POLITECNICO DI TORINO

Master degree in Engineering Mathematics

Master's Degree Thesis

Data-driven characterization and analysis of fringe social networks

Supervisors
prof. Fabrizio Dabbene
prof.ssa Chiara Ravazzi
prof. Francesco Malandrino

Candidate
Davide Grande

Academic Year 2022-2023

ai miei nonni

Abstract

In this thesis we present a data-driven characterization of fringe social networks. By the term fringe social networks we mean all those small emerging structures on the Web that are not mainstream, such as Twitter or Facebook. These networks generally promote themselves as a "free-speech" alternative to the mainstream, but often serve as an incubator of misleading information, hateful and malicious content, due to their lack of moderation. In particular, we will focus on the fringe social network Parler and report some statistical analysis on a dataset of 183 million Parler posts between August 2018 and January 2021. The main goal is to perform an analysis on the cascades of hashtags related to the first impeachment of U.S. President Donald Trump. Our analysis shows how malicious and hateful trends are pumped into the network by some bad actors or some other form of manipulation. We claim that the hashtag cascade can be modeled using the Hawkes process framework with the particular choice of exponential decay kernel. We prove the goodness of our hypothesis by performing parameter estimation and present some statistical tools to evaluate the goodness of fit. Finally, our analysis allows to unveil the correlation of level of hate and misleading information to the level of attention from these social communities.

Acknowledgements

Vorrei sinceramente ringraziare la Professoressa Chiara Ravzzi, il Professor Francesco Malandrino e il Professor Fabrizio Dabbene che mi hanno seguito durante questo percorso. Il vostro continuo supporto, la costante presenza e la vostra simpatia hanno reso più leggeri e sicuramente più piacevoli questi mesi di duro lavoro.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In recent years, Web communities have seen exponential growth in the number of users joining them, making the Web a meeting place for thousands of people to share experiences, cultures, and ideologies. By mainstream social networks we mean the most widespread and popular social media platforms, such as Facebook and Twitter. One of the immediate and biggest problems that the creators of these online social networks faced was certainly finding an appropriate policy to effect some moderation and control over the flow of information among users. Indeed, social network plazas have seen, since their inception, an increasing number of illicit activities, such as the sharing of obscene content, misinformation, hate comments, and organized content related to terrorism. To promote a healthier environment and discourage violence, harassment and any other kind of social violence, major social networks have adopted strict moderation policies. As an example, Twitter uses a moderation system based on a combination of machine learning algorithms and proactive moderation by users. Some of Twitter's general safety guidelines cover violent speech, child sexual exploitation, abuse/harassment, hateful conduct, doxxing (i.e., the act of publicly providing sensitive and private information about an individual), and deceptive identities. This oversight system is constantly updated and trained on millions of tagged content and has recently been updated to take into account thousands of behavioral factors to properly classify tweets [Larson, 2018]. Examples might be whether an account has confirmed its e-mail address or how often an account mentions users who do not follow it. If a tweet is identified as unsafe, off-topic, or a troll, it will appear lower in conversations and searches. Facebook (Meta), on the other hand, has recently adopted a slightly different policy, entrusting all decision-making power to a body called the Oversight Board, a group of 20 members chosen from 27 countries and speaking 29 languages who form a kind of "platform self-government" [Wong and Floridi, 2022] to ensure a free and safe space for all users. The punishment for violating privacy guidelines

is banning the account from the social network. This strict moderation and control over the flow of information is one of the main reasons why many users have migrated to fringe social networks, where they can find an unsupervised environment. As example, in January 2021 Donald Trump, banned by Twitter, on which he had 88 million followers, set up an account on Parler, the "Free Speech Social Network" founded in 2018 by John Matze that billed itself as a platform where any content can be posted. Then two million of people migrated to this social network: users included mainly Donald Trump supporters, dissatisfied with the last election round, and Qanon conspirators, but also politicians, Fox News hosts, and YouTube program hosts. With the term of fringe social networks we mean all those emerging structures on the Web that oppose the structure of mainstream social networks. The number of users that these platforms attract is constantly increasing, as they often find in these niches of the Web a place where the lack of moderation allows the unchecked circulation of news, comments and ideologies. This is why fringe social networks have recently been associated with the spread of disinformation, hatred and radicalization. In addition, they have been linked to some extremist political institutions that act as incubators of conspiracy theories that can harm some democratic institutions. These risks highlighted the urgency to understand how social influence can translate into real-life episodes of violence and how these marginal online communities can be monitored to prevent such episodes. The threat that these fringe social networks pose to the spread of misinformation and harmful content has been extensively analyzed in the literature. In [Zannettou, 2019] the influence that these web niches can have on traditional social networks is demonstrated by a detailed statistical analysis of the impact of fake news. Indeed, it has been shown that fake news initially finds its way into fringe networks and then manages to propagate into traditional social networks as well, reaching a huge number of users. A detailed characterization of fake news is then provided and then, using the Hawkes process framework, the influence of some fringe networks (Gab, Reddit, 4chan) on traditional social networks (Twitter) is presented. The dissemination of harmful content through fringe social networks follows these main steps: 1) the content originates from bots operating on a fringe social network; 2) it keeps gaining popularity among fringe users; 3) some of them start sharing it across mainstream networks. As an example, [Azizpour et al., 2018] reports that fake news websites dominated the discourse on Parler the week before the Capitol attacks, with a Macedonian clickbait site called Resist the Mainstream contributing the most. Such content would have been swiftly blocked on a mainstream Social Network, instead, it propagated on Parler for weeks before eventually reaching the general public. This process is so fast that the moderators' work on mainstream social networks often become pointless.

1.1 Fringe networks vs mainstream networks

The main feature of fringe social networks is the almost total lack of a moderation system which is also the reason why in the last years these niches of the online community have attracted a growing and worrying number of users, especially those who were banned or suspended from mainstream social network for violating terms of service. In [Zannettou, 2019] an overview of some popular fringe networks is presented to give a highlight on the main features and differences between fringe and mainstream social network.

Probably one of the most clear example of fringe social network is **4chan**, a discussion forum based on images. Users can create and share a post that must include an image, to which other users can reply with a text comment and an optional image or a quote to a previous post in the thread. The main characteristic of the 4chan community is the *anonymity*: having a registered account is not required to create a post. Furthermore, those who publish a post can choose a pseudonym that can be different for any post submitted to the community. 4chan is divided in sub-communities called *boards* dedicated to specific topics defined by 4chan. Of particular interest in recent bibliography is the so called Politically Incorrect board (/pol/) which has been shown to exhibit a dangerous degree of hate speech and racism [Hine et al., 2016].

Another main feature of 4chan is the *ephemerality*: threads are removed after an often short time based on a system that bumps new threads whenever they receive a new comment. If a thread is inactive and doesn't receive new comments, it is moved to the down page, and eventually it is removed permanently [Bernstein et al., 2021].

Moderation on 4chan is extremely lax, for each board there are some users called *janitors* that should handle the moderation process by removing harmful posts and banning malicious users. However, they generally allow pretty much everything to be posted.

Reddit is a very popular news aggregator and discussion website. Threads consist in a URL and a title to which other users can respond. Posts can increase popularity through a voting system that set a post score. In contrast with mainstream social networks, on Reddit the friendship/follower relation is not relevant to the structure of the platform on the user-based front.

Users are grouped in sub-communities called Subreddits which differ in topics and the moderation system is monitored by Reddit's administrators who have the power to remove inappropriate contents.

Gab is a relatively new social network born in 2016 combining some existing features of Twitter and Reddit. Posts, called 'gabs', are submitted as text messages with a limit on the characters in a community based on the friendship/follower relation as on Twitter. A voting score is adopted as on Reddit. Users can repost,

comment and use hashtags within their posts. Gab also allows the posting of obscene and malicious contents as long as they are labeled as Not-Safe-For-Work (NSFW). Moderation is very little as almost everything is allowed to be posted. Some exceptions are illegal pornography, terrorism promoting posts and doxxing other users' information.

1.2 Parler

Among these communities, Parler has gained a relatively wide audience since it was endorsed by several political figures in the period of the 2020 US Presidential Election.
Parler was officially launched in 2018 by John Matze Jr. and Jared Thomson on the promise of being a free-speech and unbiased alternative to mainstream social networks such as Twitter and Facebook. The service remained relatively unknown until it started attracting some public and political figures, especially Republican personalities as well as a consistent part of users coming from other platform where they received limitation, censorship or ban.

In the wake of the 2020 US President election Parler experienced a surge in the number of new registered users as it became a hub for the then-President candidate Donald Trump supporters, making Parler one of the most downloaded Apps on the Apple App Store [Constine, 2021].

Parler was among the social media services that were used to coordinate the riot at the US Capitol on January, 6 2021. Few days after, following the ban of Donald Trump from Twitter for breaking the social guidelines in promoting social violence, Parler experienced a last and massive wave of downloads. Following these facts the application was officially taken down from Google and Apple services as well as the Amazon Web Services. After a long lawsuit, Parler resumed services on February 15, 2021, and a new version of the app with added content filters was released on all download services after a statement from the company that US singer and runner-up candidate for the 2024 Presidential Elections, Kanye West had agreed to buy ownership of the platform.

1.2.1 Parler features

At the time that our dataset can cover (i.e. from the birth of Parler to its removal), users on Parler are presented to a feed posts called 'parleys' published from followed accounts that appear chronologically. Contents of posts can only be searched by hashtags, and not by the text content within the post. Each post can be 'voted' or 'echoed' (i.e. retweeted) by the creator's follower. A system of direct messaging is also available allowing user to privately contact each other. At the time covered by our dataset Parler guaranteed anonymity on its platform, user were not requested

to provide personal information at the time of registration. Parler adopts a voting mechanism based on up-voting and down-voting. Posts can only receive up-votes and comments to posts can be up-voted and down-voted as well [Aliapoulios et al., 2021].

1.2.2 Parler's moderation

Moderation on Parler is very minimal, and even its founders have stated that fact-checking is not contemplated. Parler's guidelines disallow and discourage some content like blackmailing, support for terrorism, false rumors and promoting drugs.

Despite some attempts by the founders to establish a collective of volunteers to supervise the contents, as January 2021, Parler executives acknowledge that rules-violating content had remained on the platform [Wikipedia, 2023]. The reason was attributed to the moderator inefficiency in processing and supervising all the malicious content that were created. Moreover, this period was marked by an increase in violence exaltation on the platform following the results of the US Presidential Elections, that led to the assault to the US Capitol on January, 6. As lately shown [Rondeaux and Dalton, 2022], many of the rioters adopted Parler as a place to coordinate the attack.

	Followers	Likes	Repost	Comments	Direct	Limited char. Post	Hashtags	Tag	Anonymous	Ban
	✓	✓	✓	✓	✓	280	✓	✓		✓
		UP/ DOWN	✓	✓	✓					
4chan				✓					✓	
PARLER		UP-Vote	✓	✓	✓		✓	✓	✓	

Figure 1.1: Overview of some features (at the time of writing) of a mainstream social network (Twitter) compare to other known fringe social networks

1.3 Review of related literature

The danger that these fringe social networks constitute for the spreading misinformation and malicious content has been widely analyzed in the literature. In [Zannettou, 2019] the influence that these niches of the Web can have on the mainstream social network is shown by performing a detailed statistical analysis of the impact of fake news. It is indeed showed that fake news originally find space in the fringe networks and then manage to propagate also in the mainstream social networks, reaching an enormous number of users. Here, a detailed characterization of fake news is given and then using the framework of the Hawkes processes, the influence of some fringe networks (i.e. Gab, Reddit, 4chan) on the mainstream social networks (Twitter) is presented. The framework of the Hawkes processes was first presented in [Hawkes, 1971] to model the so called 'self-exciting process'. Namely, these are system for which the arrival of some event triggers the likelihood of having another event within a short time interval. Due to this self-excitement character the framework of Hawkes processes has been successfully adopted for prevision on earthquake aftershocks waves [Ogata, 1999], financial prevision [Azizpour et al., 2018] and even to address the problem of latent network discovery in computational neuroscience ([Linderman and Adams, 2015]). More recently Hawkes processes have been adapted ([Morse, 2017]) to the modeling of social influence and, in particular, in the contest of social media influence. Proof can be found in [Rizoiu et al., 2017] where Hawkes processes are used to effectively model a hashtag retweet cascade following a Twitter post.

In [Laub, 2014] a consistent theoretical background of statistical tools is presented to assess the goodness of the fitting of a time sequence to a Hawkes process. [Zhou et al., 2013] give an interesting presentation of a numerical method for the parameters' estimation of a Hawkes process with some prior inference on the social structure.

1.4 Main contribution and outline of the Thesis

In this Thesis we provide several contributions to the modeling and analysis of the information ecosystem, focusing our attention on a dataset of the fringe social network Parler that covers the period surrounding the first impeachment of US President Donald Trump which eventually led to the Capitol Hill riot on January 2021 and the subsequent shutting down of Parler from all the mainstream download platforms. In more details:

- In Chapter 2 we present a coherent review of the literature about the framework of the Hawkes Processes. We start by the definition of standard point

process, and then we show how this framework naturally extends the structure of the non-homogeneous Poisson processes. We also report a first simple example of synthetic generated data from a Hawkes process.

- In Chapter 3 we address the problem of Hawkes parameters estimation, reviewing the existing literature on the construction of the likelihood function and the algorithm for the optimization of the log-likelihood function. We also provide some numerical experiments on synthetic generated data to corroborate our assumption such as statistical goodness of fit evaluation tools.
- In Chapter 4 we present a simple adaptation of the Hawkes model to the Parler dataset to infer some analysis on the most popular hashtags trends during the period of the first impeachment of Donald Trump. In particular, we distinguish two groups of trend that we call *safe trends* and *unsafe trends*. We show how the Hawkes framework is suitable to quantify whether some trends are pumped into the network by some bad actors or other form of manipulation. Finally, a sentiment analysis algorithm allows to unveil the correlation of level of hate and misleading information to the level of attention from these social communities.

1.5 Preliminaries and Notation

We present here some theoretical preliminaries and mathematical notation that are adopted in this Thesis.

- For matrices and vector we adopt the bold notation $\mathbf{A} \in \mathbb{R}^{n \times m}$, $\mathbf{a} \in \mathbb{R}^n$.
- $(\cdot)^{\mathrm{T}}$ is the transpose operator.
- $\mathbb{1}(\cdot)$ is the indicator function $\mathbb{1}_A(x) := \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$.
- Random variables are indicated with the uppercase notation X, Y, their realization with the lowercase notation x, y.
- $X \sim \text{Poisson}(\lambda)$ indicates a Poisson random variable with rate λ.
- $X \sim \text{Exp}(\lambda)$ indicates an exponential random variable with rate λ.
- $X \sim \text{Beta}(\alpha, \beta)$ indicates a beta distributed random variable with parameters α and β.
- $\mathbb{P}(A)$ indicates the probability of event A.

- $\mathbb{E}[X]$ is the expected value of the random variable X.
- $\mathbf{A} \odot \mathbf{B}$ indicates the component-wise (Hadamard) product of two matrices.
- $\|\mathbf{A}\|$, ($\|\mathbf{a}\|$) indicates the matrix (vector) norm.
- trace($\mathbf{A}$) is the trace of the matrix $\mathbf{A}$ i.e. the sum of its diagonal elements.
- $\langle \mathbf{A}, \mathbf{B} \rangle$ indicates the scalar product.
- Q-Q is an acronym to indicate the Quantile-Quantile (plot).
- $\Phi(x)$ is the standard normal cumulative distribution function.
- CDF indicates the cumulative distribution function of a random variable

Chapter 2

Hawkes Processes

2.1 Introduction

We will start by recalling some useful basic definition from standard probability theory.

Definition 2.1.1. (Point process) Let $(\Omega, \mathcal{F}, \mathbb{P})$ be some probability space. Let $(T_i)_{i\in\mathbb{N}^*}$ be a sequence of non-negative random variables such that $\forall i \in \mathbb{N}^*$, $T_i < T_{i+1}$. We call $(T_i)_{i\in\mathbb{N}^*}$ a (simple) point process on $\mathbb{R}_+$

In particular, the variables T_i can represent the times of occurrence of some kind of events (i.e. transactions, customers arrivals in a queue, posts on social networks, etc.)

Definition 2.1.2. (Counting process) Let $(T_i)_{i\in\mathbb{N}^*}$ be a point process.
The process $N(t) = \sum_{i\in\mathbb{N}^*} \mathbb{1}_{\{T_i\leq t\}}$ is called the counting process associated to $(T_i)_{i\in\mathbb{N}^*}$.

Namely, a counting process is a random function defined on time $t \geq 0$ that take integer values 0,1,2,.... Its value is the number of events of the point process by time t. We observe that $N(t)$ is piece-wise constant and has a jump of size 1 at the event times T_i, $i \in \mathbb{N}$. In the following we will make the identification $T_i = t_i$ keeping in mind that the time t_i is a *realization* of the random variable T_i.

We start our analysis by recalling an important class of counting process, the *inhomogeneous Poisson Process*, which are of particular interest to the study of Hawkes Processes.

Definition 2.1.3. (Homogeneous Poisson process) Let τ_1, τ_2, $\tau_3, \ldots$ be independent and identically distributed exponential random variables with rate λ. Let $T_0 = 0$ and for each $n \geq 1$ let

$$T_n = \tau_1 + \tau_2 + \cdots + \tau_n. \tag{2.1.1}$$

For each $t > 0$ define

$$N(t) := \max\{n \geq 0 : T_n < t\} = \text{number of arrivals by time } t. \tag{2.1.2}$$

Then $(N(t) : t \geq 0)$ is a (homogeneous) Poisson process of rate λ.

Moreover, we have that $(N(t) : t \geq 0)$ is a (homogeneous) Poisson process if and only if:

- $N(0) = 0$,
- $N(t)$ has indipendent increments,
- $N(t+s) - N(s) \sim \text{Poisson}(\lambda t)$.

To express the last condition mathematically:

$$\mathbb{P}(N(t) - N(0) = n) = e^{-\lambda t}\frac{(\lambda t)^n}{n!}, \tag{2.1.3}$$

which is the distribution of a Poisson random variable with rate λt.

