A Tutorial on Hawkes Processes for Events in Social Media

Chapter · December 2017

DOI: 10.1145/3122865.3122874

4 authors, including:

Marian-Andrei Rizoiu
University of Technology Sydney
105 PUBLICATIONS **1,380** CITATIONS

SEE PROFILE

Swapnil Mishra
Australian National University
104 PUBLICATIONS **12,419** CITATIONS

SEE PROFILE

Lexing Xie
Australian National University
172 PUBLICATIONS **7,141** CITATIONS

SEE PROFILE

1 A Tutorial on Hawkes Processes for Events in Social Media

Marian-Andrei Rizoiu, The Australian National University; Data61, CSIRO
Young Lee, Data61, CSIRO; The Australian National University
Swapnil Mishra, The Australian National University; Data61, CSIRO
Lexing Xie, The Australian National University; Data61, CSIRO

This chapter provides an accessible introduction for point processes, and especially Hawkes processes, for modeling discrete, inter-dependent events over continuous time. We start by reviewing the definitions and the key concepts in point processes. We then introduce the Hawkes process, its event intensity function, as well as schemes for event simulation and parameter estimation. We also describe a practical example drawn from social media data – we show how to model retweet cascades using a Hawkes self-exciting process. We presents a design of the memory kernel, and results on estimating parameters and predicting popularity. The code and sample event data are available as an online appendix.

1.1 Introduction

Point processes are collections of random points falling in some space, such as time and location. Point processes provide the statistical language to describe the timing and properties of events. Problems that fit this setting span a range of application domains. In finance, an event can represent a buy or a sell transaction

Figure 1.1 An point process, showing tweets about a Gaming video on Youtube. The first 10 events are shown. They correspond to the first 10 tweets in the diffusion, the time stamps of which are indicated by dashed vertical lines. An event with hollow tip denote a retweet of a previous tweet.

on the stock market that influences future prices and volumes of such transactions. In geophysics, an event can be an earthquake that is indicative of the likelihood of another earthquake in the vicinity in the immediate future. In ecology, event data consist of a set of point locations where a species has been observed. In the analysis of online social media, events can be user actions over time, each of which have a set of properties such as user influence, topic of interest, and connectivity of the surrounding network.

Fig. 1.1 depicts an example point process – a retweet cascade about a Gaming Youtube video (YoutubeID ---*U8lzusKE*). Here each tweet is an event, that happens at a certain point in continuous time. Three of the events depicted in Fig. 1.1 are depicted using hollow tips – they are retweets of a previous tweet, or the act of one user re-sharing the content from another user. We explicitly observe information diffusion via retweets, however there are other diffusion mechanisms that are not easily observed. These include offline *word-of-mouth* diffusion, or information propagating in emails and other online platforms. One way for modeling the overall information diffusion process is to use so called *self-exciting processes* – in this type of processes the probability of seeing a new event increases due to previous events. Point-process models are useful for answering a range of different questions. These include *explaining* the nature of the underlying process, *simulating* future events, and *predicting* the likelihood and volume of future events.

In Section 1.2, we first review the basic concepts and properties of point processes in general, and of Poisson processes. These provide foundations for defining the Hawkes process. In Section 1.3, we introduce the Hawkes process – including expressions of the event rate and the underlying branching structure. Section 1.4 describes two procedures for sampling from a Hawkes process. One uses thinning, rejection sampling, while the other make use of a novel variable decomposition. Section 1.5 derives the likelihood of a Hawkes process and describes a maximum likelihood estimation procedure for its parameters, given observed event sequence(s). In the last section of this chapter we present an example of estimating Hawkes processes from a retweet event sequence. We introduce the data, the problem formulation, the fitting results, and interpretations of the model. We include code snippets for this example in Sec 1.6.5, the accompanying software and data are included in an online repository.

This chapter aims to provide a self-contained introduction to the fundamental concepts and methods for self-exciting point-processes, with a particular emphasis on the Hawkes process. The goal is for the readers to be able to understand the key mathematical and computational constructs of a point process, formulate their problems in the language of point processes, and use point process in domains including but are not limited to modeling events in social media. The study of point processes has a long history, with discussions of Hawkes process dating back at least to the early 1970s [Hawkes 1971]. Despite the richness of existing literature, we found through our own recent experience in learning and applying Hawkes processes, that a self-contained tutorial centered around problem formulation and applications is still missing. This chapter aims at filling this gap, providing the foundations as well as an example of point processes for social media. Its intended audience are aspiring researchers, beginning PhD students, as well as any technical reader with a special interest in point processes and their practical applications. For in-depth reading, we refer the readers to overview papers and books [Daley and Vere-Jones 2003, Toke 2011] on Hawkes processes. We note that this chapter does not cover other important variants used in the multimedia area, such as self-inhibiting processes [Yang et al. 2015], or non-causal processes (in time or space), such as the Markov point processes [Pham et al. 2016].

1.2 Preliminary: Poisson processes

In this section, we introduce the fundamentals of point processes and its simplest subclass, the Poisson process. These serve as the foundation on which we build, in later sections, the more complex processes, such as the Hawkes point process.

1.2.1 Defining a point process

A point process on the nonnegative real line, where the nonnegative line is taken to represent time, is a random process whose realizations consists of the event times $T_1, T_2, \ldots$ of event times falling along the line. T_i can usually be interpreted as the time of occurrence of the i-th event, and T_i are often referred to as event times.

The equivalent counting process. A counting process N_t is a random function defined on time $t \geq 0$, and take integer values $1, 2, \ldots$. Its value is the number of events of the point process by time t. Therefore it is uniquely determined by a sequence of non-negative random variables T_i, satisfying $T_i < T_{i+1}$ if $T_i \leq \infty$. In other words, N_t *counts* the number of events up to time t, i.e.

$$N_t := \sum_{i \geq 1} \mathbb{1}_{\{t \geq T_i\}} \tag{1.1}$$

Here $\mathbb{1}_{\{\cdot\}}$ is the indicator function that takes value 1 when the condition is true, 0 otherwise. We can see that $N_0 = 0$. N_t is piecewise constant and has jump size of 1 at the event times T_i. It is easy to see that the set of event times $T_1, T_2, \ldots$ and the corresponding counting process are equivalent representations of the underlying point process.

1.2.2 Poisson processes: definition

The simplest class of point process is the Poisson process. It serves as the starting point for more sophisticated point process models, such as the Hawkes processes.

DEFINITION 1. *(Poisson process.) Let $(\tau_i)_{i \geq 1}$ be a sequence of i.i.d. exponential random variables with parameter λ and event times $T_n = \sum_{i=1}^{n} \tau_i$. The process $(N_t, t \geq 0)$ defined by $N_t := \sum_{i \geq 1} \mathbb{1}_{\{t \geq T_i\}}$ is called a Poisson process with intensity λ.*

Event intensity λ. The sequence of τ_j are called the *inter-arrival times*, i.e. the first event occurs at time τ_1, the second occurs at τ_2 after the first, etc. The inter-arrival times τ_i are independent, and each of them follow an exponential distribution with parameter λ. Here, the notation $f_\tau(t)$ denotes the probability density function of random variable τ taking values denoted by t.

$$f_\tau(t) = \begin{cases} \lambda e^{-\lambda t}, & \text{if } t \geq 0 \\ 0, & \text{if } t < 0 \end{cases} \tag{1.2}$$

Here $\lambda > 0$ is a positive constant. The expected value of τ_i can be computed in closed form, as follows:

$$\mathbb{E}_\tau[\tau] = \int_0^\infty t f_\tau(t) dt = \lambda \int_0^\infty t e^{-\lambda t} dt = \left[-te^{-\lambda t}\right]_{t=0}^{t=+\infty} + \int_0^\infty e^{-\lambda t} dt$$
$$= 0 - \left[\frac{1}{\lambda} e^{-\lambda t}\right]_{t=0}^{t=\infty} = \frac{1}{\lambda}. \tag{1.3}$$

Intuitively, events are arriving at an average rate of λ per unit time, since the expected time between event times is λ^{-1}. Hence we say, informally, that the Poisson process has *intensity* λ. In general, the event intensity needs not be constant, but is a function of time, written as $\lambda(t)$. This general case is called a *non-homogeneous Poisson process*, and will be discussed in Sec. 1.2.4.

Arrival times and counting process. The *arrival times*, or the event times, are given by:

$$T_n = \sum_{j=1}^{n} \tau_j, \tag{1.4}$$

where T_n is the time of the n-th arrival. The event times $T_1, T_2, \ldots$ form a random configuration of points on the real line $[0, \infty)$ and N_t counts the number of such ons in the interval $[0, t]$. Consequently, N_t increments by one for each T_i. This can be explicitly written as follows.

$$N_t = \begin{cases} 0, & \text{if } 0 \le t < T_1 \\ 1, & \text{if } T_1 \le t < T_2 \\ 2, & \text{if } T_2 \le t < T_3 \\ \vdots & \\ n, & \text{if } T_n \le t < T_{n+1}, \\ \vdots & \end{cases} \tag{1.5}$$

We observe that N_t is defined so that it is *right continuous with left limits*. The left limit $N_{t-} = \lim_{s \uparrow t} N_s$ exists and $N_{t+} = \lim_{s \downarrow t} N_s$ exists and taken to be N_t.

