Hawkes Processes on Social and Mass Media:
A Causal Study of the #BlackLivesMatter Movement in the Summer of 2020

Alfred Lindström
Abstract

In this work we study interactions in social media and the reports in mass media during the Black Lives Matter (BLM) protests following the death of George Floyd. We implement open-source pipelines to process the data at scale and employ the self-exciting counting process known as Hawkes process to address our main question: is there a causal relation between interactions in social media and reports of street protests in mass media? Specifically, we use distributed label propagation to identify such interactions in Twitter, that supported the BLM movement, and compared the timing of these interaction to those of news reports of street protests mentioning George Floyd, via the Global Database of Events, Language, and Tone (GDELT) Project. The comparison was made through a Bivariate Hawkes process model for a formal hypothesis test of Granger-causality. We show that interactions in social media that supported the BLM movement, at the beginning of nationwide protests, caused the global mass media reports of street protests in solidarity with the movement. This suggests that BLM activists have harnessed social media to mobilise street protests across the planet. Lastly, we use Hawkes processes to model retweet cascades to examine the diffusion process in the Twitter data set, and introduce a heuristic measure of user influence.
Acknowledgement

Thanks to Combient Mix and Combient Competence Centre for Data Engineering Sciences, Professor of Sociology Simon Lindgren at Umeå University, Raaz, and mom and dad.
6 Modelling Retweet Cascades
   6.1 Fitting Marked Hawkes Processes ........................................ 28
       6.1.1 Results .............................................................. 29
   6.2 User Influence ............................................................ 30
   6.3 Summary ........................................................................... 31
   6.4 Related Work ..................................................................... 32

7 Conclusion ............................................................................... 34
1 Introduction

On May 25 2020, George Floyd, a 46 year old African-American man, is arrested in Minneapolis, Minnesota for allegedly using a counterfeit $20 bill to buy cigarettes. The arrest is caught on film by passersby, showing how police officer Derek Chauvin pins the handcuffed Floyd to the ground with his knee on Floyd’s neck, while his three colleagues prevent anyone from intervening. Floyd repeatedly utters the words “I can’t breathe“ before he goes unconscious. He later dies at the hospital, and the video of the arrest goes viral on Facebook [9]. The next day protests in support of the Black Lives Matter (BLM) movement, and against police brutality, start in Minneapolis, which during the following days will spread both nationally and internationally to over 60 countries, and become what may be the largest protests in U.S. history to date, with polls estimating attendances in the range of 15-26 million people [6].

In this work we study the landscape in mass and social media during the first month of protests that followed after the murder of George Floyd. Our primary question is whether there is a statistically significant interaction between communications in socially networked communities and street protests as measured by published reports in mass media. We attempt to answer this question by devising a data processing framework to mathematically model the interactions between social and mass media via the family of point processes known as Hawkes processes and conduct statistical hypothesis tests of Granger causality, subsequent to identifying influential social media communities using network models.

1.1 BLM-movement

BLM is a decentralised grassroots movement that began on social media, using the hashtag #BlackLivesMatter in the wake of the shooting of Trayvon Martin in July 2013. The movement has since then gained attention for demonstrations following the deaths of Michael Brown and Eric Garner in 2014, and George Floyd in 2020, with its main issues being that of advocating against police brutality toward African-Americans, and policy issues related to racial injustices [18].

As reactions and critiques of the BLM movement, the phrase “All lives matter” was coined, as well as the phrase “Blue lives matter”, after the shooting of two police officers during protests in Ferguson, Missouri in 2015. Both of these slogans are associated with conservative views, and reject the BLM-movement’s idea of a need to focus on the racial injustice towards African Americans.

The decentralised nature of all three of these movements, and the way social media has played a key part in their development, leading to real life events such as mass protests, motivates our choice to analyse data from social media and from mass media to try to get a better understanding of the mobilisation in social media into real-world action.