One critical observation is that the Poisson process is *memoryless*. A point process is memoryless if the distribution of future inter-arrival times depends only on the current state and not on information in the past. This is the case of the Poisson process since the inter-arrival times are exponentially distributed. First, we compute the probability of observing an inter-arrival time τ longer than a certain value t. Since

$$\mathbb{P}(\tau \leq t) = \int_0^t \lambda e^{-\lambda x}\mathrm{d}x = 1 - e^{-\lambda t},\ t \geq 0, \tag{2.1.4}$$

we have that

$$\mathbb{P}(\tau > t) = e^{-\lambda t},\ t \geq 0. \tag{2.1.5}$$

Suppose now we waited a time m during which no events have arrived, the probability that we have to wait a further t times to see an event is given by

$$\mathbb{P}(\tau > t + m | \tau > m) = \frac{\mathbb{P}(\tau > t + m, \tau > m)}{\mathbb{P}(\tau > m)} \tag{2.1.6}$$

$$= \frac{\mathbb{P}(\tau > t + m)}{\mathbb{P}(\tau > m)} = \frac{e^{-\lambda(t+m)}}{e^{-\lambda m}} = e^{-\lambda t} = \mathbb{P}(\tau > t). \tag{2.1.7}$$

So the probability of having to wait an additional t time after having waited m time is the same as the probability of having to wait t time starting at time 0.

The homogeneous Poisson process is often an unrealistic model. In fact, we would like to take into account the fact that some events occur more frequently at specific times rather than others. For this reason we would like to admit intensities that vary with time.

Definition 2.1.4. (Inhomogeneous Poisson process) A process $(N(t) : t \geq 0)$ is an inhomogeneous Poisson process if

- $N(0) = 0$
- $N(t)$ has independent increments;
- it holds that:

$$\begin{aligned} &\mathbb{P}(N(t+h) = n+m | N(t) = n) = \lambda(t)h + o(h), && \text{if } m = 1 \\ &\mathbb{P}(N(t+h) = n+m | N(t) = n) = o(h), && \text{if } m > 1 \\ &\mathbb{P}(N(t+h) = n+m | N(t) = n) = 1 - \lambda(t)h + o(h), && \text{if } m = 0, \end{aligned} \tag{2.1.8}$$

with $\lambda : \mathbb{R}^+ \to \mathbb{R}^+$ called the *intensity function*.

The above properties imply that $N(t+h) - N(t)$ is a Poisson process with parameter (or mean)

$$\mathbb{E}[N(t+h) - N(t)] = \int_t^{t+h} \lambda(s)\mathrm{d}s, \tag{2.1.9}$$

which implies

$$\mathbb{E}[N(t)] = \int_0^t \lambda(s)\mathrm{d}s. \tag{2.1.10}$$

The model written as that is basically a Poisson process with rate that varies with time.

2.2 Hawkes processes

The memoryless property of Poisson processes means that they are unable to capture a dependence on history, or in other words, interaction between events. For example, we may want to model a system for which the occurrence of an event increases the probability of another arrival in the next short interval of time. This is a good assumption if we aim to model the cascade of events on social media. For this particular need we introduce the *Hawkes process* in which the intensity function is only *conditionally* Poisson: that is, given the history of events $\{\mathcal{H}_t\} = \{t_1, t_2, \ldots, t_{N_t}\}$ up to time t, the conditional intensity at t, $\lambda(t \,|\, \mathcal{H}_t)$ is Poisson.

The conditional intensity function is defined as

$$\lambda(t \,|\, \mathcal{H}_t) = \lim_{h \to 0} \frac{\mathbb{P}\{N(t+h) - N(t) = 1 | \mathcal{H}_t\}}{h}, \tag{2.2.1}$$

that is the probability of observing a single event in a small-time interval h given the history of events. In the rest of the work we will use the compact notation $\lambda(t) := \lambda(t \,|\, \mathcal{H}_t)$, always assuming an implicit history of events before time t.

Formally the following relations hold

$$\begin{aligned} &\mathbb{P}(N(t+h) = n + m | N(t) = n | \mathcal{H}_t) = \lambda(t)h + o(h), && \text{if } \ m = 1 \\ &\mathbb{P}(N(t+h) = n + m | N(t) = n | \mathcal{H}_t) = o(h), && \text{if } \ m > 1 \\ &\mathbb{P}(N(t+h) = n + m | N(t) = n | \mathcal{H}_t) = 1 - \lambda(t)h + o(h), && \text{if } \ m = 0. \end{aligned} \tag{2.2.2}$$

In other words, the probability of observing an event during the infinitesimal interval of time $[t, t+h]$ is *linear* with respect to h as $h \to 0$.

Definition 2.2.1. (Hawkes process). Consider a sequence of events $\{(t_i, u_i)\}_{i=1}^n$ consisting of a time t_i and dimension u_i (i.e. the i-th event occurred at time t_i in dimension u_i), for $t_i \in \mathbb{R}^+$ and $u_i \in \mathcal{U} = \{1,2,\dots,U\}$. This sequence is a *Hawkes process* if the conditional intensity function has the parametrized form

$$\lambda_u(t\,; \Theta) = \mu_u + \sum_{i:t_i<t} h_{uu_i}(t - t_i\,; \theta_{uu_i}), \tag{2.2.3}$$

where $\Theta = \{\boldsymbol{\mu}, \theta\}$ is the set of model parameters and $\mathbf{H} = [h_{ij}]$, $h_{**} : \mathbb{R}^+ \to \mathbb{R}^+$ is the matrix of *triggering kernels* (also called *excitation function* or *decay kernel*) which is different for any couple (u, u_i).

We first observe that, if $\mathbf{H} = \mathbf{0}$ we obtain U standard (homogeneous) Poisson processes with constant rates $\boldsymbol{\mu}$. For this reason we can think of $\boldsymbol{\mu}$ as the constant baseline rates of event arrival in our process that are not influenced by any other events in our system. In other words we can imagine that $\boldsymbol{\mu}$ contains the rates of arrival of some *exogenous* events whose arrivals are independent on previous events in the process.

As in [Morse, 2017] we decompose the triggering kernel matrix $\mathbf{H} = [h_{ij}]$ into an *influence matrix* (or *adjacency matrix*) $A = [a_{ij}]$ and an *exponential decay kernel* $\mathbf{G}(t) = [g_{ij}(t)]$, such that $\mathbf{H} = \mathbf{A} \odot \mathbf{G}$ and

$$h_{uu'}(t; a, \omega) := a_{uu'}\, g(t; \omega), \qquad g(t; \omega) = \omega e^{-\omega t} \tag{2.2.4}$$

The choice of an exponential kernel is to model the fact that as an event becomes more distant in time, it has exponentially less effect on the probability of a new event occurring with parameter ω that models the rapidity of this decay.

On the other hand, the adjacency matrix coefficients $a_{uu'}$ model the intensities of influence between different dimensions (including a_{uu}, the self-excitation of a dimension on itself). Intuitively, large value of $a_{uu'}$ indicates that events in u'-th dimension are more likely to trigger an event in the u-th dimension in the future.

In literature, other models have been considered, modifying the structure of the decay kernel. One common choice is the *power law function*:

$$g(t; c, k, p) = \frac{k}{(c + (t - s))^p}. \tag{2.2.5}$$

This kernel is typically used in geological models, like the Omori's law [Ogata, 1999] to predict the rate of aftershocks in seismology. As seen in [Rizoiu et al., 2017] this kernel is also suitable for predicting the size of a comment cascade following a post on Twitter. In that case author chose to implement a *marked Hawkes process*, for which every event is associated with a different kernel function, modulated in relation to the importance (i.e. number of followers) of the user that commented the post

$$g_m(t; \kappa, \beta, c, \theta) = \kappa m^\beta (\tau + c)^{-(1+\theta)}. \tag{2.2.6}$$

This approach allows for capturing heterogeneity in the data and provides a more nuanced representation of the self-exciting point process. However, our discussion will focus on the exponential kernel, which is more tractable especially when it comes to estimation of parameters, as it reduces the complexity of the model and allows for a more straightforward implementation. Finally, it's important to note that the choice of kernel ultimately depends on the characteristics of the data and the goals of the analysis, and other kernels may be more appropriate in certain scenarios.

We will now define, as in [Laub, 2014], an important tool that will come in help in parameters estimation and goodness of fit testing.

Definition 2.2.2. (Compensator) For a counting process $N(\cdot)$, with conditional intensity function $\lambda(\cdot)$ the non-decreasing function

$$\Lambda(t) = \int_0^t \lambda(s) \mathrm{d}s, \tag{2.2.7}$$

is called the *compensator* of the counting process.

In elementary probability theory it is well known that any inhomogeneous Poisson process may be rescaled into a homogeneous Poisson process with unit rate ([Taylor and Karlin, 1994]). More precisely, if $\{0 < t_1, t_2, \ldots, t_n < T\}$ is a realization from an inhomogeneous Poisson process with intensity $\lambda(t)$, then the transformed time sequence $\{\Lambda(t_1), \Lambda(t_2), \ldots, \Lambda(t_n)\}$ is a realization of a unit rate

Poisson process. A more general result, known as *general time-rescaling theorem* was obtained by [Brown et al., 2002] stating that any point process with an integrable rate function may be rescaled into a Poisson process with rate one.

Theorem 2.2.3. *(General time-rescaling theorem). Let* $0 < t_1, t_2, \ldots, t_n < T$ *be a realization from a point process with conditional intensity* $\lambda(t)$ *such that* $\lambda(t) > 0$ $\forall t$ *in* $(0, T]$ *and* $\Lambda(t) < \infty$ *with probability one* $\forall t$ *in* $(0, T]$*. Then the transformed sequence* $\{\Lambda(t_1), \Lambda(t_2), \ldots, \Lambda(t_n)\}$ *is a realization of a unit rate Poisson process.*

Proof. Let $\tau_k = \Lambda(t_k) - \Lambda(t_{k-1})$ for $k = 1, \ldots, n$ and set

$$\tau_T = \int_{t_n}^{T} \lambda(s)\mathrm{d}s. \tag{2.2.8}$$

It is now sufficient to show that $\{\tau_k\}_{k=1}^n$ are independent and identically distributed exponential random variables with rate one. Since the transformation is one-to-one and $\tau_{n+1} > \tau_t$ if and only if $t_{n+1} > T$, the joint probability density of the τ_k's is

$$f(\tau_1, \tau_2, \ldots, \tau_n \cap \tau_{n+1} > \tau_T) = f(\tau_1, \tau_2, \ldots, \tau_n)\mathbb{P}(\tau_{n+1} > \tau_T | \tau_1, \tau_2, \ldots, \tau_n). \tag{2.2.9}$$

We observe that the following two events are equivalent

$$\{\tau_{n+1} > \tau_T | \tau_1, \tau_2, \ldots, \tau_n\} = \{t_{n+1} > T | t_1, t_2, \ldots, t_n\}. \tag{2.2.10}$$

Hence,

$$\mathbb{P}(\tau_{n+1} > \tau_T | \tau_1, \tau_2, \ldots, \tau_n) = \mathbb{P}(t_{n+1} > T | t_1, t_2, \ldots, t_n) \tag{2.2.11}$$

$$= \exp\left\{ - \int_{t_n}^{T} \lambda(s)\mathrm{d}s \right\} = \exp\{-\tau_T\}. \tag{2.2.12}$$

We now express the first term on the right-hand side of Equation(2.2.9) as

$$f(\tau_1, \tau_2, \ldots, \tau_n) = |J| f(t_1, t_2, \ldots, t_n \cap N(t_n) = n) \tag{2.2.13}$$

where J is the Jacobian of the transformation between $\{t_j\}_{j=1}^n$ and $\{\tau_k\}_{k=1}^n$. Since, by definition, each τ_k is a function of t_k, t_{k-1}, J is a lower-triangular matrix and so its determinant is simply the product of its diagonal elements

$$|J| = \left| \prod_{k=1}^{n} J_{kk} \right|. \tag{2.2.14}$$

Since the mapping is one-to-one and $\lambda(t) > 0$, by the inverse differentiation theorem, the diagonal elements of J are

$$J_{kk} = \frac{\partial t_k}{\partial \tau_k} = \lambda(t_k)^{-1}, \tag{2.2.15}$$

putting all together we have

$$f(\tau_1, \tau_2, \ldots, \tau_n) = \prod_{k=1}^{n} \lambda(t_k)^{-1} \prod_{k=1}^{n} \lambda(t_k) \exp\left\{ - \int_{t_{k-1}}^{t_k} \lambda(s) \mathrm{d}s \right\} \tag{2.2.16}$$

$$= \prod_{k=1}^{n} \exp\{-[\Lambda(t_k) - \Lambda(t_{k-1})]\} \tag{2.2.17}$$

$$= \prod_{k=1}^{n} \exp\{-\tau_k\}. \tag{2.2.18}$$

Substituting Equations (2.2.18) and (2.2.12) in Equation (2.2.9) yields

$$f(\tau_1, \tau_2, \ldots, \tau_n \cap \tau_{n+1} > \tau_T) = f(\tau_1, \tau_2, \ldots, \tau_n) \mathbb{P}(\tau_{n+1} > \tau_T | \tau_1, \tau_2, \ldots, \tau_n) = \tag{2.2.19}$$

$$= \left(\prod_{k=1}^{n} \exp\{-\tau_k\} \right) \exp\{-\tau_T\}, \tag{2.2.20}$$

which is indeed the joint probability densities of n i.i.d. exponential random variables with unitary rate. ■

2.3 Simulation of a Hawkes Process

In this section we will show how to simulate a Hawkes process with exponential decay kernel starting from known parameters $\Theta = (\mu, \mathbf{A}, \omega)$. As we will show later, simulation from known parameters is very useful to compare the results of fitting methods to some "ground truth".

We will generate some example of synthetic Hawkes sequence using the Python library `tick.hawkes` available at [Bacry, accessed 2023-02-15].

From Python module 'tick.hawkes' we simulate a multivariate Hawkes process over time interval [0,50] with $U = 3$ and an exponential decay kernel with fixed parameter $\omega = 1$. The adjacency matrix and the baseline intensities are engineered as follows:

- only the first process u_0 has a non-zero baseline intensity;
- u_0 has influence on itself and on u_1;
- u_1 has influence on u_2.

To sum up our simulation data we have:

$$\mathcal{U} = \{u_0, u_1, u_2\} \tag{2.3.1}$$
$$\omega = 1 \tag{2.3.2}$$
$$\boldsymbol{\mu} = [0.3,\ 0,\ 0] \tag{2.3.3}$$
$$\mathbf{A} = \begin{bmatrix} 0.1 & 0 & 0 \\ 0.2 & 0 & 0 \\ 0 & 0.4 & 0 \end{bmatrix} \tag{2.3.4}$$
$$T = 50 \tag{2.3.5}$$

The results of the simulation are shown in Figure 2.1 in which we can see the three intensity functions of the streamlines in the process as well as the new events occurred (the dots in the plot).

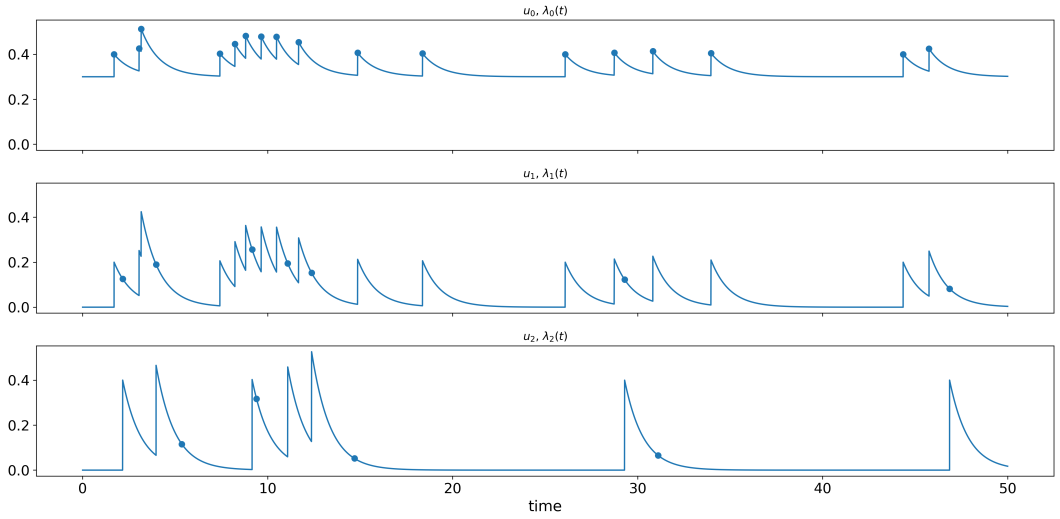

Figure 2.1: Simulation of a 3-dimensional Hawkes process, the figure shows the three intensity functions as well as the event arrivals of the process. The process u_0 has influence on itself and u_1, while u_1 has influence on u_2. All background rates are set to zero except for u_0.

The choice of the model parameters was made such that if we see an event on u_1 or u_2 we know for sure that it comes from a parent event on u_0 or u_1, respectively, and not from an *exogenous* event.

We can well observe the self-excitement character of the process: the arrivals of new events on u_0 cause spikes in the intensity $\lambda_1(t)$, which leads to new events on u_2 that cause more spikes in $\lambda_2(t)$ that cause new events on u_3, resulting in a

burst of activity. We can also appreciate some clusters of events on process u_0 due to its self-excitement character.

Hawkes processes indeed happen to model very well how some actual systems works. One main example is the behavior of users on social networks where the occurrence of one event, such as a post or a comment, can trigger a cascade of subsequent events.

Chapter 3

Parameters estimation

3.1 Maximum likelihood derivation

The main challenge of modelling self-excitement processes is the parameters estimation starting from observed data. We will start our discussion from a monodimensional point process over the timeline, and later we will provide the natural extension at the multidimensional case. Let $N(t)$ a point process on $[0, T]$ with associated conditional intensity $\lambda(t; \Theta)$ and let $\{t_1, t_2, \ldots, t_n\}$ be a realization of the point process, i.e. the event times of the process in the interval $[0, T]$. Then the data likelihood as a function of the parameter set Θ is

$$L(\Theta) = \left(\prod_{i=1}^{n} \lambda(t_i; \Theta)\right) \exp\left\{-\int_0^T \lambda(s)\mathrm{d}s\right\}. \tag{3.1.1}$$

Proof. Let $\mathcal{H}_t = \{t_1, t_2, \ldots, t_n\}$ be the history of events time up to time t. As in [Daley and Vere-Jones, 2008] we introduce the notation $f^*(t) := f(t|\mathcal{H}_t)$ to indicate the conditional probability density function of the time of next time event t_{n+1} given the history of previous event times.