1.2.3 The memorylessness property of Poisson processes

Being *memoryless* in a point process means that the distribution of future inter-arrival times depends only on relevant information about the current time, but not on information from further in the past. We show that this is the case for Poisson processes.

We compute the probability of observing an inter-arrival time τ longer than a predefined time length t. F_τ is the cumulative distribution function of the random variable τ, which is defined as $F_\tau(t) := \mathbb{P}\{\tau \le t\}$. We have

$$F_\tau(t) := \mathbb{P}(\tau \le t) = \int_0^t \lambda e^{-\lambda x} dx = \left[-e^{\lambda x}\right]_{x=0}^{x=t} = 1 - e^{-\lambda t}, \quad t \ge 0, \tag{1.6}$$

and hence the probability of observing an event at time $\tau > t$ is given by

$$\mathbb{P}(\tau > t) = e^{-\lambda t}, \quad t \ge 0. \tag{1.7}$$

Suppose we were waiting for an arrival of an event, say a tweet, the inter-arrival times of which follow an Exponential distribution with parameter λ. Assume that m time units have elapsed and during this period no events have arrived, i.e. there are no events during the time interval $[0,m]$. The probability that we will have to wait a further t time units given by

$$\begin{aligned}\mathbb{P}(\tau > t+m|\tau > m) &= \frac{\mathbb{P}(\tau > t+m, \tau > m)}{\mathbb{P}(\tau > m)} \\ &= \frac{\mathbb{P}(\tau > t+m)}{\mathbb{P}(\tau > m)} = \frac{e^{-\lambda(t+m)}}{e^{-\lambda m}} = e^{-\lambda t} = \mathbb{P}(\tau > t). \end{aligned} \tag{1.8}$$

In this derivation, we first expand the conditional probability using Bayes rule. The next step follows from the fact that $\tau > m$ always holds when $\tau > t+m$. The last step follows from Eq. (1.7).

Eq. (1.8) denotes the *memorylessness* property of Poisson processes. That is, the probability of having to wait an additional t time units after already having waited m time units is the same as the probability of having to wait t time units when starting at time 0. Putting it differently, if one interprets τ as the time of arrival of an event where τ follows an Exponential distribution, the distribution of $\tau - m$ given $\tau > m$ is the same as the distribution of τ itself.

1.2.4 Non-homogeneous Poisson processes

In Poisson processes, events arrive randomly with the constant intensity λ. This initial model is sufficient for describing simple processes, say the arrival of cars on a street over a short period of time. However, we need to able to vary the event intensity with time in order to describe more complex processes, such as simulating the arrivals of cars during rush hours and off-peak times. In a non-homogeneous Poisson process, the rate of event arrivals is a function of time, i.e. $\lambda = \lambda(t)$.

DEFINITION 2. *A point process $\{N_t\}_{t>0}$ can be completely characterized by its conditional intensity function, defined as*

$$\lambda(t|\mathcal{H}_t) = \lim_{h \to 0} \frac{\mathbb{P}\{N_{t+h} - N_t = 1|\mathcal{H}_t\}}{h} \tag{1.9}$$

where $\mathcal{H}_t$ is the history of the process up to time t, containing the list of event times $\{T_1, T_2, ..., T_{N_t}\}$.

In the rest of this chapter, we use the shorthand notation $\lambda(t) =: \lambda(t|\mathcal{H}_t)$, always assuming an implicit history before time t. The above definition gives the intensity view of a point process, equivalent with the two previously defined views with events times and the counting process. In other words, the event intensity $\lambda(t)$ determines the distribution of event times, which also determine the counting process. Formally, $\lambda(t)$ and N_t are related through the probability of an event in a small time interval h:

$$\begin{aligned}
&\mathbb{P}(N_{t+h} = n+m \,|\, N_t = n) = \lambda(t)h + o(h) && \text{if} \quad m = 1 \\
&\mathbb{P}(N_{t+h} = n+m \,|\, N_t = n) = o(h) && \text{if} \quad m > 1 \\
&\mathbb{P}(N_{t+h} = n+m \,|\, N_t = n) = 1 - \lambda(t)h + o(h) && \text{if} \quad m = 0
\end{aligned} \tag{1.10}$$

where $o(h)$ is a function so that $\lim_{h\downarrow 0} \frac{o(h)}{h} = 0$. In other words, the probability of observing an event during the infinitesimal interval of time t and $t+h$ when $h \downarrow 0$ is $\lambda(t)h$. The probability of observing more than one event during the same interval is negligible.

1.3 Hawkes processes

In the models described in the previous section, the events arrive independently, either at a constant rate (for the Poisson process) or governed by an intensity function (for the non-homogeneous Poisson). However, for some applications, it is known that the arrival of an event increases the likelihood of observing events in the near future. This is the case of earthquake aftershocks when modeling seismicity, or that of user interactions when modeling preferential attachment in social networks. In this section, we introduce a class of processes in which the event arrival rate explicitly depends on past events – i.e. *self-exciting processes* – and we further detail the most well-known self-exciting process, the Hawkes process.

1.3.1 Self-exciting processes

A self-exciting process is a point process in which the arrival of an event causes the conditional intensity function to increase. A well known self-exciting process was proposed by Hawkes [1971], and it is based on a counting process in which the intensity function depends explicitly an all previously occurred events. The Hawkes process is defined as follows:

DEFINITION 3. *(Hawkes process) Let $\{N_t\}_{t>0}$ be a counting process with associated history $\mathcal{H}_t, t \geq 0$. The point process is defined by the event intensity function $\lambda(t)$ with respects Eq.* (1.10) *(the intensity view of a non-homogeneous Poisson process). The point process is said to be a Hawkes process if the conditional intensity function $\lambda(t|\mathcal{H}_t)$ takes the form:*

$$\lambda(t|\mathcal{H}_t) = \lambda_0(t) + \sum_{i:t>T_i} \phi(t - T_i) \ , \tag{1.11}$$

where $T_i < t$ are all the event time having occurred before current time t, and which contribute to the event intensity at time t. $\lambda_0(t) : \mathbb{R} \mapsto \mathbb{R}_+$ is a deterministic base intensity function, and $\phi : \mathbb{R} \mapsto \mathbb{R}_+$ is called the memory kernel – both of which are further detailed in the next section. We observe that the Hawkes process is a particular case of non-homogeneous Poisson process, in which the intensity is stochastic and explicitly depends on previous events through the kernel function $\phi(\cdot)$.

1.3.2 The intensity function

The quantity $\lambda_0(t) > 0$ is the base (or background) intensity, describing the arrival of events triggered by external sources. These events are also known as *exogenous* or *immigrant* events, and their arrival is independent on the previous events within the process. The self-exciting flavor of the Hawkes process arises through the summation term in Eq. (1.11), where the kernel $\phi(t - T_i)$ modulates the change that an event at time T_i has on the intensity function at time t. Typically, the function $\phi(\cdot)$ is taken to be monotonically decreasing so that more recent events have higher influence on the current event intensity, compared to events having occurred further away in time. Fig. 1.2(a) shows an example realization of a Hawkes process: nine events are observed, at times $T_1, T_2, \ldots, T_9$, and their corresponding inter-arrival times $\tau_1, \tau_2, \cdots, \tau_9$.