1.2 Twitter

Twitter is a micro-blog and social media service, founded in 2006, where users post and interact via tweets – a short message restricted to 280 characters, which may also contain pictures, short videos and URLs. Tweets can be original posts, replies to other tweets, or retweets, i.e., sharing of another user’s tweet. As long as a user does not actively choose to be private, anyone is able to read the tweets of the user. To help a tweet gain attraction, and make it easier for other users to find tweets on a specific topic, the user can tag their posts by including keywords prefaced with ‘#’, the hash symbol. These tagged keywords are called hashtags and they have been used by activists in global social movements such as #BlackLivesMatter and #MeToo [18].

Users may also follow other users on Twitter. The relationship of following is asymmetrical, meaning that if user A follows user B, user B does not have to follow user A. Compare this to Facebook, where users mutually have to accept each other as friends to be able to interact. To simplify things, if Facebook is about keeping in touch and networking with your friends, Twitter is about sharing and receiving information the user finds interesting; according to a study done in 2014, 44% of Twitter’s users have never tweeted which seems to suggest that a large part of the user base only uses Twitter for receiving information [24]. Due to this asymmetrical
following relationship, which encourages a more open discourse between users, along with its magnitude of users, choosing Twitter as the social media to analyse becomes the natural choice. Furthermore, unlike Twitter, other prominent social media platforms including Facebook and Instagram do not allow researchers open access to their data. We developed the library MEP to be able to design experiments, collect and analyse data from different Twitter APIs at scale in public cloud infrastructure.

1.3 GDELT

The Global Database of Events, Language, and Tone (GDELT) project, founded in 2013, is an open database supported by Google Jigsaw, that monitors news media in print, broadcast, and web formats from all over the world in over 100 languages. It is updated every fifteen minutes and stretches back to the 1st January of 1979, containing meta-data such as the people and organisations being mentioned, events and their locations, counts of key-words along with the tone and emotions of the parsed news sources. We used the GDELT database to get a high level understanding of the mass media landscape during the given time span, by reducing the records of reported events of protests, to data points in time.

1.4 Modelling & Inference

Many different approaches on how to explore, model, and analyse the data were taken, including network analysis especially during the data exploratory stages of this work. However, using the simple realisation that what unifies the two data sets of Twitter data and news reports from GDELT, is that every datum is given a position in time, the natural choice was to model with an appropriate stochastic process over time, for purposes of estimation and statistical hypothesis testing.

To model possible interaction between events in social media and reports in mass media, we implemented different models of Hawkes processes. Hawkes processes are a class of point processes characterised by their self-exciting nature, meaning that when an event occurs, the probability of a new event happening in the near future may increase. We use Granger causality by posing a hypothesis test using models of Hawkes processes. We will formally introduce this class of point processes in the next section.

\[1\text{https://www.gdeltproject.org/}\]
2 Hawkes Processes

In this section we will introduce the background theory and definitions for Hawkes processes that are needed for the modelling done in the sequel sections. These processes were introduced by Hawkes [17], and due to their self-exciting nature have been used in fields such as epidemiology, seismology, and finance [7, 3]. Moreover, they have been implemented in analysis of diffusion processes in social media [23, 32]. The general theory of point processes and Hawkes processes presented here is based on Daley’s An Introduction to the Theory of Point Processes: Volume I [7].

2.1 Point Processes on the Half Real Line

We start by defining general point processes on the positive real line.

**Definition 2.1.** Let $A$ be any Borel subset of the positive real line, and $\{t_i\}$ with $i \in \mathbb{N}$, and $t_i \geq 0$ be the set of time epochs. We let $N(A) = \text{number of indices } i \text{ for which } t_i \text{ lies in } A$, and assume that $N(A)$ is finite for bounded sets $A$. We then define $N(A)$ as a point process on the positive real line. For brevity, let

$$N(t) = \begin{cases} N((0, t]) & t > 0 \\ 0 & t = 0. \end{cases}$$

(2)

The natural interpretation of a point process $N(t)$ is that it counts each index $i$ in the set $\{t_i\}$ with $t_i \leq t$, as the occurrence of some event.