We have that

$$f(t_1, t_2, \ldots, t_n) = \prod_{i=1}^{n} f(t_i|t_1, t_2, \ldots, t_{i-1}) = \prod_{i=1}^{n} f^*(t_i) \tag{3.1.2}$$

We now introduce an equivalent definition of the conditional intensity function, first proposed by [Rasmussen, 2018],

$$\lambda(t) = \frac{f^*(t)}{1 - F^*(t)}. \tag{3.1.3}$$

This is often referred to as the *hazard function*, and it basically corresponds to the ratio between the probability that there is an event in dt and the probability of no new events before time t. Mathematically, consider a small-time interval $\mathrm{d}t$ around t, then

$$\begin{aligned}
\lambda(t)\mathrm{d}t &= \frac{f^*(t)\mathrm{d}t}{1-F^*(t)} && (3.1.4)\\
&= \frac{\mathbb{P}(t_{n+1}\in[t,t+\mathrm{d}t]|\mathcal{H}_{t_n})}{\mathbb{P}(t_{n+1}\notin(t_n,t)|\mathcal{H}_{t_n})} && (3.1.5)\\
&= \frac{\mathbb{P}(t_{n+1}\in[t,t+\mathrm{d}t],t_{n+1}\notin(t_n,t)|\mathcal{H}_{t_n})}{\mathbb{P}(t_{n+1}\notin(t_n,t)|\mathcal{H}_{t_n})} && (3.1.6)\\
&= \mathbb{P}(t_{n+1}\in[t,t+\mathrm{d}t]|t_{n+1}\notin(t_n,t),\mathcal{H}_{t_n}) && (3.1.7)\\
&= \mathbb{P}(t_{n+1}\in[t,t+\mathrm{d}t]|\mathcal{H}_{t^-}) && (3.1.8)\\
&= \mathbb{E}[N([t,t+\mathrm{d}t])|\mathcal{H}_{t^-}], && (3.1.9)
\end{aligned}$$

where $\mathcal{H}_{t^-}$ is the history of all time events up to but not including time t. We showed that the ratio is equivalent to the expectation of an increment of the counting process $N(t+\mathrm{d}t)-N(t)$, which by Equation (2.1.8) is essentially $\lambda(t)\mathrm{d}t$.

We can now continue with the proof writing

$$\lambda(t)=\frac{f^*(t)}{1-F^*(t)}=\frac{\frac{\partial}{\partial t}F^*(t)}{1-F^*(t)}=-\frac{\partial}{\partial t}\log(1-F^*(t)). \tag{3.1.10}$$

Integrating both side in (t_n,t)

$$\int_{t_n}^{t}\lambda(s)\mathrm{d}s=-[\log(1-F^*(t))-\log(1-F^*(t_n))]. \tag{3.1.11}$$

Since $t_{n+1}>t_n$ we have $F^*(t_n)=0$ and so

$$\int_{t_n}^{t}\lambda(s)\mathrm{d}s=-\log(1-F^*(t)), \tag{3.1.12}$$

from which

$$F^*(t)=1-\exp\left(-\int_{t_n}^{t}\lambda(s)\mathrm{d}s\right). \tag{3.1.13}$$

Combining this results in Equation (3.1) gives

$$f^*(t)=\lambda(t)(1-F^*(t))=\lambda(t)\exp\left(-\int_{t_n}^{t}\lambda(s)\mathrm{d}s\right). \tag{3.1.14}$$

Plugging this last equation in Equation (3.1) we get the likelihood expression

$$L(\Theta) = \prod_{i=1}^{n} f^*(t_i) = \prod_{i=1}^{n} \lambda(t_i) \exp\left(-\int_{t_{i-1}}^{t_i} \lambda(s)\mathrm{d}s\right) \\ = \left(\prod_{i=1}^{n} \lambda(t_i)\right) \exp\left(-\int_{0}^{t_n} \lambda(s)\mathrm{d}s\right). \tag{3.1.15}$$

■

3.2 Maximum likelihood estimation

The maximum likelihood estimate of the Hawkes process can be found maximizing the likelihood function, as defined in Equation (3.1), with respect to θ over the space parameter Θ. Then the maximum likelihood estimate is defined as follows

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}}\ L(\theta). \tag{3.2.1}$$

It is a common practice to handle with the logarithm of the likelihood function, the so called *log-likelihood*

$$\mathcal{L}(\theta) := \log L(\theta) = -\int_0^T \lambda(t)\mathrm{d}t + \sum_{i=1}^{N(T)} \log \lambda(t_i). \tag{3.2.2}$$

Since the logarithm is a monotonic function, maximizing the log-likelihood automatically implies maximizing the likelihood function. Furthermore, exploiting the concavity of the function, it's easier to minimize the negative log-likelihood, resulting in the following problem

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}}\ -\mathcal{L}(\theta) = \underset{\theta \in \Theta}{\operatorname{argmin}}\ \int_0^T \lambda(t)\mathrm{d}t - \sum_{i=1}^{N(T)} \log \lambda(t_i). \tag{3.2.3}$$

3.3 Multidimensional Hawkes Processes

We now extend the above discussion to the case of multidimensional Hawkes Processes as we are often interested in modelling the influence between different streamlines of events. We first recall the formulation of the conditional intensity function for the case of U Hawkes processes

$$\lambda_u(t) = \mu_u + \sum_{i:t_i<t} a_{uu_i} g(t - t_i), \tag{3.3.1}$$

where we have explicitly written the decomposition introduced in Section 2.2 for the decay kernel. We collect the parameters into matrix-vector forms, $\boldsymbol{\mu} = (\mu_u)$ for the baseline intensities, and $\mathbf{A} = (a_{uu'})$ for the exciting coefficients. We borrow the terminology from [Zhou et al., 2013], and we call $\mathbf{A}$ the *infectivity matrix*. As a constraint we impose $\mathbf{A} \geq 0$ and $\boldsymbol{\mu} \geq 0$ indicating that we require matrix objects with non-negative entries.

Consider now a realization of a U dimensional Hawkes Process as a collection of the form $\{(t_i, u_i)\}_{i=1}^n$ where the couple (t_i, u_i) indicates that time events t_i has occurred on dimension u_i with $u_i \in \{1,2,\ldots,U\}, i \in \{1,2,\ldots,n\}$. The log-likelihood, as a function of the parameter set $\Theta = \{\mathbf{A}, \boldsymbol{\mu}\}$ can be expressed as follows

$$\mathcal{L}(\mathbf{A}, \boldsymbol{\mu}) = \sum_{i=1}^{n} \log \lambda_{u_i}(t_i) - \sum_{u=1}^{U} \int_0^T \lambda_u(s) \mathrm{d}s \tag{3.3.2}$$

$$= \sum_{i=1}^{n} \log \Big[\mu_{u_i} + \sum_{t_j < t_i} a_{u_i u_j} g(t_i - t_j) \Big] + \tag{3.3.3}$$

$$- \sum_{u=1}^{U} \int_0^T \Big[\mu_u + \sum_{i: t_i < s} a_{u u_i} g(s - t_i) \Big] \mathrm{d}s \tag{3.3.4}$$

$$= \sum_{i=1}^{n} \log \Big[\mu_{u_i} + \sum_{t_j < t_i} a_{u_i u_j} g(t_i - t_j) \Big] + \tag{3.3.5}$$

$$- T \sum_{u=1}^{U} \mu_u - \sum_{u=1}^{U} \int_0^T \sum_{i: t_i < s} a_{u u_i} g(s - t_i) \mathrm{d}s. \tag{3.3.6}$$

To proceed with the calculation we present the following observation:

Observation 3.3.1. Let $\{f_i\}_{i=1}^n$ be a collection of function and $\{F_i\}_{i=1}^n$ such that $F_i' = f_i,\ i = 1 \ldots n$. Let $\{t_i\}_{i=1}^n$ be the collection of event times in the interval $[T_0, T = T_{n+1}]$. The following relation holds

$$\int_0^T \sum_{t_i < t} f_i(T - t_i) \mathrm{d}t = \sum_{i=1}^{n} \{F_i(t - t_i) - F_i(0)\}. \tag{3.3.7}$$

Proof. We split the integration interval in n sub-intervals

$$\begin{aligned}
\int_0^T \sum_{t_i<t} f_i(t-t_i)\mathrm{d}t &= \sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} \sum_{j=1}^{i} f_j(t-t_j)\mathrm{d}t \\
&= \sum_{i=1}^{n}\sum_{j=1}^{i} \int_{t_i}^{t_{i+1}} f_j(t-t_j)\mathrm{d}t \\
&= \sum_{i=1}^{n}\sum_{j=1}^{i} \{F_j(t_{i+1}-t_j) - F_j(t_i-t_j)\} \\
&= \sum_{j=1}^{n}\sum_{i=j}^{n} \{F_j(t_{i+1}-t_j) - F_j(t_i-t_j)\} \\
&= \sum_{j=1}^{n} \{F_j(t_{n+1}-t_j) - F_j(0)\},
\end{aligned} \tag{3.3.8}$$

where in the last step we recognized the sum of a telescoping series over the index i. ■

Using this result we can express the log-likelihood as follows

$$\mathcal{L}(\mathbf{A},\boldsymbol{\mu}) = \sum_{i=1}^{n} \log\Big[\mu_{u_i} + \sum_{t_j<t_i} a_{u_i u_j} g(t_i-t_j)\Big] + \tag{3.3.9}$$

$$- T\sum_{u=1}^{U}\mu_u - \sum_{u=1}^{U}\int_0^T \sum_{i:t_i<s} a_{uu_i} g(s-t_i)\mathrm{d}s \tag{3.3.10}$$

$$= \sum_{i=1}^{n} \log\Big[\mu_{u_i} + \sum_{t_j<t_i} a_{u_i u_j} g(t_i-t_j)\Big] + \tag{3.3.11}$$

$$- T\sum_{u=1}^{U}\mu_u - \sum_{u=1}^{U}\sum_{j=1}^{n} a_{uu_j} G(T-t_j). \tag{3.3.12}$$

Where $G(t) = \int_0^t g(s)\mathrm{d}s$.

3.3.1 Sparsity and Low-Rank regularization

Since we are modelling social influence over a network of people we can infer some a priori details about the structure of the infectivity matrix $\mathbf{A}$. As in [Zhou et al., 2013], we can narrow the space of matrices to the *sparse* and *low-rank* ones. These assumptions are corroborated by the following observations:

- In the context of social networks, the number of connections that a user has is typically much smaller than the total number of users in the network. This mathematically translates into a sparse infectivity matrix.

- The social structure is often organized in communities, meaning that the average user tends to be more connected (i.e. influenced) by people that are geographically and socially closer to him. This leads to a low-rank infectivity matrix.

In order to incorporate this prior knowledge we modify the optimization problem by adding a regularization term as follows:

$$\underset{\mathbf{A}\geq 0,\boldsymbol{\mu}\geq 0}{\operatorname{argmin}} \; -\mathcal{L}(\mathbf{A},\boldsymbol{\mu}) + \lambda_1 \left\|\mathbf{A}\right\|_* + \lambda_2 \left\|\mathbf{A}\right\|_1, \tag{3.3.13}$$

where $\|\mathbf{A}\|_*$ is the nuclear norm of the matrix, and it's defined as the sum of its singular values, $\|\mathbf{A}\|_* := \sum_{i=1}^{\operatorname{rank}(\mathbf{A})} \sigma_i$. It has been shown ([Srebro, 2004]) that this regularization can estimate low-rank matrices effectively.

For the sparse regularization, we use the ℓ_1 norm defined as $\|\mathbf{A}\|_1 := \sum_{i,j} |a_{ij}|$.

The parameters λ_1 and λ_2 control the strength of the regularization terms.

We also define the following matrix scalar product

Definition 3.3.2. (Frobenius inner product) Given $\mathbf{A}$, $\mathbf{B} \in \mathbb{R}^{n\times m}$ we can define a scalar product over the matrices space, called *Frobenius inner product*, as

$$\langle \mathbf{A}, \mathbf{B}\rangle_F := \operatorname{trace}\Big(\mathbf{A}^{\mathrm{T}}\mathbf{B}\Big). \tag{3.3.14}$$

Proof. To prove this we observe that

$$\langle \mathbf{A}, \mathbf{A}\rangle_F = \sum_{i=1}^{n} (\mathbf{A}^{\mathrm{T}}\mathbf{A})_{ii} = \sum_{i=1}^{n}\sum_{j=1}^{m} A_{ij}^{\mathrm{T}} A_{ji} = \sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2 \geq 0, \tag{3.3.15}$$

$$\langle \mathbf{A}, \mathbf{A}\rangle_F = \sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^2 = 0 \iff A_{ij} = 0 \;\forall i,j \iff \mathbf{A} = 0. \tag{3.3.16}$$

Furthermore, since

$$\operatorname{trace}\Big(\mathbf{X}^{\mathrm{T}}\Big) = \operatorname{trace}\Big(\mathbf{X}\Big), \tag{3.3.17}$$

$$\operatorname{trace}\Big(\mathbf{X}+\mathbf{Y}\Big) = \operatorname{trace}\Big(\mathbf{X}\Big) + \operatorname{trace}\Big(\mathbf{Y}\Big), \tag{3.3.18}$$

$$\operatorname{trace}\Big(\lambda\mathbf{X}\Big) = \lambda \operatorname{trace}\Big(\mathbf{X}\Big), \lambda \in \mathbb{R}, \tag{3.3.19}$$

we have also

$$\langle \mathbf{A}, \mathbf{B}\rangle_F = \operatorname{trace}\Big(\mathbf{A}^{\mathrm{T}}\mathbf{B}\Big) = \operatorname{trace}\Big((\mathbf{A}^{\mathrm{T}}\mathbf{B})^{\mathrm{T}}\Big) = \operatorname{trace}\Big(\mathbf{B}^{\mathrm{T}}\mathbf{A}\Big) = \langle \mathbf{B}, \mathbf{A}\rangle_F \tag{3.3.20}$$

$$\begin{aligned}\langle(\lambda\mathbf{A}+\mu\mathbf{B}),\mathbf{C}\rangle_F &= \operatorname{trace}\Big((\lambda\mathbf{A}+\mu\mathbf{B})^{\mathrm{T}}\mathbf{C}\Big)\\ &= \operatorname{trace}\Big((\lambda\mathbf{A})^{\mathrm{T}}\mathbf{C}\Big)+\operatorname{trace}\Big((\mu\mathbf{B})^{\mathrm{T}}\mathbf{C}\Big)\\ &= \lambda\operatorname{trace}\Big(\mathbf{A}^{\mathrm{T}}\mathbf{C}\Big)+\mu\operatorname{trace}\Big(\mathbf{B}^{\mathrm{T}}\mathbf{C}\Big)\\ &= \lambda\langle\mathbf{A},\mathbf{C}\rangle_F+\mu\langle\mathbf{B},\mathbf{C}\rangle_F. \end{aligned} \tag{3.3.21}$$

The above results prove that the *Frobenius scalar product* is indeed a scalar product over the matrices space. ■

This inner product naturally induces the following norm over matrices space

Definition 3.3.3. (Frobenius norm) Given a matrix $\mathbf{A}\in\mathbb{R}^{n_1\times n_2}$ the Frobenius norm of $\mathbf{A}$ is defined as follows

$$\|\mathbf{A}\| = \sqrt{\langle\mathbf{A},\mathbf{A}\rangle_{\mathrm{F}}} = \sqrt{\operatorname{trace}\Big(\mathbf{A}^{\mathrm{T}}\mathbf{A}\Big)} \tag{3.3.22}$$

3.3.2 Optimization with ADMM and EM method

Since the objective function in Equation (3.3.1) is non-differentiable, we have to adopt some numerical method to solve the optimization problem. We choose to apply the idea, as presented in [Zhou et al., 2013], of the Alternating Direction Method of Multipliers (ADMM) that allows us to split the problem into several easier sub-problems.

To derive the steps of ADMM method we consider the equivalent problem obtained by adding two auxiliary variables $\mathbf{Z}_1=\mathbf{A}$, $\mathbf{Z}_2=\mathbf{A}$. The problem becomes

$$\underset{\mathbf{A}\geq 0,\boldsymbol{\mu}\geq 0,\mathbf{Z}_1,\mathbf{Z}_2}{\operatorname{argmin}} \; -\mathcal{L}(\mathbf{A},\boldsymbol{\mu})+\lambda_1\|\mathbf{Z}_1\|_*+\lambda_2\|\mathbf{Z}_2\|_1\,, \tag{3.3.23}$$

$$\text{s.t. } \mathbf{A}=\mathbf{Z}_1,\ \mathbf{A}=\mathbf{Z}_2, \tag{3.3.24}$$

which is equivalent to

$$\underset{\mathbf{A}\geq 0,\boldsymbol{\mu}\geq 0,\mathbf{Z}_1,\mathbf{Z}_2}{\operatorname{argmin}} \; -\mathcal{L}(\mathbf{A},\boldsymbol{\mu})+\lambda_1\|\mathbf{Z}_1\|_*+\lambda_2\|\mathbf{Z}_2\|_1+\frac{\rho}{2}(\|\mathbf{A}-\mathbf{Z}_1\|^2+\|\mathbf{A}-\mathbf{Z}_2\|^2)$$

$$\text{s.t. } \mathbf{A}=\mathbf{Z}_1,\ \mathbf{A}=\mathbf{Z}_2, \tag{3.3.25}$$

where $\rho>0$ is called *penalty parameter*. We can now construct the *augmented Lagrangian* of the constrained problem as follows:

$$\begin{aligned}\mathcal{L}_\rho(\mathbf{A},\boldsymbol{\mu},\mathbf{Z}_1,\mathbf{Z}_2,\mathbf{Y}_1,\mathbf{Y}_2) &= -\mathcal{L}(\mathbf{A},\boldsymbol{\mu})+\lambda_1\|\mathbf{Z}_1\|_*+\lambda_2\|\mathbf{Z}_2\|_1\\ &\quad+\operatorname{trace}(\mathbf{Y}_1^T(\mathbf{A}-\mathbf{Z}_1))+\operatorname{trace}(\mathbf{Y}_2^T(\mathbf{A}-\mathbf{Z}_2))\\ &\quad+\frac{\rho}{2}(\|\mathbf{A}-\mathbf{Z}_1\|^2+\|\mathbf{A}-\mathbf{Z}_2\|^2),\end{aligned} \tag{3.3.26}$$

where $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are the matrices of the dual variables associated with the constraints $\mathbf{A} = \mathbf{Z}_1$ and $\mathbf{A} = \mathbf{Z}_2$. It is often common to deal with the *scaled* form of the ADMM obtained by introducing the following two matrices $\mathbf{U}_1 := \mathbf{Y}_1/\rho$ and $\mathbf{U}_2 := \mathbf{Y}_2/\rho$.