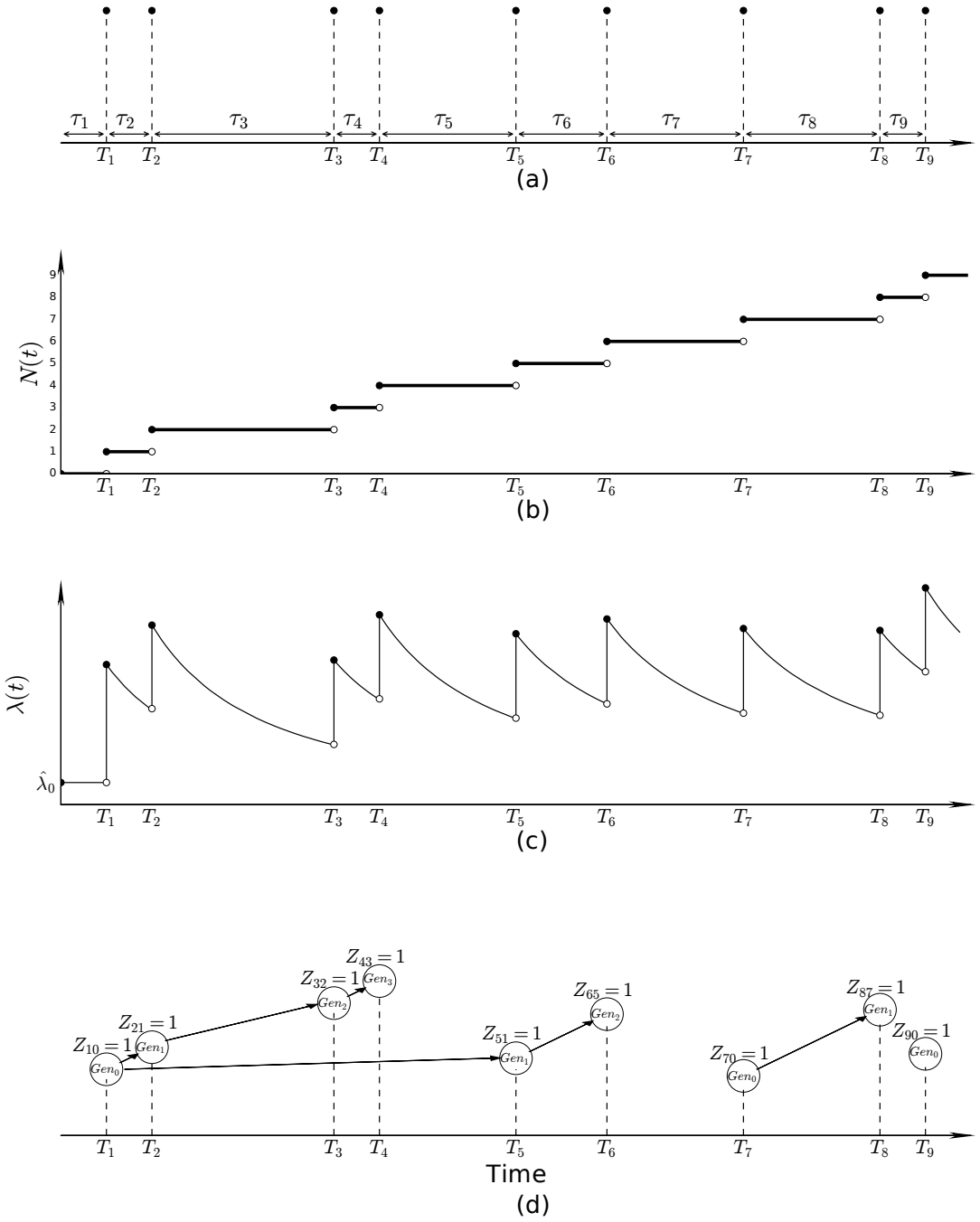

Figure 1.2 Hawkes process with an exponential decay kernel. (a) The first nine event times are shown. T_i represent event times, while τ_i represent inter-arrival times. (b) Counting process over time, N_t increases by one unit at each event time T_i. (c) Intensity function over time. Note how each event provokes a jump, followed by an exponential decay. Later decays unfold on top of the tail of earlier decays, resulting in apparently different decay rates. (d) The latent or unobserved branching structure of the Hawkes process. Every circle represents one event having occurred at T_i, the arrows represent the root-offspring relation. $\mathcal{G}en_i$ specifies the generation of the event, with $i = 0$ for immigrants or $i > 0$ for the offspring. Z_{ij} are random variables, such that $Z_{i0} = 1$ if event i is an immigrant, and $Z_{ij} = 1$ if event i is an offspring of event j.

Fig 1.2(b) shows the corresponding counting process N_t over time, which increases by one unit for each T_i as defined in Eq. (1.5). Fig. 1.2(c) shows the intensity function $\lambda(t)$ over time. Visibly, the value of the intensity function increases suddenly immediately at the occurrence of an event T_i, and diminishes as time passes and the effect of the given event T_i decays.

Choice of the kernel ϕ. The kernel function $\phi(\cdot)$ does not have to be monotonically decreasing. However, in this chapter we restrict the discussion to the decreasing families of functions, given that it is natural to see the influence of an event decay over time, as shown in Sec. 1.6. A popular decay function is the exponential function [Hawkes 1971], taking the following form:

$$\phi(x) = \alpha e^{-\delta x}, \tag{1.12}$$

where $\alpha \geq 0$, $\delta > 0$ and $\alpha < \delta$. Another kernel that is widely used in the literature is the power-law kernel:

$$\phi(x) = \frac{\alpha}{(x+\delta)^{\eta+1}}, \tag{1.13}$$

where $\alpha \geq 0$, $\delta, \eta > 0$ and $\alpha < \eta\delta^{\eta}$. This kernel is commonly used within the seismology literature [Ozaki 1979] and in the social media literature [Rizoiu et al. 2017]. The exponential kernel defined by Eq. 1.12 is typically the popular choice of kernel with Hawkes processes [Embrechts et al. 2011], unless demanded otherwise by the phenomena modeled using the self-exciting process (for example, we use a power-law kernel for modeling information diffusion in Social Media, in Sec. 1.6).

Other self-exciting point processes have been proposed, which follow the canonical specification given in Eq. (1.11) and which extend the initial self-exciting process proposed by Hawkes [1971]. We do not cover these processes in this chapter, however we advise the reader of Hawkes extensions such as the non-linear Hawkes processes [Brémaud and Massoulié 1996, Daley and Vere-Jones 2003], the general space time self-exciting point process [Ogata 1988, Veen and Schoenberg 2008], processes with exponential base event intensity [Dassios and Zhao 2011], or self-inhibiting processes [Yang et al. 2015].

1.3.3 The branching structure

Another equivalent view of the Hawkes process refers to the Poisson cluster process interpretation [Hawkes and Oakes 1974], which separates the events in a Hawkes process into two categories: *immigrants* and *offspring*. The offspring events are tiggered by existing (previous) events in the process, while the immigrants arrive independently and thus do not have an existing parent event. The offspring are said to be structured into *clusters*, associated with each immigrant event. This is called *the branching structure*. In the rest of this section, we further details the branching structure and we compute two quantities: the *branching factor* – the expected number of events directly triggered by a given event in a Hawkes process – and the estimated total number of events in a cluster of offspring. As shown in Sec. 1.6, both of these quantities become very important when the Hawkes processes are applied to practical domains, such as online social media.

An example branching structure. We consider the case that immigrant events follow a homogeneous Poisson process with base intensity $\lambda_0(t)$, while offspring are generated through the self-excitement, governed by the summation term in Eq. (1.32). Fig. 1.2(d) illustrates the branching structure of the nine event times of the example Hawkes process discussed earlier. Event times T_i are denoted by circles and the 'parent-

offspring' relations between the events are shown by arrows. We introduce the random variables Z_{ij}, where $Z_{i0} = 1$ if event i is an immigrant, and $Z_{ij} = 1$ if event i is an offspring of event j. The text in each circle denotes the generation to which the event belongs to, i.e. $\mathcal{G}en_k$ denotes the k-th generation. Immigrants are labeled as $\mathcal{G}en_0$, while generations $\mathcal{G}en_k, k > 0$ denote their offspring. For example T_3 and T_6 are immediate offspring of the immigrant T_2, i.e. mathematically expressible as $Z_{32} = 1$, $Z_{62} = 1$ and $Z_{20} = 1$.

The cluster representation states that the immediate offspring events associated with a particular parent arrive according to a non-homogeneous Poisson process with intensity $\phi(\cdot)$, i.e. T_3 and T_6 are realizations from non-homogeneous Poisson process with intensity $\phi(t - T_2)$ for $t > T_2$. The event that produces an offspring is described as the immediate ancestor or root of the offspring, T_7 is the immediate ancestor of T_8. The events which are directly or indirectly connected to an immigrant form the *cluster* of offspring associated with that immigrant, e.g. T_1 is an immigrant and T_2, T_3, T_4, T_5 and T_6 form its cluster of offspring. Similarly, T_7 and T_8 form another cluster. Finally, T_9 is a cluster by itself.

Branching factor (branching ratio). One key quantity that describes the Hawkes processes is its branching factor n^*, defined as the expected number direct offspring spawned by a single event. The branching factor n^* intuitively describes the amount of events to appear in the process, or informally, *virality* in the social media context. In addition, the branching factor gives an indication about whether the cluster of offspring associated with an immigrant is an infinite set. For $n^* < 1$, the process in a *subcritical regime*: the total number of events in any cluster is bounded. Immigrant event occur according to the base intensity $\lambda_0(t)$, but each one of them has associated a finite cluster of offspring, both in number and time extent. When $n^* > 1$, the process is in a so-called *supercritical regime* with $\lambda(t)$ increasing and the total number of events in each cluster being unbounded. We compute the branching factor by integrating $\phi(t)$ – the contribution of each event – over event time t:

$$n^* = \int_0^\infty \phi(\tau) d\tau \ . \tag{1.14}$$

Expected number of events in a cluster of offspring. The branching factor n^* indicates whether the number of offspring associated with each immigrant is finite ($n^* < 1$) or infinite ($n^* > 1$). When $n^* < 1$ a more accurate estimate of the size of each cluster can be obtained. Let A_i be the expected number of events in $Generation_i$, and $A_0 = 1$ (as each cluster has only one immigrant). The expected number of total events in the cluster, N_∞, is defined as:

$$N_\infty = \sum_{i=0}^{\infty} A_i \ . \tag{1.15}$$

To compute $A_i, i \geq 1$, we notice that each of the A_{i-1} events in the previous generation has on average n^* children events. This leads to a inductive relationship $A_i = A_{i-1} n^*$. Knowing that $A_0 = 1$, we derive:

$$A_i = A_{i-1}\, n^* = A_{i-2}\, (n^*)^2 = \ldots = A_0\, (n^*)^i = (n^*)^i, i \geq 1 \tag{1.16}$$

We obtain an estimate of the size of each cluster of immigrants N_∞ as the sum of a converging geometric progression (assuming $n^* < 1$):

$$N_\infty = \sum_{i=0}^{\infty} A_i = \frac{1}{1 - n^*} \text{ where } n^* < 1 \tag{1.17}$$

1.4 Simulating events from Hawkes processes

In this section, we focus on the problem of simulating series of random events according to the specifications of a given Hawkes process. This is a useful for gathering statistics about the process, and can form the basis for diagnostics, inference or parameter estimation. We present two simulation techniques for Hawkes processes. The first technique, the thinning algorithm [Ogata 1981], applies to all non-homogeneous Poisson processes, and can be applied to Hawkes processes with any kernel function $\phi(\cdot)$. The second technique, recently proposed by Dassios and Zhao [2013], is computationally more efficient, as it designs a variable decomposition technique for Hawkes processes with exponential decaying kernels.

