal Statistical Society. Series B (Methodological), 26*(3), 398–456.
43. Lewis, P. A. W. (1965). Some results on tests for Poisson processes. *Biometrika, 52*(1/2), 67–77.
44. Lewis, P. A. W. (1970). Remarks on the theory, computation and application of the spectral analysis of series of events. *Journal of Sound and Vibration, 12*(3), 353–375.
45. Lewis, P. A. W., & Shedler, G. S. (1979). Simulation of nonhomogeneous Poisson processes by thinning. *Naval Research Logistics Quarterly, 26*(3), 403–413.

46. Liniger, T. J. (2009). *Multivariate Hawkes Processes*. Ph.D. thesis, Swiss Federal Institute of Technology Zurich (ETH Zurich).
47. Lorenzen, F. (2012). *Analysis of Order Clustering Using High Frequency Data: A Point Process Approach*. Ph.D. thesis, Tilburg School of Economics and Management, Finance Department.
48. Medvegyev, P. (2007). *Stochastic Integration Theory* (vol. 14). Oxford University Press.
49. Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics, 3*(1), 125–144.
50. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics, 21*(6), 1087–1092.
51. Meyer, P.-A. (1971). Demonstration simplifiee d'un theoreme de Knight. In *Séminaire de Probabilités V Université de Strasbourg* (pp. 191–195).
52. Mohler, G. O., Short, M. B., Brantingham, P. J., Schoenberg, F. P., & Tita, G. E. (2011). Self-exciting point process modeling of crime. *Journal of the American Statistical Association, 106*(493), 100–108.
53. Ogata, Y. (1978). The asymptotic behaviour of maximum likelihood estimators for stationary point processes. *Annals of the Institute of Statistical Mathematics, 30*(1), 243–261.
54. Ogata, Y. (1981). On Lewis' simulation method for point processes. *IEEE Transactions on Information Theory, 27*(1), 23–31.
55. Ogata, Y. (1988). Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical Association, 83*(401), 9–27.
56. Ogata, Y. (1999). Seismicity analysis through point-process modeling: A review. *Pure and Applied Geophysics, 155*(2/4), 471–507.
57. Ozaki, T. (1979). Maximum likelihood estimation of Hawkes' self-exciting point processes. *Annals of the Institute of Statistical Mathematics, 31*(1), 145–155.
58. Pai, J. S. (1997). Generating random variates with a given force of mortality and finding a suitable force of mortality by theoretical quantile-quantile plots. *Actuarial Research Clearing House, 1*, 293–312.
59. Papangelou, F. (1972). Integrability of expected increments of point processes and a related random change of scale. *Transactions of the American Mathematical Society, 165*, 483–506.
60. Raftery, A. E., & Lewis, S. M. (1995). The number of iterations, convergence diagnostics and generic Metropolis algorithms. In *In Practical Markov Chain Monte Carlo (W.R. Gilks, D.J. Spiegelhalter)* (pp. 115–130)
61. Rasmussen, J. G. (2013). Bayesian inference for Hawkes processes. *Methodology and Computing in Applied Probability, 15*(3), 623–642.
62. Rasmussen, J. G. (2018). Lecture notes: Temporal point processes and the conditional intensity function. arXiv:1806.00221.
63. Rubin, I. (1972). Regular point processes and their detection. *IEEE Transactions on Information Theory, 18*(5), 547–557.
64. Tankov, P. (2003). *Financial Modelling with Jump Processes*. Chapman & Hall/CRC Financial Mathematics Series. Taylor & Francis.
65. Veen, A., & Schoenberg, F. P. (2008). Estimation of space-time branching process models in seismology using an EM-type algorithm. *Journal of the American Statistical Association, 103*(482), 614–624.
66. Watanabe, S. (1964). On discontinuous additive functionals and Lévy measures of a Markov process. *Japanese Journal of Mathematics, 34*(53-70), 82.
67. Zhu, L. (2013). Central limit theorem for nonlinear Hawkes processes. *Journal of Applied Probability, 50*(3), 760–771.
68. Zipkin, J. R., Schoenberg, F. P., Coronges, K., & Bertozzi, A. L. (2016). Point-process models of social network interactions: Parameter estimation and missing data recovery. *European Journal of Applied Mathematics, 27*(3), 502–529.