The ADMM algorithm is implemented with the following iterative steps:
for $k \geq 0$:

$$\mathbf{A}^{k+1},\, \boldsymbol{\mu}^{k+1} = \underset{\mathbf{A}\geq 0, \boldsymbol{\mu}\geq 0}{\operatorname{argmin}}\, \mathcal{L}_\rho(\mathbf{A}, \boldsymbol{\mu}, \mathbf{Z}_1^k, \mathbf{Z}_2^k, \mathbf{U}_1^k, \mathbf{U}_2^k) \tag{3.3.27}$$

$$\mathbf{Z}_1^{k+1} = \underset{\mathbf{Z}_1}{\operatorname{argmin}}\, \mathcal{L}_\rho(\mathbf{A}^{k+1}, \boldsymbol{\mu}^{k+1}, \mathbf{Z}_1, \mathbf{Z}_2^k, \mathbf{U}_1^k, \mathbf{U}_2^k) \tag{3.3.28}$$

$$\mathbf{Z}_2^{k+1} = \underset{\mathbf{Z}_2}{\operatorname{argmin}}\, \mathcal{L}_\rho(\mathbf{A}^{k+1}, \boldsymbol{\mu}^{k+1}, \mathbf{Z}_1^k, \mathbf{Z}_2, \mathbf{U}_1^k, \mathbf{U}_2^k) \tag{3.3.29}$$

$$\mathbf{U}_1^{k+1} = \mathbf{U}_1^k + (\mathbf{A}^{k+1} - \mathbf{Z}_1^{k+1}) \tag{3.3.30}$$

$$\mathbf{U}_2^{k+1} = \mathbf{U}_2^k + (\mathbf{A}^{k+1} - \mathbf{Z}_2^{k+1}). \tag{3.3.31}$$

The great advantage of this method is that now we have to deal with different and separate sub-problems that can be optimized one at a time. We will first focus on the problems for $\mathbf{Z}_1$ and $\mathbf{Z}_2$ and then the one for $\mathbf{A}$ and $\boldsymbol{\mu}$

We note that, when solving for $\mathbf{Z}_1$ in Equation (3.3.28) the relevant terms in $\mathcal{L}_\rho$ reduce to

$$\underset{\mathbf{Z}_1}{\operatorname{argmin}}\, \lambda_1 \left\|\mathbf{Z}_1\right\|_* + \operatorname{trace}((\mathbf{U}_1^k)^T(\mathbf{A}^k - \mathbf{Z}_1)) + \frac{\rho}{2}\left\|\mathbf{A}^{k+1} - \mathbf{Z}_1\right\|^2, \tag{3.3.32}$$

which can be simplified in the following problem

$$\underset{\mathbf{Z}_1}{\operatorname{argmin}}\, \lambda_1 \left\|\mathbf{Z}_1\right\|_* + \frac{\rho}{2}\left\|\mathbf{A}^{k+1} - \mathbf{Z}_1 + \mathbf{U}_1^k\right\|^2. \tag{3.3.33}$$

This can be proved by showing the equivalence between the two objective functions. Using the parallelogram law:

$$\begin{aligned}\lambda_1 \left\|\mathbf{Z}_1\right\|_* + \frac{\rho}{2}\left\|\mathbf{A}^{k+1} - \mathbf{Z}_1 + \mathbf{U}_1^k\right\|^2 &= \lambda_1 \left\|\mathbf{Z}_1\right\|_* + \frac{\rho}{2}\left\|\mathbf{A}^{k+1} - \mathbf{Z}_1\right\|^2 + \frac{\rho}{2}\left\|\mathbf{U}_1^k\right\|^2 \\ &\quad + \rho\left\langle \mathbf{U}_1^k, \mathbf{A}^{k+1} - \mathbf{Z}_1\right\rangle_F \\ &= \lambda_1 \left\|\mathbf{Z}_1\right\|_* + \frac{\rho}{2}\left\|\mathbf{A}^{k+1} - \mathbf{Z}_1\right\|^2 + \frac{\rho}{2}\left\|\mathbf{U}_1^k\right\|^2 \\ &\quad + \rho \operatorname{trace}\Big((\mathbf{U}_1^k)^{\mathrm{T}}(\mathbf{A}^{k+1} - \mathbf{Z}_1)\Big).\end{aligned} \tag{3.3.34}$$

Since the term $\frac{\rho}{2}\left\|\mathbf{U}_1^k\right\|^2$ is irrelevant to the minimization with respect to $\mathbf{Z}_1$ we have proved our statement.

We have now to deal with the following problem

$$\mathbf{Z}_1^{k+1} = \underset{\mathbf{Z}_1}{\operatorname{argmin}}\ \lambda_1 \|\mathbf{Z}_1\|_* + \frac{\rho}{2}\left\|\mathbf{A}^{k+1} - \mathbf{Z}_1 + \mathbf{U}_1^k\right\|^2. \tag{3.3.35}$$

It can be shown that there exists a closed form solution

$$\mathbf{Z}_1^{k+1} = \mathcal{S}_{\lambda_1/\rho}(\mathbf{A}^{k+1} + \mathbf{U}_1^k), \tag{3.3.36}$$

where $\mathcal{S}_\alpha(\mathbf{X})$ is called *soft-thresholding operator* defined as

$$\mathcal{S}_\alpha(\mathbf{X}) := \mathbf{U}\,\mathrm{diag}((\sigma_i - \alpha)_+)\mathbf{V}^{\mathrm{T}} \tag{3.3.37}$$

for all matrix $\mathbf{X}$ with singular value decomposition $\mathbf{X} = \mathbf{U}\mathrm{diag}(\sigma_i)\mathbf{V}^{\mathrm{T}}$ and where $(\sigma_i - \alpha)_+ = \max(\sigma_i - \alpha, 0)$.

This is a consequence of the following theorem

Theorem 3.3.4. *$\forall \tau \geq 0$, $\mathbf{Y} \in \mathbb{R}^{n_1 \times n_2}$ the soft-thresholding operator $\mathcal{S}_\tau$ obeys:*

$$\mathcal{S}_\tau(\mathbf{Y}) = \underset{\mathbf{X}}{\mathit{argmin}}\ \{\tau \|\mathbf{X}\|_* + \frac{1}{2}\|\mathbf{X} - \mathbf{Y}\|^2\}. \tag{3.3.38}$$

Proof. The objective function $h_0(\mathbf{X}) := \tau \|\mathbf{X}\|_* + \frac{1}{2}\|\mathbf{X} - \mathbf{Y}\|^2$ is strictly convex, so we know that there exists a unique minimizer. We have to prove that this is equal to $\mathcal{S}_\tau(\mathbf{Y})$. To do this we recall the definition of subgradient of a convex function $f : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}$. We say that $\mathbf{Z}$ is a subgradient of f at $\mathbf{X}_0$, and we denote it as $\partial f(\mathbf{X}_0)$, if

$$f(\mathbf{X}) \geq f(\mathbf{X}_0) + \langle \mathbf{Z}, \mathbf{X} - \mathbf{X}_0 \rangle, \quad \forall \mathbf{X}. \tag{3.3.39}$$

Furthermore, we know that $\hat{\mathbf{X}}$ minimizes h_0 if and only if $\mathbf{0}$ is a subgradient of h_0 in $\hat{\mathbf{X}}$, i.e.

$$\mathbf{0} \in \hat{\mathbf{X}} - \mathbf{Y} + \tau\partial\left\|\hat{\mathbf{X}}\right\|_* \tag{3.3.40}$$

where $\partial\left\|\hat{\mathbf{X}}\right\|_*$ is the set of subgradients of the nuclear norm. Let now $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}$ be the singular value decomposition (SVD) of matrix $\mathbf{X}$. It is known [Candes and Recht, 2008] that

$$\partial\|\mathbf{X}\|_* = \{\mathbf{U}\mathbf{V}^{\mathrm{T}} + \mathbf{W} : \mathbf{W} \in \mathbb{R}^{n_1 \times n_2},\ \ \mathbf{U}^{\mathrm{T}}\mathbf{W} = 0,\ \ \mathbf{W}\mathbf{V} = 0,\ \ \|\mathbf{W}\|_2 \leq 1\}.$$

Set $\hat{\mathbf{X}} := \mathcal{S}_\tau(\mathbf{Y})$. We can decompose the SVD of $\mathbf{Y}$ as

$$\mathbf{Y} = \mathbf{U}_0\mathbf{\Sigma}_0\mathbf{V}_0^{\mathrm{T}} + \mathbf{U}_1\mathbf{\Sigma}_1\mathbf{V}_1^{\mathrm{T}}, \tag{3.3.41}$$

where $\mathbf{U}_0$,$\mathbf{V}_0$ (respectively $\mathbf{U}_1$,$\mathbf{V}_1$) are the factors of the SVD associated with the singular values greater that τ (respectively smaller or equal to τ).

With these definitions we have that

$$\hat{\mathbf{X}} = \mathcal{S}_\tau(\mathbf{Y}) = \mathbf{U}_0(\Sigma_0 - \tau\mathbf{I})\mathbf{V}_0^{\mathrm{T}}, \tag{3.3.42}$$

so that

$$\begin{aligned}\mathbf{Y} - \hat{\mathbf{X}} &= \mathbf{U}_0\mathbf{\Sigma}_0\mathbf{V}_0^{\mathrm{T}} + \mathbf{U}_1\mathbf{\Sigma}_1\mathbf{V}_1^{\mathrm{T}} - \mathbf{U}_0(\Sigma_0 - \tau\mathbf{I})\mathbf{V}_0^{\mathrm{T}} \\ &= \tau\mathbf{U}_0\mathbf{V}_0^{\mathrm{T}} + \mathbf{U}_1\mathbf{\Sigma}_1\mathbf{V}_1^{\mathrm{T}} = \tau(\mathbf{U}_0\mathbf{V}_0^{\mathrm{T}} + \mathbf{W}),\end{aligned} \tag{3.3.43}$$

where $\mathbf{W} := \tau^{-1}\mathbf{U}_1\mathbf{\Sigma}_1\mathbf{V}_1^{\mathrm{T}}$. By construction, we have that $\mathbf{U}_0^{\mathrm{T}}\mathbf{W} = 0$, $\mathbf{W}\mathbf{V}_0 = 0$ and, since the diagonal elements of $\mathbf{\Sigma}_1$ have magnitudes bounded by τ, we also have $\|\mathbf{W}\|_2 \leq 1$.

Thus, we have proven that $\mathbf{Y} - \hat{\mathbf{X}} \in \tau\partial\left\|\hat{\mathbf{X}}\right\|_*$. ■

Similarly, when solving for $\mathbf{Z}_2$ the problem can be simplified as follows:

$$\underset{\mathbf{Z}_2}{\operatorname{argmin}}\ \lambda_2\|\mathbf{Z}_2\|_1 + \frac{\rho}{2}\left\|\mathbf{A}^{k+1} - \mathbf{Z}_2 + \mathbf{U}_1^k\right\|^2, \tag{3.3.44}$$

which is a standard optimization problem with an ℓ_1 norm regularization. It can be shown that also this problem has a closed form solution given by

$$(\mathbf{Z}_2^{k+1})_{ij} = \begin{cases}(\mathbf{A}^{k+1} + \mathbf{U}_1^k)_{ij} - \frac{\lambda_2}{\rho}, & (\mathbf{A}^{k+1} + \mathbf{U}_1^k)_{ij} \geq \frac{\lambda_2}{\rho}, \\ (\mathbf{A}^{k+1} + \mathbf{U}_1^k)_{ij} + \frac{\lambda_2}{\rho}, & (\mathbf{A}^{k+1} + \mathbf{U}_1^k)_{ij} \leq \frac{\lambda_2}{\rho}, \\ 0, & |(\mathbf{A}^{k+1} + \mathbf{U}_1^k)_{ij}| < \frac{\lambda_2}{\rho}\end{cases} \tag{3.3.45}$$

Proof. We compute the subgradient of the equivalent objective function $h_0(\mathbf{Z}_2) = \frac{\lambda_2}{\rho}\|\mathbf{Z}_2\|_1 + \frac{1}{2}\left\|\mathbf{A}^{k+1} - \mathbf{Z}_2 + \mathbf{U}_1^k\right\|^2$

$$\partial h_0(\mathbf{Z}_2) = -(\mathbf{A}^{k+1} - \mathbf{Z}_2 + \mathbf{U}_1^k) + \frac{\lambda_2}{\rho}\partial\|\mathbf{Z}_2\|_1 \tag{3.3.46}$$

We write component-wise the term $\partial\|\mathbf{Z}_2\|_1$ as

$$\frac{\partial}{\partial(\mathbf{Z}_2)_{hk}}\sum_{i,j}|(\mathbf{Z}_2)_{ij}| = \operatorname{sign}((\mathbf{Z}_2)_{ij}), \tag{3.3.47}$$

where $\operatorname{sign}((\mathbf{Z}_2)_{ij}) = \frac{(\mathbf{Z}_2)_{ij}}{|(\mathbf{Z}_2)_{ij}|}$. So we have the following for the subgradient component-wise

$$-(\mathbf{A}^{k+1} + \mathbf{U}_1^k)_{ij} + (\mathbf{Z}_2)_{ij} + \frac{\lambda_2}{\rho}\operatorname{sign}((\mathbf{Z}_2)_{ij}). \tag{3.3.48}$$

We know that $\mathbf{Z}_2^{k+1}$ is a minimizer if and only if it satisfies

$$-(\mathbf{A}^{k+1} + \mathbf{U}_1^k)_{ij} + (\mathbf{Z}_2)_{ij} + \frac{\lambda_2}{\rho}\text{sign}((\mathbf{Z}_2)_{ij}) = 0. \tag{3.3.49}$$

We distinguish three cases

- if $(\mathbf{Z}_2)_{ij} > 0$

$$(\mathbf{Z}_2^{k+1})_{ij} = (\mathbf{A}^{k+1} + \mathbf{U}_1^k)_{ij} - \frac{\lambda_2}{\rho}, \quad \text{if } (\mathbf{A}^{k+1} + \mathbf{U}_1^k)_{ij} > \frac{\lambda_2}{\rho} \tag{3.3.50}$$

- if $(\mathbf{Z}_2)_{ij} < 0$

$$(\mathbf{Z}_2^{k+1})_{ij} = (\mathbf{A}^{k+1} + \mathbf{U}_1^k)_{ij} + \frac{\lambda_2}{\rho}, \quad \text{if } (\mathbf{A}^{k+1} + \mathbf{U}_1^k)_{ij} < -\frac{\lambda_2}{\rho} \tag{3.3.51}$$

- if $(\mathbf{Z}_2)_{ij} = 0$

$$(\mathbf{Z}_2^{k+1})_{ij} = 0, \quad \text{if } |(\mathbf{A}^{k+1} + \mathbf{U}_1^k)_{ij}| \leq -\frac{\lambda_2}{\rho}. \tag{3.3.52}$$

■

The last optimization problem to be addressed, is the one defined in (3.3.27). This has the following equivalent form

$$\mathbf{A}^{k+1}, \boldsymbol{\mu}^{k+1} = \underset{\mathbf{A}\geq 0, \boldsymbol{\mu}\geq 0}{\text{argmin}}\ f(\mathbf{A}, \boldsymbol{\mu}), \tag{3.3.53}$$

where $f(\mathbf{A}, \boldsymbol{\mu}) = -\mathcal{L}(\mathbf{A}, \boldsymbol{\mu}) + \frac{\rho}{2}(\left\|\mathbf{A} - \mathbf{Z}_1^k + \mathbf{U}_1^k\right\|^2 + \left\|\mathbf{A} - \mathbf{Z}_2^k + \mathbf{U}_2^k\right\|^2)$. We solve this using a Majorization-Minimization algorithm which consists, given some currents estimates $\mathbf{A}^{(m)}$ and $\boldsymbol{\mu}^{(m)}$ of $\mathbf{A}$ and $\boldsymbol{\mu}$, in a minimization of a function $Q(\mathbf{A}, \boldsymbol{\mu}; \mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)})$ which is an upper-bound for $f(\mathbf{A}, \boldsymbol{\mu})$.

$$\begin{aligned} Q(\mathbf{A}, \boldsymbol{\mu}; \mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)}) := &-\sum_{i=1}^{n}\left(p_{ii}\log\frac{\mu_{u_i}}{p_{ii}} + \sum_{j=1}^{i-1} p_{ij}\log\frac{a_{u_iu_j}g(t_i - t_j)}{p_{ij}}\right) + \\ &-\left(T\sum_u \mu_u + \sum_{u=1}^{U}\sum_{j=1}^{n} a_{uu_j}G(t - t_j)\right) + \\ &+\frac{\rho}{2}(\left\|\mathbf{A} - \mathbf{Z}_1^k + \mathbf{U}_1^k\right\|^2 + \left\|\mathbf{A} - \mathbf{Z}_2^k + \mathbf{U}_2^k\right\|^2), \end{aligned} \tag{3.3.54}$$

where

$$p_{ii} = \frac{\mu_{u_i}}{\mu_{u_i}^{(m)} + \sum_{j=1}^{i-1} a_{u_i u_j}^{(m)} g(t_i - t_j)}, \tag{3.3.55}$$

$$p_{ij} = \frac{a_{u_i u_j}^{(m)} g(t_i - t_j)}{\mu_{u_i}^{(m)} + \sum_{j=1}^{i-1} a_{u_i u_j}^{(m)} g(t_i - t_j)}. \tag{3.3.56}$$

We can think as p_{ij} as the probability that the i-th event is influenced by a previous event j in the event sequence and p_{ii} as the probability that the event i comes from the baseline intensity.