1.4.1 The thinning algorithm

The basic goal of an sampling algorithm is to simulate inter-arrival times τ_i, $i = 1, 2, \ldots$ according to an intensity function λ_t. We first review the sampling method for a homogeneous Poisson process, then we introduce the thinning (or additive) property of Poisson processes, and we use this to derive the sampling algorithm for Hawkes processes.

Inter-arrival times in a homogeneous Poisson process follow an exponential distribution as specified in 1.3: $f_\tau(t) = \lambda e^{-\lambda t},\ t > 0$ and its cumulative distribution function is $F_\tau(t) = 1 - e^{-\lambda t}$. Because both $F_\tau(t)$ and $F_\tau^{-1}(t)$ have a closed-form expression, we can use the *inverse transform sampling* technique to sample waiting times. Intuitively, if X is a random variable with the cumulative distribution function F_X and $Y = F_X(X)$ is a uniformly distributed random variable ($\sim U(0,1)$), then $X^* = F_X^{-1}(Y)$ has the same distribution as X. In other words, sampling $X^* = F_X^{-1}(Y), Y \sim U(0,1)$ is identical with sampling X. For the exponentially distributed waiting times of the Poisson process, the inverse cumulative distribution function has the form $F_\tau^{-1}(u) = \frac{-\ln u}{\lambda}$. Consequently, sampling a waiting interval τ in a Poisson process is simply:

$$\text{Sample } u \sim U(0,1), \text{ then compute } \tau = \frac{-\ln u}{\lambda} \tag{1.18}$$

The thinning property of the Poisson processes states that a Poisson process with the intensity λ can be split into two independent processes with intensities λ_1 and λ_2, so that $\lambda = \lambda_1 + \lambda_2$. In other words, each event of the original process can be assigned to one of the two new processes that are running independently. From this property, we can see that we can simulate of a non-homogeneous Poisson process with the intensity function $\lambda(t)$ by *thinning* a homogeneous Poisson process with the intensity $\lambda^* \geq \lambda(t), \forall t$.

A thinning algorithm to simulate Hawkes processes is presented in Algorithm 1. For any bounded $\lambda(t)$ we can find a constant λ^* so that $\lambda(t) \leq \lambda^*$ in a given time interval. In particular, for Hawkes processes with a monotonically decreasing kernel function $\phi(t)$, it is easy to see that between two consecutive event times $[T_i, T_{i+1})$, $\lambda(T_i)$ is the upper bound of event intensity. We exemplify the sampling of event time T_{i+1}, after having already sampled $T_1, T_2, \ldots, T_i$. We start our time counter $T = T_i$. We sample an inter-arrival time τ, using Eq. (1.18), with $\lambda^* = \lambda(T)$ and we update the time counter $T = T + \tau$ (steps 3a to 3c in Algorithm 1) We accept or reject this inter-arrival time according to the ratio of the true event rate to the thinning rate λ^* (step 3e). If accepted, we record the event time $i+1$ as $T_{i+1} = T$. Otherwise, we repeat the sampling of an inter arrival time until one is accepted. Note that, even if an inter-arrival time is rejected, the time counter T is still updated, i.e. the principle of thinning a homogeneous Poisson process with a higher intensity value.

Algorithm 1 Simulation by thinning.

1. Given Hawkes process as in Eq (1.11)
2. Set current time $T = 0$ and event counter $i = 1$
3. While $i \leq N$
   (a) Set the upper bound of Poisson intensity $\lambda^* = \lambda(T)$ (using Eq (1.11)).
   (b) Sample inter-arrival time: draw $u \sim U(0,1)$ and let $\tau = -\frac{ln(u)}{\lambda^*}$ (as described in Eq (1.18)).
   (c) Update current time: $T = T + \tau$.
   (d) Draw $s \sim U(0,1)$.
   (e) If $s \leq \frac{\lambda(T)}{\lambda^*}$, accept the current sample: let $T_i = T$ and $i = i + 1$.
   Otherwise reject the sample, return to step (a).

Also note that, for efficiency reasons, the upper bound λ^* can be updated even in the case of a rejected inter-arrival time, given the strict monotonicity of $\lambda(t)$ in between event times. The temporal complexity of sampling N events is $O(N^2)$, since brute-force computation of event intensity using Eq (1.11) is $O(N)$. Furthermore, if event rates decay fast, then the number of rejected samples can be high before there is an accepted new event time.

1.4.2 Efficient sampling by decomposition

We now outline a more efficient sampling algorithm for Hawkes processes with an exponential kernel that does not resort to rejection sampling. Recently proposed by Dassios and Zhao [2013], it scales linearly to the number of events drawn.

First, the proposed algorithm applies to a Hawkes process with exponential immigrant rates and exponential memory kernel. This is a more general form than what we defined in Sec 1.3.2. The immigrant rate is described by a non-homogenous Poisson process following a exponential function $a + (\lambda_0 - a)e^{-\delta t}$. For each new event, the *jump* it introduces in event intensity is described by a constant γ.

$$\lambda(t) = a + (\lambda_0 - a)e^{-\delta t} + \sum_{T_i < t} \gamma e^{-\delta(t - T_i)},\ t > 0 \tag{1.19}$$

We can envision to generalize this even more by introducing a distribution to *gamma*, this is out of scope for this tutorial.

We note that a process is a Markov process, if it has the property that, conditional on the present, the future is independent of the past. Ogata [1981] showed that the intensity process is a Markov process when ϕ is exponential. This can be intuitively understood for event intensity function above, due to $\lambda(t_2) = e^{-\delta(t_2 - t_1)}\lambda(t_1)$, for any $t_2 > t_1$. In other words, given current event intensity $\lambda(t_1)$, future intensity only depend on the time elapsed since time t_1.

We use this Markov property to decompose the inter-arrival times into two independent simpler random variables. The first random variable s_0, represents the inter-arrival time of the next event, if it were to come from the constant background rate a. It is easy to see that this is sampled according to Eq (1.18). The second random variable s_1, represents the inter-arrival time of the next event if it were to come from either

Algorithm 2 Simulation of Hawkes with Exponential Kernel

1. Set $T_0 = 0$, initial event rate $\lambda(T_0) = \lambda_0$.
2. For $i = 1, 2, ..., N$
   (a) Draw $u_0 \sim U(0,1)$ and set $s_0 = -\frac{1}{a}\ln u_0$.
   (b) Draw $u_1 \sim U(0,1)$. Set $d = 1 + \frac{\delta \ln u_1}{\lambda(T_{i-1}^+) - a}$.
   (c) If $d > 0$, set $s_1 = -\frac{1}{\delta}\ln d$, $\tau_i = \min\{s_0, s_1\}$.
   Otherwise $\tau_i = s_0$
   (d) Record the i^{th} jump time $T_i = T_{i-1} + \tau_i$.
   (e) Update event intensity at the left side of T_i with exponential decay:
   $\lambda(T_i^-) = (\lambda(T_{i-1}^+) - a)e^{-\delta\tau_i} + a$
   (f) Update event intensity at the right side of T_i with a jump from the i^{th} event:
   $\lambda(T_i^+) = \lambda(T_i^-) + \gamma$

the exponential immigrant kernel $(\lambda_0 - a)e^{-\delta t}$ or the Hawkes self-exciting kernels from each of the past events $\sum_{T_i<t} e^{-\delta(t-T_i)}$. The cumulative distribution function of s_1 can be explicitly inverted due to its Markov property, a full derivation can be found in [Dassios and Zhao 2013]. Intuitively the sampled inter-arrival time is the minimum of these two cases. It is also worth noting that the second arrival time may not be finite, this is expected, as the exponential kernel decays fast. In this case, the next event will be an immigrant from the constant rate. This algorithm is outlined in Algorithm 2.

This algorithm is efficient because the intensity function can be updated in constant time for each event with steps (2e) and (2f), and that this algorithm does not rely on rejection sampling. The decomposition method above cannot be easily used on the power law kernel, since the power law does not have the Markov property.

1.5 Estimation of Hawkes processes parameters

One challenge when modeling using self-exciting point processes is estimating parameters from observed data. In the case of the Hawkes process with exponential kernel, one would typically have to determine the function $\lambda_0(t)$ (the base intensity defined in Eq. 1.11), and the values of the parameters of the decaying kernel $\phi(t)$ (α and δ, see Eq. 1.19). One can achieve this is by maximizing the likelihood over the observed data. In Sec. 1.5.1 we derive the formula of the likelihood function for a Hawkes process and in Sec. 1.5.2 we discuss a few practical concerns of using maximum likelihood estimation.

1.5.1 Likelihood function for Hawkes process

Let $N(t)$ be a point process on $[0,T]$ for $T < \infty$ and let $\{T_1, T_2, ..., T_n\}$ denote a realization, i.e. the set of event times, of $N(t)$ over the period $[0,T]$. Then the data likelihood L as a function of parameter set θ is:

$$L(\theta) = \prod_{i=1}^{n} \lambda(T_i) e^{-\int_0^T \lambda(t)dt}. \tag{1.20}$$

We sketch the derivation of the likelihood formula, along the lines of [Daley and Vere-Jones 2003, Laub et al. 2015, Rasmussen 2013]. If we are currently at some time t, recall that the history $\mathcal{H}_t$ is the list of times of events $T_1, T_2, ..., T_n$ up to but not including time t. Borrowing the $*$ notation from [Daley and Vere-Jones 2003], we define $f^*(t) := f(t|\mathcal{H}_t)$ be the conditional probability density function of the time of the next event T_{n+1} given the history of previous event $T_1, T_2, ..., T_n$. Recall that $\mathbb{P}\{T_{n+1} \in (t, t+dt)\} = f_{T_{n+1}}(t)dt$. We have

$$f(T_1, T_2, ..., T_n) = \prod_{i=1}^{n} f(T_i|T_1, T_2, ..., T_{i-1}) = \prod_{i=1}^{n} f^*(T_i) \tag{1.21}$$

It turns out that the event intensity $\lambda(t)$ can be expressed in terms of the conditional density f^* and its corresponding cumulative distribution function F^* [Rasmussen 2011].

$$\lambda(t) = \frac{f^*(t)}{1 - F^*(t)} \quad . \tag{1.22}$$

The expression above is given without a formal proof, but it can be interpreted heuristically as follows. Consider an infinitesimal interval dt around t, $f^*(t)dt$ correspond to the probability that there is an even in dt, and $1 - F^*(t)$ correspond to the probability of no new events before time t. After manipulating the expression using Bayes rule [Rasmussen 2011], the ratio of the two can be shown to be equivalent to the expectation of an increment of the counting process $N_{t+dt} - N_t$, which by Eq (1.10) is essentially $\lambda(t)dt$.

We can write the conditional intensity function in terms of the cumulative distribution function F^*:

$$\lambda(t) = \frac{f^*(t)}{1 - F^*(t)} = \frac{\frac{\partial}{\partial t} F^*(t)}{1 - F^*(t)} = -\frac{\partial}{\partial t} \log(1 - F^*(t)). \tag{1.23}$$

Denote the last known event time before t as T_n, integrating both sides from (T_n, t), we get

$$\int_{T_n}^{t} \lambda(s)ds = -[\log(1 - F^*(t)) - \log(1 - F^\star(T_n))]. \tag{1.24}$$

Note that $F^\star(T_n) = 0$ since $T_{n+1} > T_n$ and so

$$\int_{T_n}^{t} \lambda(s)ds = -\log(1 - F^*(t)). \tag{1.25}$$

Rearranging gives the following expression

$$F^*(t) = 1 - \exp\left(-\int_{T_n}^{t} \lambda(s)ds\right) \tag{1.26}$$

Combining the relationship between $\lambda(t)$, $f^*(t)$, and $F^*(t)$ in Eq 1.22 gives

$$f^*(t) = \lambda(t)\,(1 - F^*(t)) = \lambda(t) \exp\left(-\int_{T_n}^{t} \lambda(s)ds\right). \tag{1.27}$$

Plugging in Eq (1.27) above into the likelihood function, and combining integration ranges, we get the likelihood expression.

$$L(\theta) = \prod_{i=1}^{n} f^*(T_i) = \prod_{i=1}^{n} \lambda(T_i) e^{-\int_{T_{i-1}}^{T_i} \lambda(u)du} = \prod_{i=1}^{n} \lambda(T_i) e^{-\int_{0}^{T_n} \lambda(u)du}. \tag{1.28}$$

1.5.2 Maximum likelihood estimation

Let θ be the set of parameters of the Hawkes process, its maximum likelihood estimate can be found by maximizing the likelihood function in Eq. 1.20 with respect to θ over the space of parameter Θ. More precisely, the maximum likelihood estimate $\hat{\theta}$ is defined to be $\hat{\theta} = \arg\max_{\theta \in \Theta} l(\theta)$. From a standpoint of computational and numerical complexity, we note that summing is less expensive than multiplication. But more importantly, likelihoods would become very small and would risk the running out of floating point precision very quickly, yielding an underflow, thus it is customary to maximize the log of the likelihood function:

$$l(\theta) = \log L(\theta) = -\int_0^T \lambda(t)dt + \sum_{i=1}^{N(T)} \log \lambda(T_i) \tag{1.29}$$

The natural logarithm is a monotonic function and maximizing the log-likelihood automatically implies maximizing the likelihood function. The negative log-likelihood can be minimized with optimization packages for non-linear objective, such as the L-BFGS [Zhu et al. 1997] software.

Local maxima. One may run into problems of multiple local maxima in the log-likelihood. The shape of the negative log-likelihood function can be fairly complex and may not be globally convex. Due to the possible non-convex nature of the log-likelihood, performing maximum likelihood estimation would result in the estimate being the local maximum rather than the global maximum. A usual approach used in trying to identify the global maximum involves using several sets of different initial values for the maximum likelihood estimation. Note that this does not mitigate the problem entirely and it is well possible that a local maximum may still be wrongly established as the global maximum. Alternatively, one can use different optimization methods in conjunction with several different sets of initial values. If the differing optimizations result in a consistent set of calibrated parameters, then we can have a higher certainty that the calibrated point is the actual global maximum.

Edge effects. Recall that N_t is the number of 'arrivals' or 'events' of the process by time t and that the sequence of event times $T_1, T_2, ..., T_{N_T}$ is assumed to be observed within the time interval $[0, T]$, where $T < \infty$. As discussed in Sec. 1.3.3, in a Hawkes process, the events usually arrive clustered in time: an immigrant and its offspring. In practical applications, the process might have started sometime in the past, prior to the moment when we start observing it, denoted as $t = 0$. Hence, there may be unobserved event times which occurred before time 0, which could have generated offspring events during the interval $[0, T]$. It is possible that the unobserved event times have an impact during the observation period, i.e. sometime after $t > 0$, but because we are not aware of them, their contribution to the event intensity is not recorded. Such phenomenon are referred to as *edge effects* and are discussed in [Daley and Vere-Jones 2003] and [Rasmussen 2013]. One possible avenue to address this issue is to assume that the initial value of the intensity process equals the base intensity and disregard edge effects from event times occurring before the observation period, see Daley and Vere-Jones [2003]. This is usually the modeling setup in most applications within the Hawkes literature. As pointed out by [Rasmussen 2013], the edge effects on the estimated model would turn out to be negligible if the used dataset is large enough. In this chapter, we set the base intensity to be a constant $\lambda(0) = \lambda_0$ and ignore edge effects from events that have occurred before the start of the observation period. For detailed discussions on handling edge effects, we refer the reader to the extensive works of [Baddeley and Turner

2000, Bebbington and Harte 2001, Daley and Vere-Jones 2003, Møller and Rasmussen 2005, Rasmussen 2013] which are summarized in [Lapham 2014].

Computational bottleneck. A major issue with maximum likelihood estimation for Hawkes is the computational costs for evaluating the log-likelihood, in particular the evaluation of the intensity function, as shown here-after. Note that the two components of the log-likelihood in Eq. (1.29) can be maximized separately since if they do not have common terms, see [Daley and Vere-Jones 2003, Ogata 1988, Zipkin et al. 2016]. The computational complexity arises due to the calculation of a double summation operation. This double sum comes from the second part of the log-likelihood:

$$\sum_{i=1}^{N_T} \log \lambda(T_i) = \sum_{i=1}^{N_T} \left(\log(a + (\lambda_0 - a)e^{-\delta t} + \sum_{j:T_j<T_i} \alpha e^{-\delta(T_i - T_j)}) \right). \tag{1.30}$$

Note the complexity for most Hawkes process is usually of the order $O(N_T^2)$, where N_T is the number of event times. Hence estimating the parameters can be relatively slow when N_T is of a big number, and it may be exacerbated if loop calculations cannot be avoided. In the case of an exponential kernel function, the number of operations required to evaluate Eq. (1.30) can be reduced to $O(N_T)$ using a recursive formula [Ogata 1981]. For a more complicated Hawkes process involving a power-law decay kernel, such as the Epidemic Type Aftershock-Sequences (ETAS) model [Ogata 1988] or the social media kernel constructed in Sec. 1.6, this strategy does not hold. For the ETAS model, the event intensity is defined as:

$$\lambda(t) = \lambda_0 + \sum_{i:t>T_i} \alpha \frac{e^{\delta \eta_1}}{(t - T_i + \gamma)^{\eta_2 + 1}} \tag{1.31}$$

for some constants $\lambda_0, \alpha, \eta_1, \gamma, \eta_2$. The ETAS model is a point process used typically to represent the temporal activity of earthquakes for a certain geophysical region. To reduce the computational complexity for the ETAS model, [Ogata et al. 1993] presented a methodology which involved multiple transformations and numerical integration. They showed that there is a signification reduction in the time taken to learn the parameters and further demonstrated that they are close approximation of the maximum likelihood estimates.