To prove that Q is an upper bound for f, we can use Jensen's inequality applied to the random variable

$$X := \mu_{u_i} + \sum_{j=1}^{i-1} a_{u_i u_j} g(t_i - t_j). \tag{3.3.57}$$

We know that, given a convex function f,

$$f\Big(\mathbb{E}[X]\Big) \leq \mathbb{E}\Big[f(X)\Big]. \tag{3.3.58}$$

Choosing $f(x) = \log(x)$ we obtain

$$\log\Big(\mu_{u_i} + \sum_{j=1}^{i-1} a_{u_i u_j} g(t_i - t_j)\Big) \geq p_{ii} \log \frac{\mu_{u_i}}{p_{ii}} + \sum_{j=1}^{i-1} p_{ij} \log \frac{a_{u_i u_j} g(t_i - t_j)}{p_{ij}}. \tag{3.3.59}$$

Summing over i we have that

$$Q(\mathbf{A}, \boldsymbol{\mu}; \mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)}) \geq f(\mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)}) \tag{3.3.60}$$

It is also immediate to observe that

$$Q(\mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)}; \mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)}) = f(\mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)}). \tag{3.3.61}$$

The two properties (3.3.60) and (3.3.61) imply that, if

$$(\mathbf{A}^{(m+1)}, \boldsymbol{\mu}^{(m+1)}) = \operatorname*{argmin}_{\mathbf{A}, \boldsymbol{\mu}} Q(\mathbf{A}, \boldsymbol{\mu}; \mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)}) \tag{3.3.62}$$

then

$$\begin{aligned} f(\mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)}) &= Q(\mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)}; \mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)}) \\ &\geq Q(\mathbf{A}^{(m+1)}, \boldsymbol{\mu}^{(m+1)}; \mathbf{A}^{(m)}, \boldsymbol{\mu}^{(m)}) \\ &\geq f(\mathbf{A}^{(m+1)}, \boldsymbol{\mu}^{(m+1)}). \end{aligned} \tag{3.3.63}$$

This ensures that f decreases monotonically. Solving for Q allows us now to separate independently the problems for $\mathbf{A}$ and $\boldsymbol{\mu}$. Furthermore, each of them has a closed form solution [Zhou et al., 2013] given by

$$\mu_u^{(m+1)} = \frac{\sum_{i:i\leq n, u_i=u} p_{ii}}{T}, \tag{3.3.64}$$

$$a_{uu'}^{(m+1)} = \frac{-B + \sqrt{B^2 + 8\rho C}}{4\rho}, \tag{3.3.65}$$

where $B = \sum_{j:u_j=u} G(T - t_j) + \rho(-z_{1,uu'} + u_{1,uu'} - z_{2,uu'} + u_2, uu')$ and $C = \sum_{i=1}^{n} \sum_{j<i, u_j=u'} p_{ij}$.

The full algorithm, named ADM4 in [Zhou et al., 2013], can be schematized as follows:

Algorithm 1 ADM4 for estimating $\mathbf{A}$ and $\boldsymbol{\mu}$

Initialize $\mathbf{A}$ and $\boldsymbol{\mu}$ randomly;
Set $\mathbf{U}_1 = \mathbf{0}$, $\mathbf{U}_2 = \mathbf{0}$;
while $k = 1,2,\ldots$ **do**
    Update $\mathbf{A}^{k+1}$ and $\boldsymbol{\mu}^{k+1}$ as in (3.3.64), (3.3.65).
    Update $\mathbf{Z}_1^{k+1}$ as in (3.3.36)
    Update $\mathbf{Z}_2^{k+1}$ as in (3.3.45)
    Update $\mathbf{U}_1^{k+1} = \mathbf{U}_1^k + (\mathbf{A}^{k+1} - \mathbf{Z}_1^{k+1})$
    Update $\mathbf{U}_2^{k+1} = \mathbf{U}_2^k + (\mathbf{A}^{k+1} - \mathbf{Z}_2^{k+1})$
end while
return $\mathbf{A}$ and $\boldsymbol{\mu}$

3.4 Estimating synthetic data

Algorithm 1 is implemented in the Python library `ticks.hawkes`. The method is called `HawkesADM4` and performs a parametric inference for the particular case of Hawkes processes with exponential decay kernel. The parameter ω that modulates the decay rate is treated as a global variable, and it must be passed to the function as a parameter. Another parameter of the function is `lasso_nuclear_ratio`, a float variable between 0 and 1 that represents the ratio between the ℓ_1 norm penalty and the nuclear norm penalty. The default value is 0.5 which means equal weight to both penalties. To check the performance of the algorithm we first apply it to a synthetic sequence generated from known parameters. We scale $\mathbf{A}$ such that its spectral radius (i.e. the maximum eigenvalue in absolute value) is lesser than

one. This is a necessary and sufficient condition for the well-definition of the point process. The designed parameters are summarized in the following

$$\mathcal{U} = \{u_0, u_1, u_2, u_3, u_4, u_5\} \tag{3.4.1}$$

$$\omega = 3 \tag{3.4.2}$$

$$\boldsymbol{\mu} = [0.3,\ 0.3,\ 0.3,\ 0.3,\ 0.3,\ 0.3] \tag{3.4.3}$$

$$\mathbf{A} = \begin{bmatrix} 0.15 & 0.15 & 0.15 & 0 & 0 & 0 \\ 0.15 & 0.15 & 0.15 & 0 & 0 & 0 \\ 0.15 & 0.15 & 0.15 & 0.1 & 0.1 & 0.1 \\ 0 & 0 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0 & 0 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0 & 0 & 0.1 & 0.1 & 0.1 & 0.1 \end{bmatrix} \tag{3.4.4}$$

As discussed above, we choose to design an adjacency matrix that is sparse and low-rank to better model the social infectivity.

We apply the ADM4 method for an increasing sequence of time events using both the ℓ_1 regularization and the nuclear norm regularization (i.e. we set `lasso_nuclear_ratio`$= 0.5$), and we choose to evaluate the following metrics

- Given $\hat{\mathbf{A}}$, the estimate of $\mathbf{A}$, we evaluate

$$r_{\text{adjacency}} = \max_{ij} r_{ij}, \tag{3.4.5}$$

  where

$$r_{ij} = \begin{cases} \dfrac{|a_{ij} - \hat{a}_{ij}|}{|a_{ij}|}, & \text{for } a_{ij} \neq 0 \\ |a_{ij} - \hat{a}_{ij}|, & \text{for } a_{ij} = 0 \end{cases} \tag{3.4.6}$$

- Given $\hat{\boldsymbol{\mu}}$, the estimate of $\boldsymbol{\mu}$ we evaluate

$$r_{\text{baseline}} = \max_{i} |(\boldsymbol{\mu})_i - (\hat{\boldsymbol{\mu}})_i|. \tag{3.4.7}$$

Results are shown in Figure 3.1. As we can see the error decreases to as the number of time events increases, and it can also be shown [Zhou et al., 2013] that the method outperforms the algorithms where the original sparse and low-structure is not take into account for the estimation of the parameters.

Figure 3.2 shows the estimate adjacency matrix and the values of the entries (i.e. the kernel norms) plotted using the method `plot_hawkes_kernel_norms` implemented in the library `tick.plot`. As we can see the algorithm can well reconstruct the sparse and low-rank structure of the original matrix.

Another interesting plot is the one generated using the method `plot_hawkes_kernels` that shows both the true and the estimated exponential triggering kernels over time. In Figure 3.3 we can see the result of the method for a 2-dimensional Hawkes process with parameters set to

$$\omega = 3 \tag{3.4.8}$$

$$\boldsymbol{\mu} = [0.3,\ 0.3] \tag{3.4.9}$$

$$\mathbf{A} = \begin{bmatrix} 0.15 & 0.1 \\ 0 & 0.1 \end{bmatrix} \tag{3.4.10}$$

$$T = 100000 \tag{3.4.11}$$

3.5 Goodness of fit

Finding some metric to assess the goodness of the assumption that a given time sequence is indeed a realization of a Hawkes process is a main goal of this work. In Chapter 2 we reported what will be the cardinal theorem of our analysis, called *General time-rescaling theorem* (2.2.3) which basically states that any point process with an integrable rate function may be rescaled into a Poisson process with rate one. Therefore, if we can reach a closed form expression for the compensator function, the goodness of our assumption can be evaluated by performing some standard statistical tests for the Poisson and exponential distribution [Laub, 2014]. Namely, the transformed sequence should be a realization of a unitary Poisson process which means that the inter-arrival transformed times sequence should be a realization of an exponential random variable with unitary rate. We now present a closed form expression for a mono-dimensional Hawkes process with an exponential decay kernel function.

We have that

$$\lambda(t) = \mu + \sum_{j:t_j<t} a\omega e^{-\omega(t-t_j)} \tag{3.5.1}$$

We can calculate the *compensator* function as

$$\Lambda(t) = \int_0^t \lambda(s)\mathrm{d}s \tag{3.5.2}$$

Thus,

$$\begin{aligned}\int_0^t \lambda(s)\mathrm{d}s &= \int_0^t \{\mu + \sum_{j:t_j<s} a\omega e^{-\omega(s-t_j)}\}\,\mathrm{d}s \\ &= \mu\, t + \int_0^t \{\sum_{j:t_j<s} a\,\omega e^{-\omega(s-t_j)}\}\,\mathrm{d}s \\ &= \mu\, t + a\sum_{j=1}^{n}\{e^{-\omega(t-t_j)} - 1\},\end{aligned} \tag{3.5.3}$$

where in the last step we used the result obtained in Observation 3.3.1.

So the compensator for a mono-dimensional Hawkes process has the closed expression

$$\Lambda(t) = \mu\, t + a\sum_{j=1}^{n}\{e^{-\omega(t-t_j)} - 1\}. \tag{3.5.4}$$

Once the Expression 3.5.4 is given we can apply it to our time sequence to obtain the transformed sequence

$$\{t_1^*, t_2^*, \ldots, t_n^*\} = \{\Lambda(t_1), \Lambda(t_2), \ldots, \Lambda(t_n)\}, \tag{3.5.5}$$

from which we can extract the inter-arrival transformed sequence

$$\{\tau_1, \tau_2, \tau_3 \ldots, \tau_n\} = \{t_1^*, t_2^* - t_1^*, t_3^* - t_2^*, \ldots, t_n^* - t_{n-1}^*\}. \tag{3.5.6}$$

To assess the goodness of our assumption, statistical test should be performed to ensure that $\tau_i \overset{\text{i.i.d.}}{\sim} \mathrm{Exp}(1)$.

3.5.1 Q-Q plot and approximated Kolmogorov-Smirnov test

To check whether the inter-arrival times actually come from an exponential distribution of unitary rate a first qualitative approach can be the construction of a Q-Q plot of the interarrival times. In a Q-Q plot the sample quantiles are plotted against the quantiles of the known theoretical distribution that we are assuming to be the distribution of the observed data. If the points of the Q-Q lie on the identity line $y = x$ we can assess that the two compared distributions are similar.

To give a simple example, in Figure 3.4 we display the Q-Q plot obtained from a simulated sequence of 10^4 Hawkes events with parameters $(\omega, a, \mu) = (3, 0.1, 0.3)$. To do this we used the method `probplot` available in the Python library `scipy.stats` that takes as arguments the sequence of inter-arrival transformed event times and the theoretical distribution that we assumed to be the one of the data.

As we can see, most of the points lie on the identity line except for some last quantile that fall below the bisector. This can be the result of some numerical

error or a result of the fact that the tails of the sample distribution are lighter than those of the theoretical one, meaning that extreme values are less frequent in the data sample than in the theoretical distribution.

Another qualitative approach is given by [Daley and Vere-Jones, 2003] suggesting an approximate Kolmogorv-Smirnnov-type based on the Brownian motion approximation that is implemented as follows:

1. Give the sequence $\{t_1^*, \dots, t_{N(t)}^*\}$, plot the cumulative step function $Y(x)$ which has the points $(x_i, y_i) = (t_i^*/T, i/N(T))$;
2. Plot the confidence lines $y = x \pm Z_{1-\alpha/2}/\sqrt{T}$, where $\Phi(Z_{1-\alpha/2}) = 1 - \alpha/2$;
3. Accept the hypothesis that $\{t_i^*\}$ come from a unit rate Poisson process if the plot of $Y(x)$ stays within the confidence lines (with $100(1-\alpha)\%$ certainty).

To achieve even better performances, in [Laub, 2014] an alternative Brownian motion approximation test is presented.

Starting from the Poisson process $N(t)$ of rate T, define

$$M(t) := \frac{N(t) - tT}{\sqrt{T}}, \quad \text{for} \quad t \in [0,1]. \tag{3.5.7}$$

It is known that, as $T \to \infty$ then $(M(t), t \in [0,1])$ converges (in the sense of distribution) to the standard Brownian motion $(B(t), t \in [0,1])$. The proposed test is based on the first arcsine law for Brownian motion. This states that the random time $M^* = \underset{s \in [0,1]}{\operatorname{argmin}} B(s)$, is arcsine distributed (i.e. $M^* \sim \text{Beta}(1/2,1/2)$).

The modified algorithm can be eventually written as follows:

1. Give the sequence $\{t_1^*, \dots, t_{N(t)}^*\}$, transform it to $\{t_1^*/T, \dots, t_{N(t)}^*/T\}$ which is a Poisson process of rate T;
2. Constructs the Brownian motion approximation $M(t)$ as in (3.5.7), then finds the maximizer M^*;
3. Accept the hypothesis that $\{t_i^*\}$ come from a unit rate Poisson process if M^* lies within the $(\alpha/2, 1-\alpha/2)$ quantiles of the Beta(1/2,1/2) distribution, otherwise reject it.

The result of the application of the procedure given by [Daley and Vere-Jones, 2003] is shown in Figure 3.5. As we can see the empirical CDF of the data lies within the confidence lines, so we can conclude that the process is indeed well modeled by a Poisson process of unitary rate.

Furthermore, a test with the modified algorithm proposed by [Laub, 2014] was conducted. Even for this case the test led to the acceptance of the unitary Poisson process assumption.

3.5.2 Independence test

To check the independence of the inter-arrival times we use a graphic Python method called `plot_acf` available within the library `statsmodels`. This method is based on the work of [Brockwell and Davis, 2016] and basically plots the autocorrelation function of the data defined, for a generic sequence of event times $\{X_t\}$, as

$$\rho_X(s) = \frac{\text{Cov}(X_{t+s}, X_t)}{\text{Cov}(X_t, X_t)} = \text{Cor}(X_{t+s}, X_t), \tag{3.5.8}$$

where $\text{Cov}(X_r, X_h) = \mathbb{E}[(X_r - \mathbb{E}[X_r])(X_h - \mathbb{E}[X_h])]$. We applied this method to a synthetic mono-dimensional sequence of 10^4 Hawkes event times with parameters $(\omega, a, \mu) = (3, 0.1, 0.3)$. The plot is shown in Figure 3.6

The shaded area represents the confidence region which is set with the default value of $\alpha = 0.05$. That means that anything that falls into this region represent a value with no significant correlation with previous values. The dotted lines are the values of the autocorrelation evaluated at a specific time lag (here we plotted 10 time lags). As we can see, except for the obvious peak at a zero time lag (which means that every random variable has maximum autocorrelation with itself) all the values beyond time lag equals to zero are negligible, and they all fall in the confidence region.

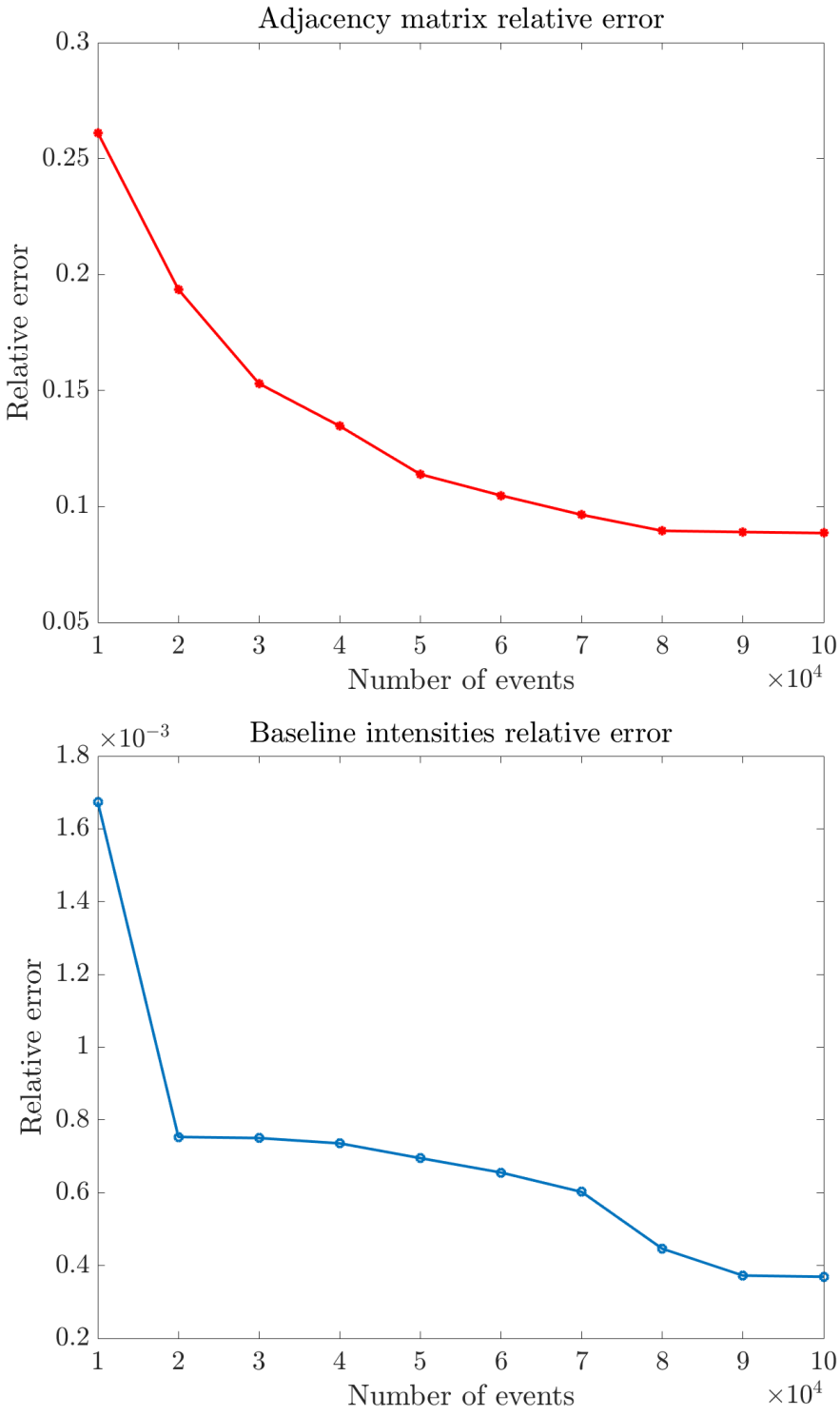

Figure 3.1: Maximum relative errors of the ADM4 method for the estimation of the adjacency matrix and the baseline intensities