1.6 Constructing a Hawkes model for Social Media

The previous sections of this chapter introduced the theoretical bases for working with Hawkes processes. Sec. 1.2 and 1.3 gave the definitions and the basic properties of point processes, Poisson processes and Hawkes processes. Sec. 1.4 and 1.5 respectively presented methods for simulating events in a Hawkes process and fitting the parameters of a Hawkes process to data. The aim of this section is to provide a guided tour for using Hawkes processes with social media data. We will start from customizing the memory kernel with a goal of predicting the popularity of an item. The core techniques here is from a recent paper [Mishra et al. 2016] on predicting the size of a retweet cascade. In Sec. 1.6.1 we argue why a Hawkes process is suitable for modeling the retweet cascades and we present the construction of the kernel function $\phi(t)$; in Sec. 1.6.2 we estimate model parameters from real-life data using Twitter data; in Sec. 1.6.3 we predict the expected size of a retweet cascade, i.e. its popularity.

1.6.1 A marked Hawkes process for information diffusion

We model *word of mouth* diffusion of online information: users share content, and other users consume and sometimes re-share it, broadcasting to more users. For this application, we consider each retweet as an event in the point process. We also formulate information diffusion in Twitter as a self-exciting point process, in which we model three key intuitions of the social network: *magnitude of influence*, tweets by users with many followers tend to get retweeted more; *memory over time*, that most retweeting happens when the content is *fresh* [Wu and Huberman 2007]; and *content quality*.

The event intensity function. A retweet is defined as the resharing of another person's tweet via the dedicated functionality on the Twitter interface. A retweet cascade is defined as the set of retweets of an initial tweet. Using the branching structure terminology introduced in Sec. 1.3, a retweet cascade is made of an *immigrant* event and all of its *offsprings*. We recall the definition of the event intensity function in a Hawkes process, introduced in Eq. (1.11):

$$\lambda(t) = \lambda_0(t) + \sum_{T_i<t} \phi_{m_i}(t - T_i) \quad . \tag{1.32}$$

$\lambda_0(t)$ is the arrival rate of immigrants events into the system. The original tweet is the only immigrant event in a cascade, therefore $\lambda_0(t) = 0, \forall t > 0$. Furthermore, this is modeled as a *marked* Hawkes process. The *mark* or magnitude of each event models the user influence for each tweet. The initial tweet has event time $T_0 = 0$ and mark m_0. Each subsequent tweet has the mark m_i at event time T_i.

We construct a power-law kernel $\phi_m(\tau)$ with mark m:

$$\phi_m(\tau) = \kappa m^{\beta} (\tau + c)^{-(1+\theta)} \quad . \tag{1.33}$$

κ describes the *virality* – or quality – of the tweet content and it scales the subsequent retweet rate; β introduces a warping effect for user influences in social networks; and $1 + \theta$ ($\theta > 0$) is the power-law exponent, describing how fast an event is *forgotten*, parameter $c > 0$ is a temporal shift term to keep $\phi_m(\tau)$ bounded when $\tau \simeq 0$. Overall, κm^{β} accounts for the magnitude of influence, and the power-law kernel $(\tau + c)^{-(1+\theta)}$ models the memory over time. We assume user influence m is observed the number of followers obtained from Twitter API.

In a similar fashion, we can construct an exponential kernel for social media, based on the kernel defined in Eq. (1.12):

$$\phi_m(\tau) = \kappa m^{\beta} \theta e^{-\theta\tau} \quad . \tag{1.34}$$

We have experimented with this kernel and Fig. 1.3(c) shows the its corresponding intensity function over time for a real twitter diffusion cascade. However, we have found that the exponential kernel for social media provides lower prediction performances compared to the power-law kernel defined in Eq. 1.33. Consequently, in the rest of this chapter, we only present the power-law kernel.

1.6.2 Estimating the Hawkes process

The marked Hawkes process has four parameters $\theta = \{\kappa, \beta, c, \theta\}$, which we set out to estimate using maximum likelihood estimation technique described in Sec. 1.5. We can obtain the its log-likelihood by

introducing the marked memory kernel (1.33) into the general log-likelihood formula shown in Eq. (1.29). The first two terms in Eq. 1.35 are from the likelihood computed using the event rate $\lambda(t)$, the last term is a normalization factor from integrating the event rate over the observation window $[0,T]$.

$$\mathcal{L}(\kappa,\beta,c,\theta) = \sum_{i=2}^{n} \log \kappa + \sum_{i=2}^{n} \log \left(\sum_{t_j<t_i} \frac{(m_j)^{\beta}}{(t_i - t_j + c)^{1+\theta}} \right) - \kappa \sum_{i=1}^{n} (m_i)^{\beta} \left[\frac{1}{\theta c^{\theta}} - \frac{(T+c-t_i)^{-\theta}}{\theta} \right] . \tag{1.35}$$

Eq. 1.35 is a non-linear objective that need to be maximized. There are a few natural constraints for each of model parameter, namely: $\theta > 0$, $\kappa > 0$, $c > 0$, and $0 < \beta < \alpha - 1$ for the branching factor to be meaningful (and positive). Furthermore, while the supercritical regimes $n^* > 1$ are mathematically valid, it will lead to a prediction of infinite cascade size – a clearly unrealistic outcome. We further incorporate $n^* < 1$ as a non-linear constraint for the maximum likelihood estimation. `Ipopt` [Wächter and Biegler 2006], the large-scale interior point solver can be used to handles both non-linear objectives and non-linear constraints. For efficiency and precision, it needs to be supplied with pre-programmed gradient functions. Details of the gradient computation and optimization can be found in the online supplement [Mishra et al. 2016].

Sec. 1.5.2 warned about three possible problems that can arise when using maximum likelihood estimates with Hawkes processes: edge effects, squared computational complexity and local minima. In this application, since we always observe a cluster of events generated by an immigrant, we do not having *edge effects*, i.e, missing events early in time. The *computational complexity* of calculating the log-likelihood and its gradients is $O(n^2)$, or quadratic with respective to the number of observed events. In practice, we use three techniques to make computation more efficient: vectorization in the R programming language, storing and reusing parts of the calculation, and data-parallel execution across a large number of cascades. With these techniques, we estimated tens of thousands of moderately-sized retweet cascades containing hundreds of events in reasonable amount of time. Lastly, the problem of *local minima* can be addressed using multiple random initializations, as discussed in Sec. 1.5.2.

1.6.3 The expected number of future events

Having observed a retweet cascade until time T for a given Hawkes process, one can simulate a possible continuation of the cascade using the thinning technique presented in Sec. 1.4. Assuming a subcritical regime, i.e. $n^* < 1$, the cascade is expected to die out in all possible continuation scenarios. In addition to simulation a handful of possible endings, it turns out there is a close-form solution to the expected number of future events in the cascade over all possible continuations, i.e., the total popularity that the cascade will reach at the time of its ending.

There are three key ideas for computing the expected number of future events. The first is to compute the expected size of a direct offsprings to a event at T_i after time T; the second is that the expected number all descendent events can be obtained via the branching factor of the Hawkes process, as explained in Sec. 1.3.3. Lastly, the estimate of total popularity emerges when we put these two ideas together.

The number of future children events. In retweet cascades, the base intensity is null $\lambda_0(t) = 0$, therefore no new immigrants will occur at $t > T$. Eq. (1.17) gives the expected size of a cluster of offprings associated with an immigrant. In the marked Hawkes process Eq (1.32), each of the $i = 1, \ldots, n$ events that happened at $T_i < T$ adds $\phi_{m_i}(t - T_i)$ to the overall event intensity. We can obtain the expectation of A_1, the total number of events directly triggered by event $i = 1, \ldots, n$, by integrating over the memory kernels of each event. The summation and integration are exchangeable here, since the effect of each event on future event intensity is additive.

$$\begin{aligned} A_1 &= \int_T^\infty \lambda(t)\mathrm{d}t = \int_T^\infty \sum_{t>T_i} \phi_{m_i}(t - T_i)\mathrm{d}t \\ &= \sum_{t>T_i} \int_T^\infty \phi_{m_i}(t - T_i)\mathrm{d}t = \kappa \sum_{i=1}^{n} \frac{{m_i}^\beta}{\theta\,(T + c - T_i)^\theta} \end{aligned} \tag{1.36}$$

The branching factor The branching factor was defined in Eq. (1.14) for an unmarked Hawkes process. We compute the branching factor of the marked Hawkes process constructed in Sec 1.6.1 by taking expectations over both event times and event marks. We assume that the event marks m_i are *i.i.d.* samples from a power law distribution of social influence [Kwak et al. 2010]: $P(m) = (\alpha - 1)\, m^{-\alpha}$. α is an exponent which controls the heavy tail of the distribution and it is estimated from a large sample of tweets. We obtain the closed-form expression of the branching factor (see Mishra et al. [2016] for details):

$$n^* = \kappa \frac{\alpha - 1}{\alpha - \beta - 1} \frac{1}{\theta c^\theta}, \text{ for } \beta < \alpha - 1 \text{ and } \theta > 0 \ . \tag{1.37}$$

Total size of cascade. Putting both Eq (1.36) and Eq (1.37) together, we can see that each expected event in A_1 is expected to generate n^* direct children events, n^{*2} grand-children events, ..., n^{*k} k-th generation children events, and so on. The calculation of geometric series shows that the number of all descendants is $\frac{A_1}{1-n^*}$. This quantity plus the observed number of events n is the total number of expected events in the cascade. See [Mishra et al. 2016] for complete calculations.

$$N_\infty = n + \frac{\kappa}{(1 - n^*)} \left(\sum_{i=1}^{n} \frac{{m_i}^\beta}{\theta\,(T + c - t_i)^\theta} \right), n^* < 1 \tag{1.38}$$

1.6.4 Interpreting the generative model

A Hawkes process is a generative model, meaning that it can be used to interpret statistical patterns in diffusion processes, in addition to being used in predictive tasks. Fig. 1.3 presents a diffusion cascade about a New York Times news article with its corresponding intensity functions with the power-law and exponential memory kernels, respectively. Note that the top and lower two graphics are temporally aligned. In other words, each occurred event causes a jump in the intensity function, i.e. increasing the likelihood of future events. Each jump is followed by a rapid decay, governed by the decay kernel $\phi_m(\tau)$, defined in Sec 1.6.1. In terms of event marks, the cascade attracts the attention of some very well-followed accounts. The original poster (`@screencrushnews`) has 12,122 followers, and among the users who retweeted, `@TasteOfCountry` (country music) has 193,081 followers, `@Loudwire` (rock) had 110,824 followers,

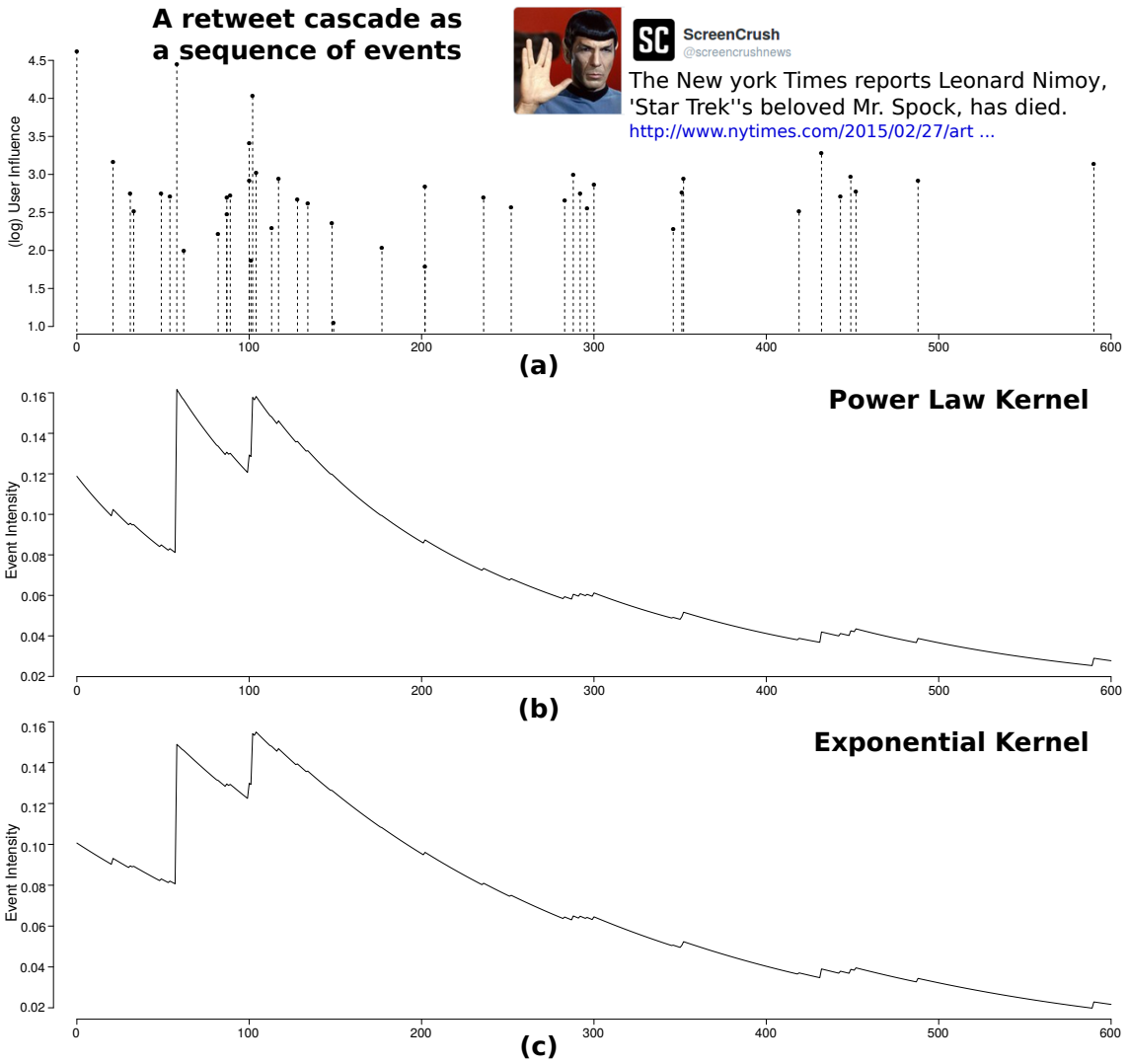

Figure 1.3 An example retweet cascade on a news article by The New York Times. (a) Representation of the first 600 seconds of the retweet cascade as a marked point process, to each (re)tweet corresponds an event time. (b) Event intensity ($\lambda(t)$) over time, assuming the point process to be a Hawkes process with power-law kernel. The maximum-likelihood model parameter estimates are $\{\kappa = 1.00, \beta = 1.01, c = 250.65, \theta = 1.33\}$ with a corresponding $n^* = 0.92$ and a predicted cascade size of 216. The true cascade size is 219. (c) The event intensity over time for the same event time series, when the point process is assumed to the a Hawkes process with the exponential kernel define din Eq. 1.34. The fitted parameters for this kernel are $\{\kappa = 0.0003, \beta = 1.0156, \theta = 0.0054\}$, the corresponding $n^* = 0.997$ and the predicted cascade size is 1603.

`@UltClassicRock` (classic rock) has 99,074 followers and `@PopCrush` (pop music) has 114,050 followers.

For popularity prediction, the cascade is observed for 10 minutes (600 seconds) and the parameters of the Hawkes process are fitted as shown in Sec. 1.6.2. The maximum-likelihood estimate of parameters with a power-law kernel are $\{\kappa = 1.00, \beta = 1.01, c = 250.65, \theta = 1.33\}$, with a corresponding $n^* = 0.92$. According to the power-law kernel, this news article has high content virality (denoted by κ) and large waiting time (c), which in turn lead to a slow diffusion: the resulting cascade reached 1/4 its size after half an hour, and the final tweet was sent after 4 days. By contrast, most retweet cascades finish in a matter of minutes, tens of minutes at most. Using the formula in Eq. (1.38), we predict the expected total cascade size $N_\infty = 216$, this is very close to the real cascade size of 219 tweets, after observing only the initial 10 minutes of the 4 day Twitter diffusion. When estimated with an exponential kernel, the parameters of Hawkes point process are $\{\kappa = 0.0003, \beta = 1.0156, \theta = 0.0054\}$ and the corresponding branching factor is $n^* = 0.997$. This produces an very imprecise total cascade size prediction of 1603 tweets, largely due to the high n^*.

1.6.5 Hands-on tutorial

In this section, we provide a short hand-on tutorial, together with code snippets required for modeling information diffusion through retweet cascades. A detailed version of tutorial with example data and code is available at https://github.com/s-mishra/featuredriven-hawkes. All code examples presented in this section assume a Hawkes model with the power-law kernel. The complete online tutorial also presents examples which use an exponential kernel. All code was developed using the R programming language.

We start with visualizing in Fig. 1.4 the shape of the power-law kernel (defined in Eq. (1.33)) generated by an event with the mark $m = 1000$, and defined by the parameters $\kappa = 0.8$, $\beta = 0.6$, $c = 10$ and $\theta = 0.8$. The code for generating the figure is shown in Listing. 1.1. Furthermore, we can simulate (Listing 1.2) the entire cluster of offspring generated by this initial immigrant event using the thinning procedure described in Sec. 1.4.1. The initial event is assumed to have occurred at time $t = 0$, and the simulation is ran for 50 time intervals.

We now show to estimate the parameters of a Hawkes process with a power-law kernel for a real Twitter diffusion cascade and how to estimate the total size of the cascade. The file `example_book.csv` in the online tutorial records the retweet diffusion cascade around a news article announcing the death of "Mr. Spock" shown in Fig. 1.3. Fig. 1.3(a) depicts the cascade as a point process: the tweet posting times are the event times, whereas the number of followers of the user emitting the tweets are considered the event marks. The code in Listing 1.3 reads the CSV file and performs a maximum likelihood estimation of the Hawkes process parameters, based on the events in the cascade having occurred in the first 600 seconds (10 minutes). With the obtained estimates for model parameters, we can predict (using the code in Listing 1.4) the total size of the diffusion cascade.