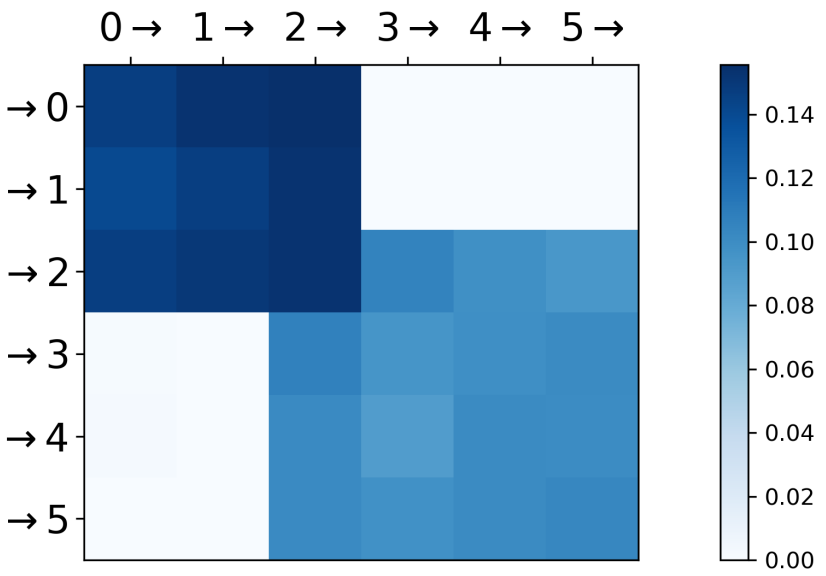

Figure 3.2: Estimated kernel norms for the infectivity matrix

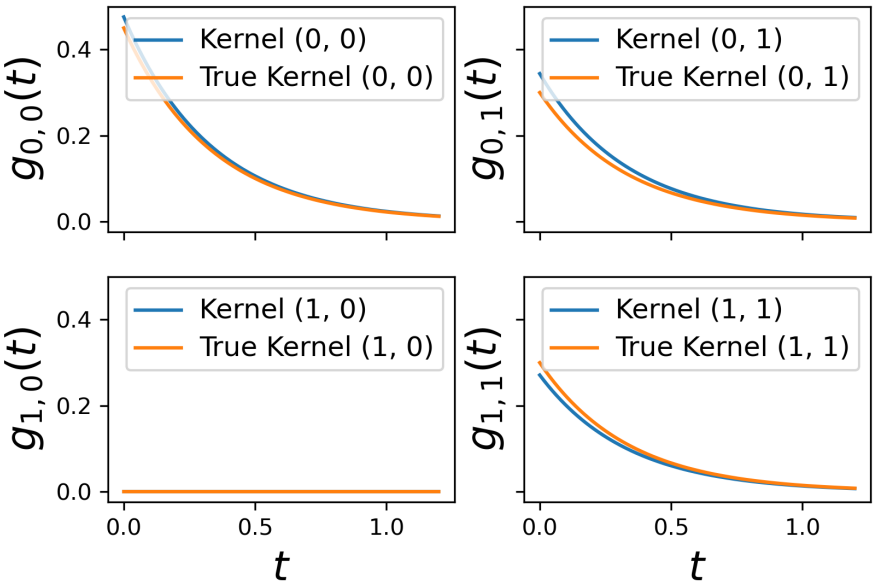

Figure 3.3: True and estimated kernel triggering function for a two-dimensional Hawkes process

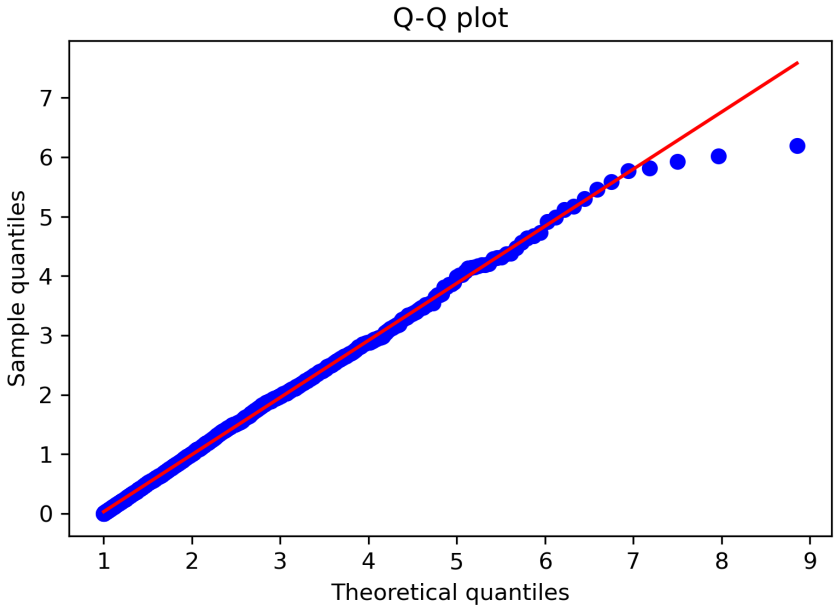

Figure 3.4: Q-Q plot for a synthetic mono-dimensional Hawkes process of size 10^4 with parameters $(\omega,\, a,\, \mu) = (3,\, 0.1,\, 0.3)$

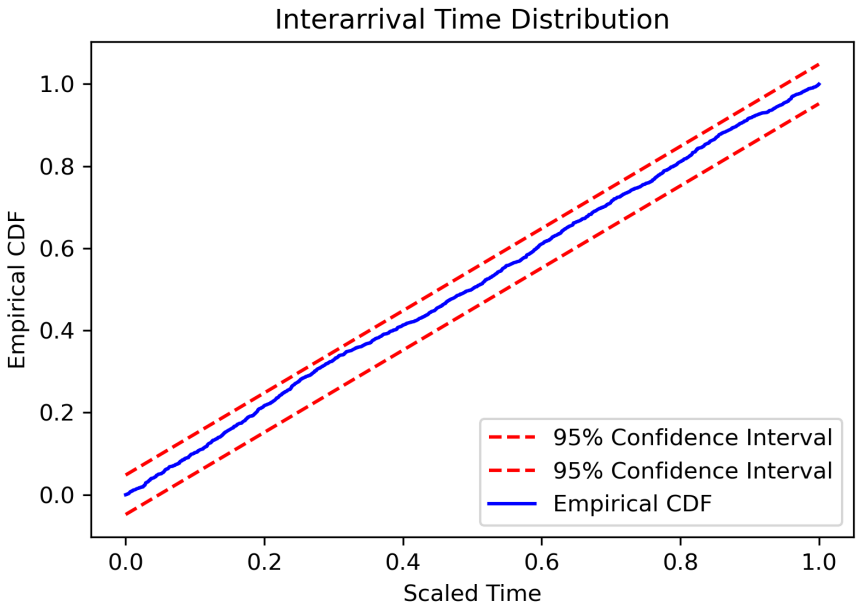

Figure 3.5: Result of the test proposed by [Daley and Vere-Jones, 2003] for a synthetic mono-dimensional Hawkes process of size 10^4 with parameters $(\omega, a, \mu) = (3, 0.1, 0.3)$

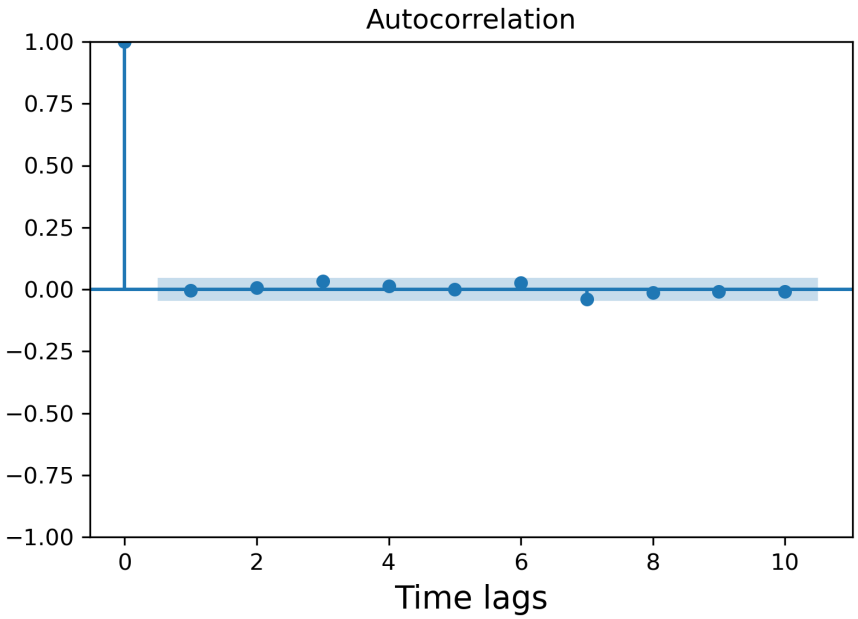

Figure 3.6: Autocorrelation of the transformed interarrival times for a synthetic mono-dimensional Hawkes process of size 10^4 with parameters $(\omega, a, \mu) = (3, 0.1, 0.3)$

Chapter 4

Analysis of trends surrounding Donald Trump's impeachment

4.1 Dataset analysis

For this work, we use the online public dataset given in [Aliapoulios et al., 2021]. This is a dataset of 183.063M Parler posts made by 4.08M users between August 2018 and January 2021, as well as metadata from 143.25M user profiles. Overall, the data consists of newline-delimited JSON files (.ndjson), obtained by crawling three main Parler API (Application Programming Interface). Some main features given by the post/comment API are:

- *id*: Parler generated universally unique ID that is associated to the creator of the post.
- *createdAt*: Timestamp of the post/comment in UTC (Coordinate Universal Time)
- *upvotes*: Number of upvotes that a post/comment received.
- *followers*: Integer number of followers that the creator of the post/comment has.
- *following*: Integer number of accounts followed by the post/comment creator.
- *score*: Number of upvotes minus the sum of the downvotes a post/comment received.

- *hashtags*: List of strings that corresponds to the hashtags used in a post/comment.
- *urls*: List of dictionaries correspond to URLs and their respective id and metadata used in a post/comment.
- *verified*: Boolean value that indicates whether the post creator has a verified account.

A simple visualization of a single post/comment structure is given in Figure 4.1

POST/COMMENT
'id': 1fb8797ba9aa4721a491588805b1dc2c
'upvotes': 14
...
'createdAtformatted': YYYY-MM-DD hh:mm:ss UTC
...
'hashtags': ["hashtag1", "hashtag1", ... , "hashtagN"]
...
'urls': [urldict_1, urldict_2, ... , urldict_N]
...
'verified': TRUE
...
'body': "body of the post"
...

Figure 4.1: Visualization of a single post/comment structure given by the Parler API

4.1.1 Data preparation

Given the dataset, our main purpose is to extract information about the hashtags cascades relative to specific hashtags. To do this we adopt the Python data analysis tool `pandas`. We first proceed to thin out the dataset getting rid of some keys for each post that are irrelevant for our case study.

The 'hashtag' key of a single post is a list of string objects. To extract a convenient dataset we associate each hashtag with the corresponding post id and time of

creation to eventually end up with a data frame in which we have all the hashtags of the original dataset alongside with the id of the post and the creation time.

Figure 4.2 shows a simple visualization of the resulting data frame.

	hasthag	time
0	Hashtag1	20200216231556
1	Hashtag2	20200805000311
2	Hashtag3	20200805000311
3	Hashtag4	20200805000311
4	Hashtag5	20190705123951
5	Hashtag6	20200630073103
6	Hashtag7	20200630073103
7	Hashtag8	20200630073103
8	Hashtag9	20200630073103
9	Hashtag10	20200630073103
	...	

Figure 4.2: Visualization of the data frame used for the analysis

We first note that many of the hashtags in the data frame are associated to the same post id and creation time. This is due to the fact that there is no limitation on how many hashtags a user can include in his post or comment.

4.1.2 Data extraction

Since our aim is to model hashtags cascades with the framework of Hawkes Process we proceed to extract the time events of some hashtags trends. We choose to study two different groups of hashtags that can be distinguished in 'Anti-Trump' and 'Pro-Trump' following the analysis presented in [Rossetti and Zaman, 2022] that was made on a Twitter dataset covering the same period of time as our Parler dataset. The partisanship of the hashtags was inferred through a neural network-based sentiment analysis. The selected trends for this study are shown in Table 4.1.

The selected trends are the ones that directly concern the two main figures of the period covering the 2020 US Presidential Elections, namely the two candidates, Donald Trump and Joe Biden (#trump, #trump2020, #biden, #biden2020). Other

Pro-Trump trends	Anti-Trump trends
#maga	#biden
#qanon	#biden2020
#stopthesteal	#blm
#trump	#impeach
#trump2020	#impeachtrump

Table 4.1: Table of selected trends for this study

Pro-Trump selected trends are #stopthesteal, which is a hashtag and a slogan used by Trump supporters who claimed that the election of Joe Biden was fraudulent and that the result had to be overturned; #maga which is an acronym for 'Make America Great Again', the slogan adopted by Donald Trump during its presidential campaign; #qanon stands for a fair-right conspiracy theory that alleges the existence of a "cabal of Satanic, cannibalistic sexual abusers of children operating a global child sex trafficking ring conspired against former U.S. President Donald Trump during his term in office" [Wikipedia, 2021].

Anti-Trump selected trends include #blm, acronym of 'Black Lives Matter', a movement with the aim of bring attention to racism and violence against Black people communities; #impeach and #impeachtrump referred to the hashtag and calls to action to relieve President Donald Trump from his position through the impeachment process.

Having selected the trends we then proceed to sort the data frame chronologically and to filter it to keep only the selected trends. In this way we can easily extract the time events of each hashtag cascade to be given to our model.

4.2 Modeling Parler with Hawkes processes

We choose to model our extracted streamlines of events as a U-dimensional Hawkes Process with exponential decay kernels. We recall here the definition of the intensity function associated to the model

$$\lambda_u(t) = \mu_u + \sum_{i:t_i<t} a_{uu_i}\omega e^{-\omega(t-t_i)}, \quad u = 1, \dots, U. \tag{4.2.1}$$

A first interpretation of the parameters could be the following:

- μ_u represents the background intensity of process u meaning the rate at which an event on streamline u occurs without being influenced by any other event in the process. In the context of a social network analysis this rate can be interpreted as the rate through which trend u is 'pumped' in the social network by some external agent of information;

- The adjacency matrix coefficient a_{uu_i} models the intensity of influence between streamlines u and u_i. In our analysis context this can intuitively be interpreted as follows: larger values of coefficient a_{uu_i} indicates that an event (i.e. a hashtag appearance) on trend u_i is more likely to trigger a subsequent event (i.e. a consequent hashtag appearance) on trend u. Indeed, matrix $\mathbf{A}$ would give us the underlying influence structure of our network of trends.

- Parameter ω, which is the exponent of the decay kernel, modulate how fast the influence of an event decays over time. In a nutshell, ω is an indication of the 'network memory' of recent events. In the social media context this can be an indication of how long a trend remains influential on the network.

The decay parameter ω deserves particular attention in the development of our analysis. As mentioned in Section 3.4 the algorithm implemented for the estimation of the parameters treats the decay coefficient as a global parameter that should be given to the method. Hence, to perform the estimation we should infer some a priori knowledge on ω. As the event times gathered from the dataset are converted in number of days we choose to put $\omega = 4$ as a 'network memory' of 4 days appears to be a reasonable assumption in the context of a social network dynamic.

4.3 Numerical results

4.3.1 Parameters estimation

Once the streamlines of the two different categories of trends are extracted, we can apply the Algorithm 1 as described in Section 3.3.2 implemented in the `tick` Python library for the estimation of $\boldsymbol{\mu}$ and $\mathbf{A}$. We first proceed with the estimation of the parameters of the complete process. Namely, we apply the method considering all the streamlines of trends of as a multidimensional Hawkes process. The `tick` library method `plot_hawkes_kernel_norms` is then used to show the resulting approximated adjacency matrix $\mathbf{A}$ of the complete process. Plot is shown in Figure 4.3.

The kernels are displayed such that the plot shows norm of column influence's on row. Namely, higher value of entry a_{ij} indicates a high level of influence of process j on process i. A legend to read the labels on the kernel plots is provided in Table 4.2.

As we can see, diagonal values are in general higher than off-diagonal norms. This means that, generally the trend express a most dominant self-excitement behavior rather than an influential one. Another interesting thing to note is that Anti-Trump trends are very unlikely to influence Pro-Trump trends while, on the contrary, an event on a Pro-Trump trend is very likely to influence an event on

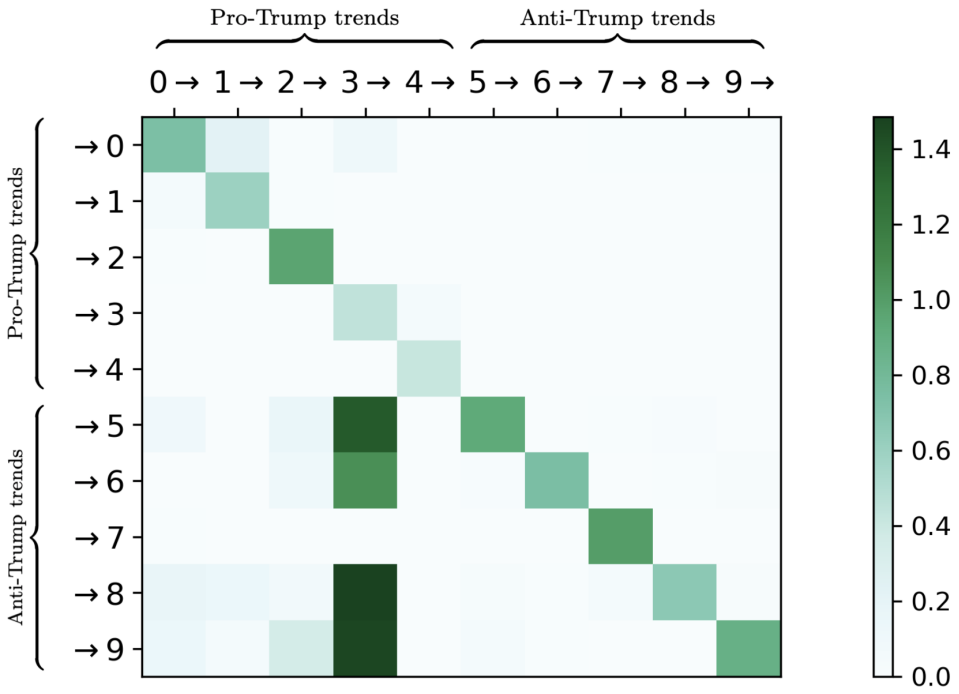

Figure 4.3: Estimated adjacency matrix **A** for the complete process

Trend	**Trend label**
#trump2020	0
#maga	1
#trump	2
#stopthesteal	3
#qanon	4
#biden	5
#biden2020	6
#blm	7
#impeach	8
#impeachtrump	9

Table 4.2: Table of selected trends for this study

the Anti-Trump streamlines. Not surprisingly the greater influence flow is estimated on the trend paths #stopthesteal → #biden, #stopthesteal → #biden2020, #stopthesteal → #impeach and #stopthesteal → #impeachtrump. This is a consequence of the fact that Parler became a huge plaza for Trump supporters after

the outcome of the US Presidential Elections.

To further investigate the underlying influence structure of the network of trends we proceed to analyze the two categories of trend separately, distinguishing the Anti-Trump (AT) trends from the Pro-Trump (PT) trends treating them as separate and independent processes. We apply the estimation algorithm to find the parameters $(\mathbf{A}_{\mathrm{PT}}, \boldsymbol{\mu}_{\mathrm{PT}})$ and $(\mathbf{A}_{\mathrm{AT}}, \boldsymbol{\mu}_{\mathrm{AT}})$.

The output of the algorithm is shown in Figure 4.4 and the associated labels for the trends are displayed in Table 4.3.

We can appreciate a strong self-excitement character for the two group of trends, as the diagonal elements of the adjacency matrix have greater magnitude than off-diagonal entries. In general, the magnitude of the influence is slightly greater for the Pro-Trump trends than for the Anti-Trump ones. This indicates that Pro-Trump events are not only more likely to self-excite themselves but also to trigger other Pro-Trump events.

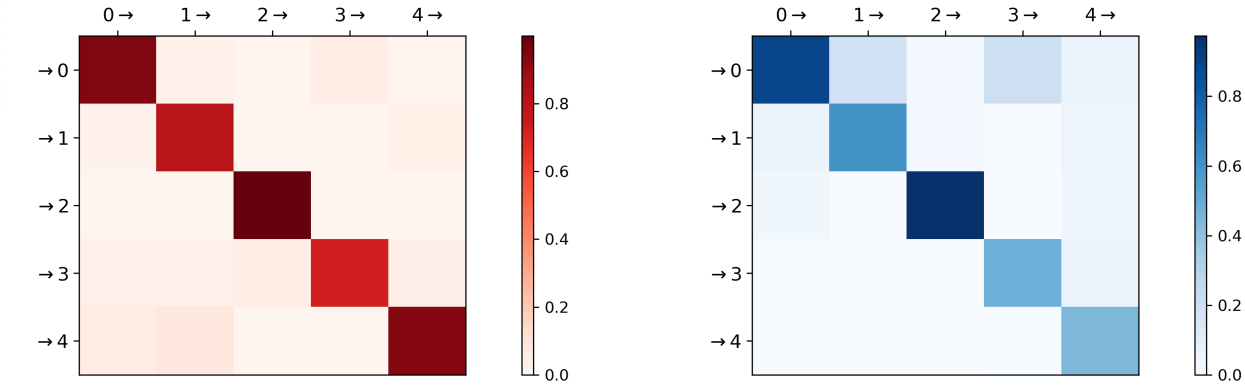

Figure 4.4: Estimated adjacency matrix $\mathbf{A}_{PT}$ for the Pro-Trump process (on the left) and $\mathbf{A}_{AT}$ for the Anti-Trump process (on the right)

Pro-Trump trends	
Trend	**Trend label**
#trump2020	0
#maga	1
#trump	2
#stopthesteal	3
#qanon	4

Anti-Trump Trends	
Trend	**Trend label**
#biden	0
#biden2020	1
#blm	2
#impeach	3
#impeachtrump	4

Table 4.3: Trend labels for the two separate processes

In Table 4.4 and Figure 4.5 we display the estimated background rates for each hashtag trend.

Pro-Trump trends	
Trend	μ
#trump2020	8.80×10^{-2}
#maga	2.25×10^{-3}
#trump	1.00×10^{-6}
#stopthesteal	7.12×10^{-2}
#qanon	5.40×10^{-2}

Anti-Trump Trends	
Trend	μ
#biden	6.41×10^{-2}
#biden2020	9.75×10^{-3}
#blm	1.19×10^{-2}
#impeach	1.57×10^{-2}
#impeachtrump	7.12×10^{-3}

Table 4.4: Estimated background rates $\boldsymbol{\mu}_{\mathrm{AT}}$ for the Anti-Trump process (on the left) and $\boldsymbol{\mu}_{\mathrm{PT}}$ for the Anti-Trump process (on the right)

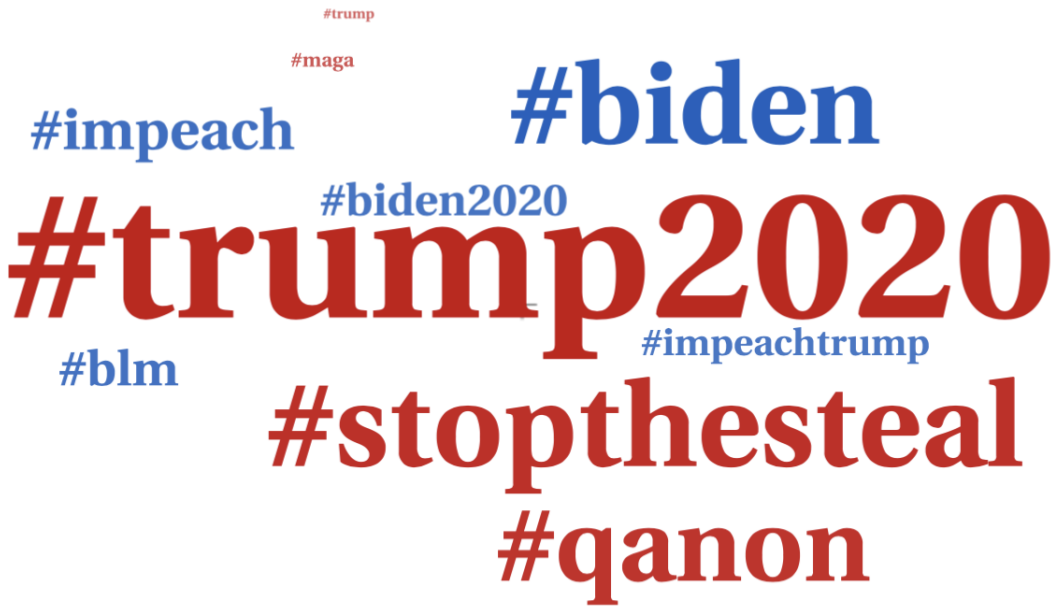

Figure 4.5: Word clouds representation of the hashtag trends for the Pro-Trump (in red) and Anti-Trump (in blue), each hashtag is sized based on the estimated background rate of the associated process

As we can see from the numerical results the background rates associated with Pro-Trump trends express in general higher magnitudes except for the #trump hashtag, that has a negligible value. In particular, Pro-Trump trends like #trump2020 and #stopthesteal reveal a higher rate than the Anti-Trump counterpart #biden2020

and #impeachtrump. Following the mathematical interpretation provided in Section 2.2, a higher level of the background rate means that there is a higher probability to observe an event that comes uninfluenced by any other event in the process. In the context of the social network analysis the higher rate related to the Pro-Trump trends could mean that this processes tend to express a higher risk of being 'pumped' in the network perhaps by some sort of external and malicious agents with the aim of disseminating targeted information in the network.

This can also be a consequence of the above-mentioned fact that Parler saw a huge influx of Trump supporters in the late months of 2020, just before its removal from the servers, that made use of the platform to spread their dissent and to foment a violent rebellion towards the results of the Presidential Elections.

4.3.2 Goodness of fit

Once the model parameter $\mathbf{A}$ and $\boldsymbol{\mu}$ are estimated we can proceed to perform the statistical tests introduced in Section 3.5 to assess the goodness of our assumption that the extracted hashtag cascades are indeed a realization of a Hawkes process. We treat each streamlines of the analyzed trends as a mono dimensional Hawkes process, and we use the estimated parameters $\mathbf{A}$ and $\boldsymbol{\mu}$ to come up with a closed form expression of the compensator function $\Lambda(t)$. We then transform each streamline using the corresponding compensator to obtain the sequence of transformed times needed to perform the tests.

Following the dissertation presented in Section 3.5.1 we display the Q-Q plots of the two categories of trends in Figure 4.6 (Pro-Trump trends) and in Figure 4.7 (Anti-Trump) obtained by performing the time transformation on each streamline of hashtag cascade.

As we can see, for most of the data, the tests give a quite good response as the theoretical quantile (evaluated from a unitary rate exponential density function) line up with the quantiles evaluated from the dataset.

Some exceptions are noteworthy: #trump and #stopthesteal cascade, concerning the Pro-Trump trends and #blm cascade, regarding the Anti-Trump trends. For these streamlines we can see how the theoretical and the sample quantiles fail to line up over the identity line $y = x$. This is considerably noticeable for the #trump hashtag.

These discrepancies are also recognizable in the result of the approximated Kolmogorov-Smirnnov-type tests, shown in Figures 4.8 and 4.9. It is immediate to note that the empirical CDFs of the above-mentioned trends fail to fall within the 95%-confidence lines, while for the other CDFs we can state (with a 95%-confidence level) that they are indeed a realization of a unitary Poisson process.

This is indeed confirmed by the application of the modified algorithm proposed by [Laub, 2014] which leads to the acceptance of the unitary rate Poisson process

assumption for all the trends except for the aforementioned ones.

The last tests performed are the ones to check the independence of the interarrival transformed times. The autocorrelation functions are shown in Figures 4.10 and 4.11. Again we can see, most of the trends express autocorrelation values that tend to fall into the confidence region as the time lag exceed the value 1 except for #trump and #blm hashtags.

These test results imply that not all the trends are indeed well modeled by a Hawkes process. A first explanation could be the found by looking at the history of Parler. As already reported in Section 1.2, the users' growth on the platform wasn't characterized by a constant increase, but instead it registered separated bursts of new accounts registrations. These increase of the network population inevitably led to a burst of activity and propagation of some particular centre-of-attention trends like #trump, #blm (Black Lives Matter) and #stopthesteal that possibly failed to be well captured by the model.

Further studies should be oriented to inspect some more suitable model to better capture this behavior. Another limitation of the present model, that can be of interest for future works, is that the entire model is based on a priori knowledge of parameter ω that tunes the rapidity of the decay of the exponential kernel. Further developments can be oriented to the construction of an extended algorithm that also estimates ω as a model parameter.

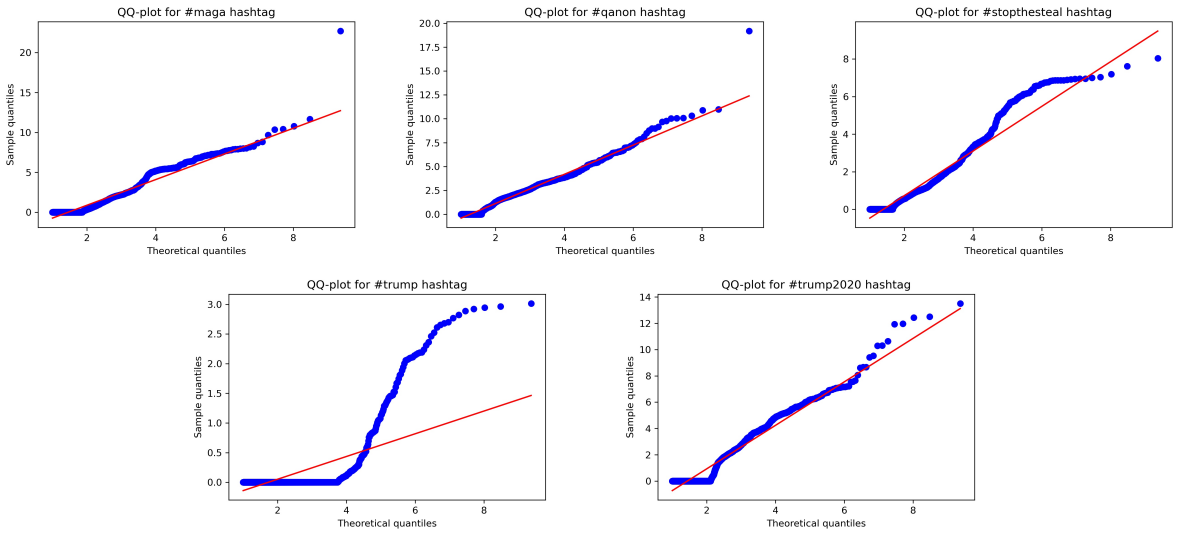

Figure 4.6: Q-Q plots for the Pro-Trump trends

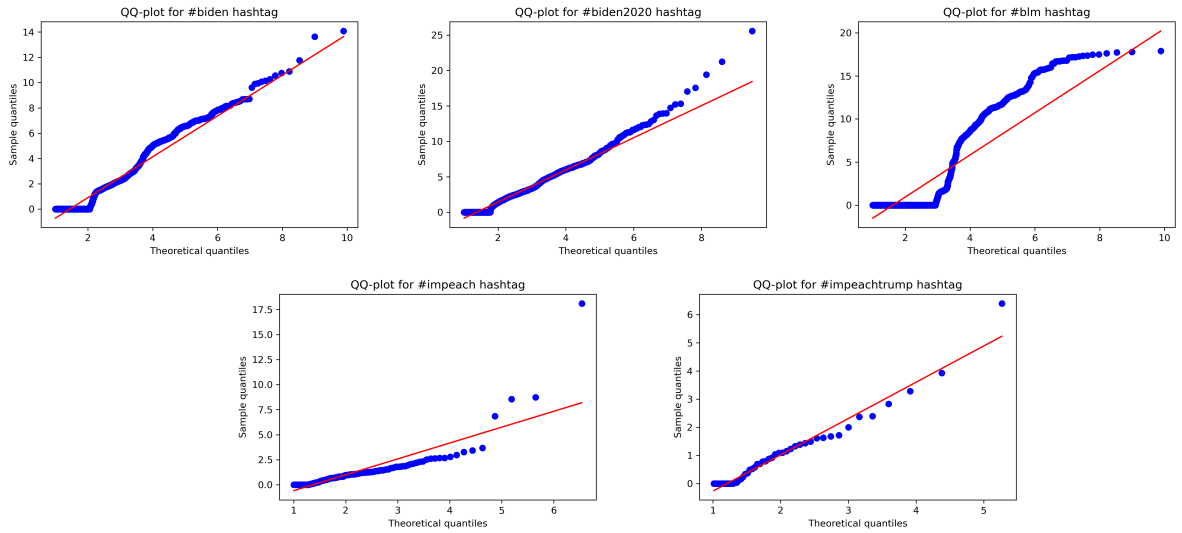

Figure 4.7: Q-Q plots for the Anti-Trump trends

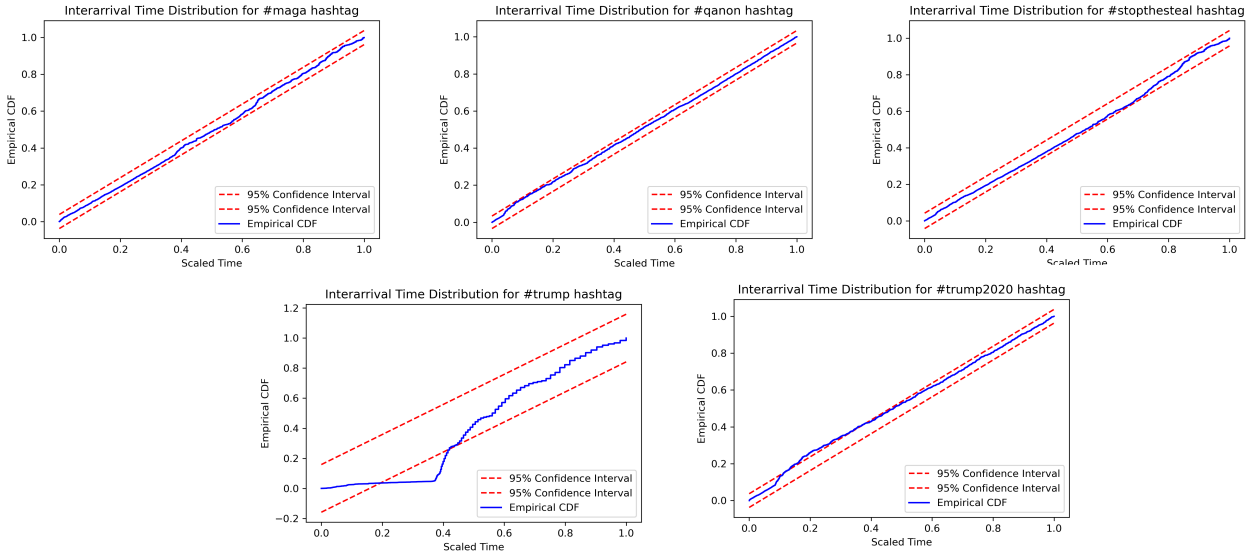

Figure 4.8: Approximated Kolmogorov-Smirnov type test for the Pro-Trump trends

4.4 Sentiment analysis

The last part of this work is dedicated to the analysis of the toxicity of the published posts on Parler to unveil the correlation between the most high-profile trends and their level of hate and maliciousness.