1.7 Summary

This chapter provided a gentle introduction for Hawkes self-exciting process. We covered the key definitions of point processes and Hawkes processes. We introduced the notion of event rate, branching factor, and the use of these quantities to predict future events. We described procedures for simulating a Hawkes process,

and derived the likelihood function used for parameter estimation. We also included a practical example for estimating a Hawkes process from retweet cascades, along with code snippets and online notebook. Where applicable, we have included discussions of the point-process literature. The goal of the materials above is to provide the fundamentals to researchers who are interested in formulating and solving application problems with point processes. Interested readers are invited to explore more advanced materials, including: alternative inference algorithms such as using expectation-maximization, sampling, or moment matching; flexible specifications and extensions of self-exciting processes such as multi-variate mutually-exciting Hawkes processes, doubly-stochastic processes, to name a few.

Listing 1.1 Code for computing the power-law kernel function, generated by an event with mark 1000.

```
## initial event that starts the cascade (i.e. the immigrant)
## mark = 1000 at time t = 0
event <- c(mark = 1000, time = 0)
## the timepoints for which we compute the kernel function
t <- seq(from = 0, to = 100, length=1000)
## set the parameters of the kernel
K <- 0.8
beta <- 0.6
c <- 10
theta <- 0.8

## compute the Power Law Kernel
## call the kernelFunction to get the values
values.PL <- kernelFct(event = event, t = t, K = K, beta = beta, c = c,
                       theta = theta, kernel.type='PL')

## plot the obtained kernel
plot(x = t, y = values.PL, type = "l", col = "blue",
     xlab = "", ylab = "", main = "Power-law memory kernel over time")
```

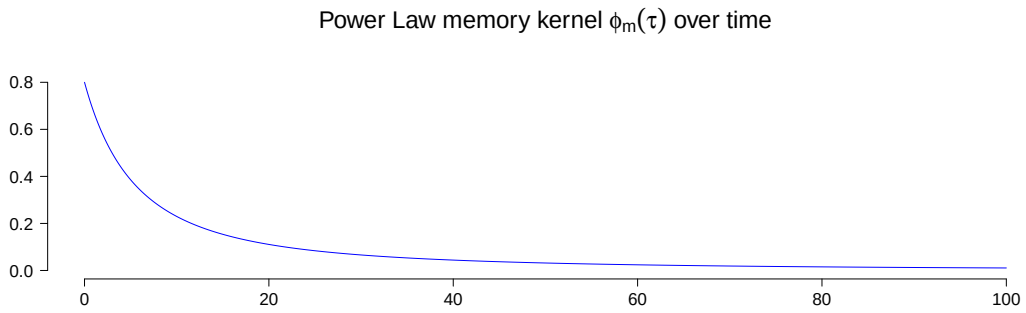

Figure 1.4 Graphic obtained by running the code in Listing 1.1: The power-law kernel over time, generated by an event with the mark 1000.

Listing 1.2 Simulation using the thinning method (see Sec. 1.4.1) of an entire cluster of offspring, generated by the immigrant defined in Listing 1.1 ($m = 1000, t = 0$).

```
## simulating an event series
events <- generate_Hawkes_event_series(K = K, beta = beta, c = c, theta = theta,
                                       M = event["mark"], Tmax = 50)
```

Listing 1.3 Load the information about the real tweet cascade from Fig. 1.3, and fit parameters using the events observed in the first 600 seconds.

```
## read the real cascade provided in the file "example.csv"
real_cascade <- read.csv(file = 'example_book.csv', header = T)
## retain only the events that occurred in the first 600 seconds (10min). These
## will be used for fitting the model parameters.
predTime <- 600
history <- real_cascade[real_cascade$time <= predTime, ]
## removing the first column, which  is event index
## retaining column 2 (event mark) and column 3 (event time).
history <- history[ , 2:3]
## call the fitting function, which uses IPOPT internally. The fitting algorithm
## requires an initial guess of the parameters. This can a random point within
## the domain of definition of parameters.
startParams <- c(K = 1, beta = 1, c = 250, theta = 1)
result <- fitParameters(startParams, history)
```

Listing 1.4 Predict the total size for the twitter cascade shown in Fig. 1.3, using the parameters fitted as in Listing 1.3

```
## Using the fitted model parameters, we call getTotalEvents to get predictions.
prediction <- getTotalEvents(history = history, bigT = predTime,
                             K = result$solution[1],
                             beta = result$solution[2],
                             c = result$solution[3],
                             theta = result$solution[4])
## The "predction" object contains other values, such as
## the branching factor (nstor) and A1
nPredicted = prediction['total']
```

Bibliography

A. Baddeley and R. Turner. 2000. Practical maximum pseudolikelihood for spatial point patterns. *Australian & New Zealand Journal of Statistics*, 42: 283–322.

M. Bebbington and D. S. Harte. 2001. On the statistics of the linked stress release model. *Journal of Applied Probability*, pp. 176–187.

P. Brémaud and L. Massoulié. 1996. Stability of nonlinear Hawkes processes. *The Annals of Probability*, 24(3): 1563–1588.

D. J. Daley and D. Vere-Jones. 2003. *An Introduction to the Theory of Point Processes*, 2nd. Springer-Verlag New York.

A. Dassios and H. Zhao. 2011. A dynamic contagion process. *Advances in Applied Probability*, pp. 814–846.

A. Dassios and H. Zhao. 2013. Exact simulation of Hawkes process with exponentially decaying intensity. *Electronic Communications in Probability*, 18: 1–13.

P. Embrechts, T. Liniger, and L. Lin. 08 2011. Multivariate hawkes processes: an application to financial data. *Journal of Applied Probability*, pp. 367–378.

A. G. Hawkes. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, pp. 89–90.

A. G. Hawkes and D. Oakes. 1974. A cluster process representation of a self-exciting process. *Journal of Applied Probability*, pp. 493–503.

H. Kwak, C. Lee, H. Park, and S. Moon. 2010. What is twitter, a social network or a news media? In *WWW '10*, pp. 591–600.

B. M. Lapham. 2014. Hawkes processes and some financial applications. Thesis, University of Cape Town.

P. J. Laub, T. Taimre, and P. K. Pollett. 2015. Hawkes processes. *arXiv preprint arXiv:1507.02822*.

S. Mishra, M.-A. Rizoiu, and L. Xie. 2016. Feature Driven and Point Process Approaches for Popularity Prediction. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, CIKM '16. Indianapolis, IN, USA. DOI: 10.1145/2983323.2983812.

J. Møller and J. G. Rasmussen. 2005. Perfect simulation of Hawkes processes. *Advances in Applied Probability*, 37(3): 629–646.

Y. Ogata. 1981. On Lewis' simulation method for point processes. *IEEE Transactions on Information Theory*, 27(1): 23–31.

Y. Ogata. 1988. Statistical Models for Earthquake Occurrences and Residual Analysis for Point Processes. *Journal of the American Statistical Association*, 83(401).

Y. Ogata, R. S. Matsuúra, and K. Katsura. 1993. Fast likelihood computation of epidemic type aftershock-sequence model. *Geophysical Research Letters*, 20(19): 2143–2146.

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

T. T. Pham, S. Hamid Rezatofighi, I. Reid, and T.-J. Chin. 2016. Efficient point process inference for large-scale object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2837–2845.

J. Rasmussen, 2011. Temporal point processes: the conditional intensity function. .

J. G. Rasmussen. 2013. Bayesian inference for Hawkes processes. *Methodology and Computing in Applied Probability*, 15(3): 623–642. ISSN 1387-5841.

M.-A. Rizoiu, L. Xie, S. Sanner, M. Cebrian, H. Yu, and P. Van Hentenryck. 2017. Expecting to be HIP: Hawkes Intensity Processes for Social Media Popularity. In *World Wide Web 2017, International Conference on*, pp. 1–9. Perth, Australia. http://arxiv.org/abs/1602.06033.

I. M. Toke. 2011. An introduction to hawkes processes with applications to finance.

A. Veen and F. P. Schoenberg. 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.

A. Wächter and L. T. Biegler. 2006. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1): 25–57.

F. Wu and B. A. Huberman. nov 2007. Novelty and collective attention. *PNAS '07*, 104(45): 17599–601. ISSN 0027-8424. http://www.pnas.org/content/104/45/17599.abstract. DOI: 10.1073/pnas.0704916104.

Q. Yang, M. J. Wooldridge, and H. Zha. 2015. Trailer Generation via a Point Process-Based Visual Attractiveness Model. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pp. 2198–2204. AAAI Press. ISBN 9781577357384.

C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. 1997. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4): 550–560.

J. R. Zipkin, F. P. Schoenberg, K. Coronges, and A. L. Bertozzi. 2016. Point-process models of social network interactions: Parameter estimation and missing data recovery. *European Journal of Applied Mathematics*, 27: 502–529.