To conduct this analysis we use a Python library called `Detoxify` designed

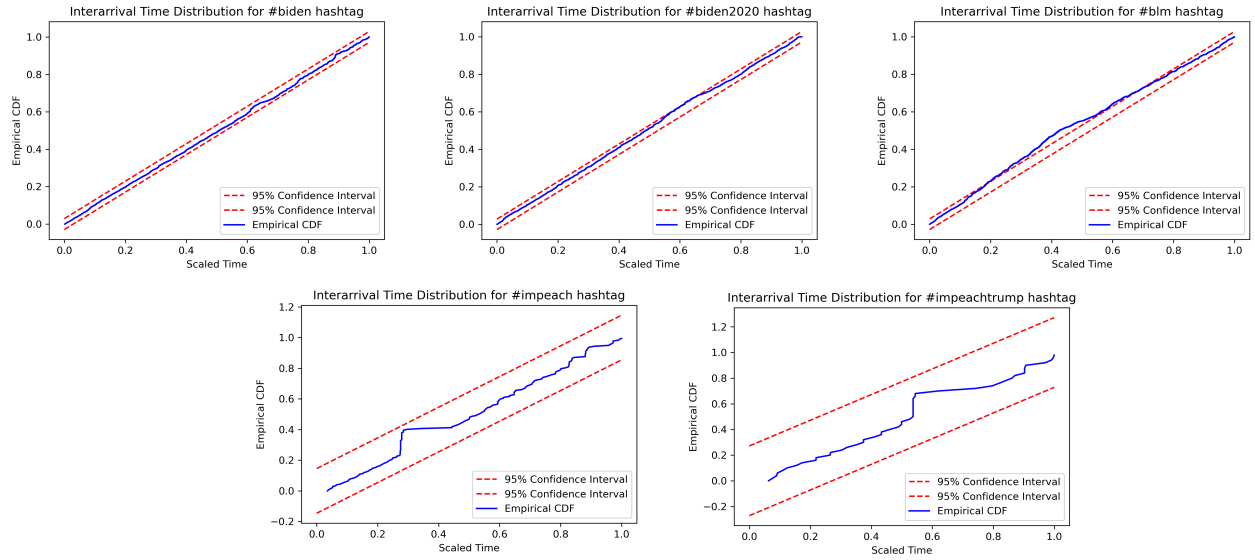

Figure 4.9: Approximated Kolmogorov-Smirnov type test for the Anti-Trump trends

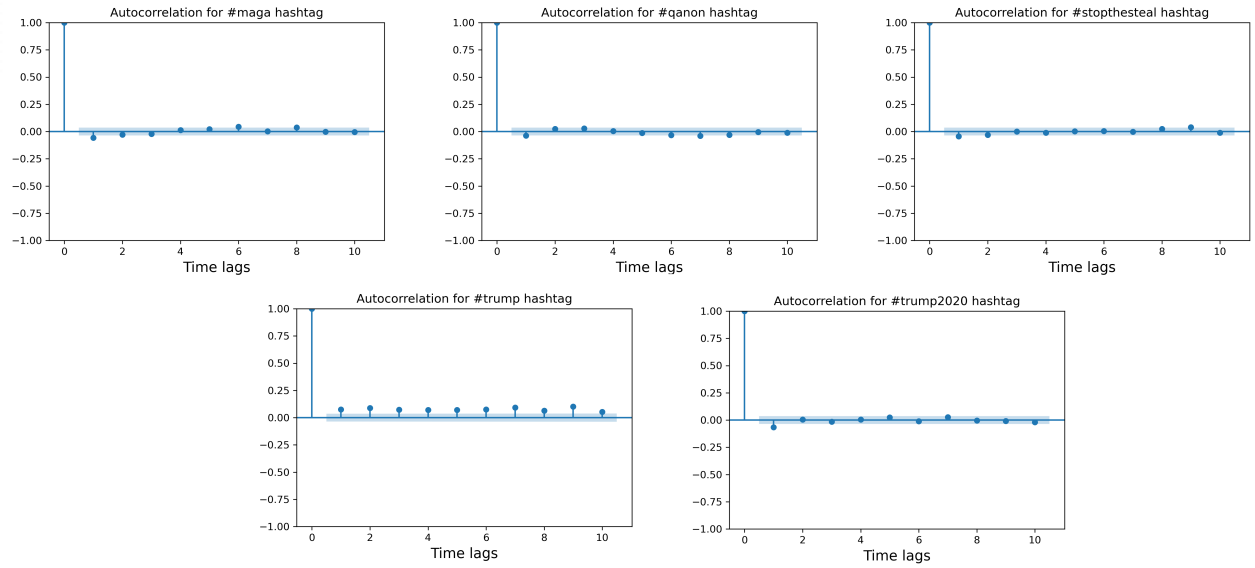

Figure 4.10: Autocorrelation functions for the Pro-Trump trends

by [Hanu and Unitary team, 2020] to easily predict if a comment contains toxic and obscene language. This library offers a Machine Learning-based method to evaluate a score (between zero and one) associated to the level of toxicity in a given input text distinguishing different categories of toxicity such as 'toxicity',

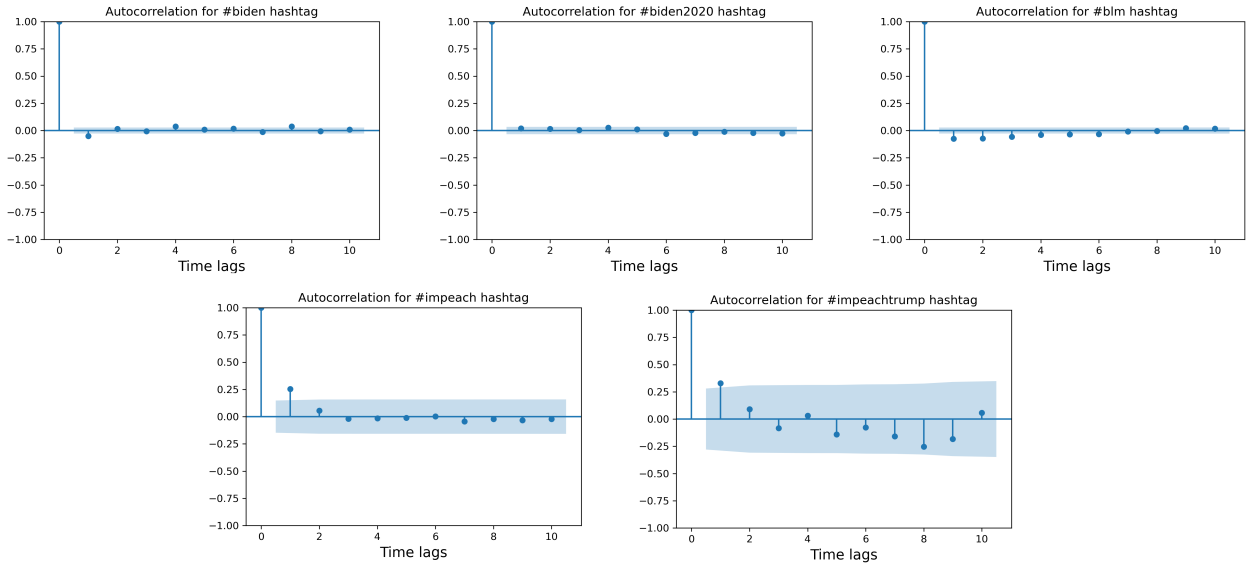

Figure 4.11: Autocorrelation functions for the Anti-Trump trends

'severe toxicity', 'obscene', 'identity attack', 'insult', 'threat' and 'sexual explicit'.

The model was trained to recognize different languages (English, Italian, French, Russian, Portuguese, Spanish and Turkish).

To proceed with the analysis of our dataset we read and extract a data frame containing the 'body' entry of a sample of posts (see Figure 4.2) filtered using some of the most high-profile trends in our dataset, namely "trump", "maga", "biden" and "blm". We first perform some simple text cleaning using Regular Expressions (Python library `re`) to make the extraction more efficient. For instance, we transform all the comments to lowercase, we drop all the hashtag and mention symbols (#, @) in front of words and replace all the occurrences of some phrases (like "make america great again" and "black lives matter") with the corresponding acronym ("maga" and "blm"). We then evaluate a mean score for each category of toxicity and for each selected trend. Due to computational cost of the algorithm we work with a smaller sample of the dataset. Results are shown in the bar plots in Figure 4.12. As we can see the resulting mean score for the Anti-Trump trend is considerably bigger than the one associated to the Pro-Trump trends for all the toxicity categories analyzed. This means that there is a high probability of observing a comment that contains a toxic content associated to an Anti-Trump trend. To give an example the comment in which the word 'Biden' appears, are associated with a larger score in the toxicity field marked as 'insult'. This confirms the strong bias in the ideologies of Parler users towards a Republican mindset with a peculiar inclination to denigrate different political leanings.

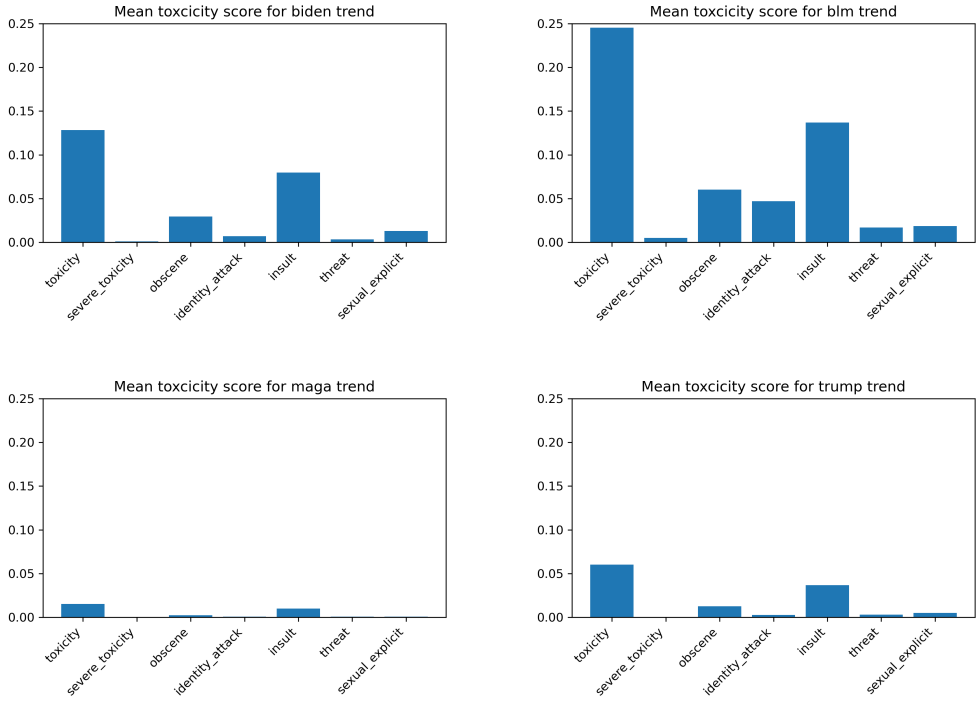

Figure 4.12: Bar plots of the mean toxicity scores of the selected trends

Chapter 5

Conclusions

This work was mainly oriented to provide an analysis and characterization of the Parler marginal social network in the time period surrounding Donald Trump's first impeachment. We first presented the mathematical framework of Hawkes processes, which have been shown to model well the self-excitation character of some real systems. As demonstrated in the literature, this self-excitation characteristic is very suitable for modeling events on social networks. We presented an adaptation of the model to infer some statistical analysis on a set of Parler's posts to show how the cascades of hashtags related to selected trends are indeed well modeled by a Hawkes process. Hawkes' framework allowed us to estimate the mutual influence between trends and the rates through which they are "inflated" in the network. The goodness of our hypothesis was evaluated using some basic statistical tests provided in the literature reviewed. The results showed that the population of Parler users was minimally affected by Anti-Trump trends, even though their rate of appearance in the network is not much different from that of Anti-Trump trends. This suggests a tendency for Parler users to partially ignore this information. In contrast, pro-Trump trends appear to have a greater degree of influence on the community and are also associated with slightly higher values of background rates. This bias in the ideology of Parler users is also confirmed by the sentiment analysis performed on the text comments of the published posts. Comments aimed at the anti-Trump movement are those that receive a higher level of toxicity, hatred, and harassing content.

Bibliography

Max Aliapoulios, Emmi Bevensee, Jeremy Blackburn, Barry Bradlyn, Emiliano De Cristofaro, Gianluca Stringhini, and Savvas Zannettou. A Large Open Dataset from the Parler Social Network, January 2021. URL https://doi.org/10.5281/zenodo.4442460.

S Azizpour, K. Giesecke, and G. Schwenkler. Exploring the sources of default clustering. *Journal of Financial Economics*, 129(1):154–183, 2018. ISSN 0304-405X. doi: https://doi.org/10.1016/j.jfineco.2018.04.008. URL https://www.sciencedirect.com/science/article/pii/S0304405X1830103X.

Emmanuel Bacry. Tick: a python library for statistical learning, with a particular emphasis on time-dependent modeling. https://x-datainitiative.github.io/tick/, accessed 2023-02-15.

Michael Bernstein, Andrés Monroy-Hernández, David Harry, Paul André, Katrina Panovich, and Guillermo Vargas. 4chan and /b/: An analysis of anonymity and ephemerality in a large online community. *Proceedings of the International AAAI Conference on Web and Social Media*, 5(1):50–57, June 2021. doi: 10.1609/icwsm.v5i1.14133. URL https://ojs.aaai.org/index.php/ICWSM/article/view/14133.

Peter J. Brockwell and Richard A. Davis. *Introduction to Time Series and Forecasting*. Springer Texts in Statistics. Springer Cham, 3 edition, 2016. ISBN 978-3-319-29852-8. doi: 10.1007/978-3-319-29854-2. URL https://doi.org/10.1007/978-3-319-29854-2.

Emery N Brown, Riccardo Barbieri, Valerio Ventura, Robert E Kass, and Loren M Frank. The time-rescaling theorem and its application to neural spike train data analysis. *Neural computation*, 14(2):325–346, 2002. doi: 10.1162/08997660252741149.

Emmanuel J. Candes and Benjamin Recht. Exact matrix completion via convex optimization, 2008. URL https://arxiv.org/abs/0805.4471.

Josh Constine. Parler jumps to no. 1 on app store after facebook and twitter bans. *TechCrunch*, January 2021. URL https://techcrunch.com/2021/01/09/parler-jumps-to-no-1-on-app-store-after-facebook-and-twitter-bans/.

D. J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes*. Probability and Its Applications. Springer, New York, NY, 2 edition, 2003. ISBN 978-0-387-95541-4. doi: 10.1007/b97277. URL https://doi.org/10.1007/b97277. Originally published in the series: Springer Series in Statistics.

D.J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes*. Probability and Its Applications. Springer, 2 edition, 2008. ISBN 978-0-387-21337-8. doi: 10.1007/978-0-387-49835-5. URL https://doi.org/10.1007/978-0-387-49835-5. Originally published in one volume in the series: Springer Series in Statistics.

Laura Hanu and Unitary team. Detoxify. Github. https://github.com/unitaryai/detoxify, 2020.

Alan G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 04 1971. ISSN 0006-3444. doi: 10.1093/biomet/58.1.83. URL https://doi.org/10.1093/biomet/58.1.83.

Gabriel Emile Hine, Jeremiah Onaolapo, Emiliano De Cristofaro, Nicolas Kourtellis, Ilias Leontiadis, Riginos Samaras, Gianluca Stringhini, and Jeremy Blackburn. A longitudinal measurement study of 4chan's politically incorrect forum and its effect on the web. *CoRR*, abs/1610.03452, 2016. URL http://arxiv.org/abs/1610.03452.

Selena Larson. Twitter is now hiding trolls by default. *CNN Money*, May 2018. URL https://money.cnn.com/2018/05/15/technology/twitter-push-down-trolls-spam/index.html.

Patrick Laub. Hawkes processes: Simulation, estimation, and validation. Master's thesis, 2014.

Scott W. Linderman and Ryan P. Adams. Scalable bayesian inference for excitatory point process networks, 2015. URL https://arxiv.org/abs/1507.03228.

Steven T Morse. Persistent cascades and the structure of influence in a communication network. Master's thesis, 2017.

Yosihiko Ogata. Seismicity analysis through point-process modeling: A review. *Pure and Applied Geophysics*, 155(2-4):471–507, 1999. doi: 10.1007/s000240050275.

Jakob Gulddahl Rasmussen. Lecture notes: Temporal point processes and the conditional intensity function, 2018. URL https://arxiv.org/abs/1806.00221.

Marian-Andrei Rizoiu, Young Lee, Swapnil Mishra, and Lexing Xie. A tutorial on hawkes processes for events in social media, 2017. URL https://arxiv.org/abs/1708.06401.

Candace Rondeaux and Ben Dalton. What role did the far-right platform parler play in the jan. 6 insurrection? *Slate*, Jan 2022. URL https://slate.com/technology/2022/01/parler-role-jan-6-insurrection.html.

Michael Rossetti and Tauhid Zaman. Bots, disinformation, and the first trump impeachment, 2022. URL https://arxiv.org/abs/2204.08915.

Nathan Srebro. Learning with matrix factorizations. Technical report, 11 2004. URL http://hdl.handle.net/1721.1/30507.

Howard M Taylor and Samuel Karlin. *An Introduction to Stochastic Modeling*. Academic Press, San Diego, CA, rev. ed. edition, 1994.

Wikipedia. QAnon - Wikipedia, 2021. URL https://en.wikipedia.org/wiki/QAnon#cite_note-9. [Accessed: 11-Mar-2023].

Wikipedia. Parler — Wikipedia, the free encyclopedia, 2023. URL https://en.wikipedia.org/wiki/Parler. [Online; accessed 7-March-2023].

David Wong and Luciano Floridi. Meta's oversight board: A review and critical assessment. *Minds & Machines*, 32(1):1–17, October 2022. doi: 10.1007/s11023-022-09613-x. URL https://doi.org/10.1007/s11023-022-09613-x.

Savvas Zannettou. Towards understanding the information ecosystem through the lens of multiple web communities, 2019. URL https://arxiv.org/abs/1911.10517.

Ke Zhou, Hongyuan Zha, and Le Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In Carlos M. Carvalho and Pradeep Ravikumar, editors, *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, volume 31 of *Proceedings of Machine Learning Research*, pages 641–649, Scottsdale, Arizona, USA, 29 Apr–01 May 2013. PMLR. URL https://proceedings.mlr.press/v31/zhou13a.html.