If $A$ is a union of disjoint sets $A_1, .., A_r$, i.e.,

$$A = \bigcup_{i=1}^{r} A_i \text{ where } A_i \cap A_j = \emptyset \text{ for } i \neq j,$$

it follows from (1) that

$$N\left(\bigcup_{i=1}^{r} A_i\right) = \sum_{i=1}^{r} N(A_i).$$

(3)

Moreover, we have that $N(A)$ is non-negative. If we extend and allow $r = \infty$, we have that $N(\cdot)$ is a counting measure on the $\sigma$-field $\mathcal{B}_{\mathbb{R}^+}$ of all Borel subsets on the positive real line $\mathbb{R}^+$.

To solidify the relationship between the set of time-epochs $\{t_i\}$ and our point process $N(t)$, while also defining each time-epoch in a more natural way, we let

$$t_i = \inf\{t > 0 : N(t) \geq i\} \quad i = 1, 2, \ldots.$$

(4)

From this, it is now more natural to talk about timestamps, where for each index $i$, $t_i$ is the exact time where some sort of event occurs for the the $i$:th time. Henceforth, we will denote the history of a point process up to time $t$, as the set $\mathcal{H}_t$ containing all timestamps $\{t_i\}$ up to time $t$. We now propose without proof that there exists a probability space where $N(A), N(t)$, and $t_i$ are well-defined random variables. For further reading on the existence of this probability space, we refer to Chapter 9 in Daley [8].

One important concept for point processes is that of stationarity.

**Definition 2.2.** A point process is stationary if for all $r = 1, 2, \ldots$ and all Borel subsets $A_1, \ldots, A_r$ of the positive real line, the joint distribution

$$\{N(A_1 + t), \ldots, N(A_r + t)\},$$

does not depend on $0 < t < \infty.$
Stationarity of a point process means that the distribution of events in an interval only depends on the interval itself, and not its location in time.

**Definition 2.3.** Let $N$ be a point process with history $\mathcal{H}_t$ up to time $t$. The *conditional intensity* of $N$ is then defined as

$$
\lambda(t) = \lim_{h \to 0} \mathbb{E} \left[ \frac{N(t+h) - N(t)}{h} \middle| \mathcal{H}_t \right].
$$

The (conditional) intensity $\lambda(t)$ of a point process is interpreted as the rate at which new events occur at time $t$.

### 2.2 Hawkes Processes

We will now introduce Hawkes processes.

**Definition 2.4.** Let $N$ be a point process with history $\mathcal{H}_t$. If the intensity $\lambda(t)$ of $N$ is of the form

$$
\lambda(t) = \mu + \sum_{t_i \in \mathcal{H}_t} \phi(t - t_i),
$$

we define $N$ as a *Hawkes process*, where $\mu$ is the *baseline intensity* and $\phi(t)$ is the *kernel*.

The events (i.e., points on the positive real line) of a Hawkes process, can be interpreted as being of two types. First we have the *immigrants* which arrive at a constant rate of the baseline intensity $\mu$. Next, we have the *offspring* which are produced by existing events. These arrive after time $t$ via the intensity of the kernel $\phi$, from any historical event $t_i \in \mathcal{H}_t$, which is often chosen to be monotonically decreasing, and is thus a descendant of an already existing event in history. Note that all events in history, whether they are immigrants or offspring, may produce new offspring.

As the arrival of an event increases the rate of new events arriving close in time, intuitively we can talk about Hawkes processes having a self-exciting nature; events will naturally cluster around immigrant events (see Figure 1). For a concrete example, one can think of the immigrant events as an earthquake occurring, with the offspring being after-shocks.

![Figure 1: Realisation of a Hawkes process with an exponential kernel. Each black dot represent the occurrence of an event.](image-url)
For a Hawkes process to be stationary, we require some constraints on the kernel.

**Definition 2.5.** Let \( \phi(t) \) be a kernel for a Hawkes process \( N \). We define
\[
\nu = \int_0^\infty \phi(t) dt ,
\]
as the *branching factor* of the Hawkes process.

The branching factor tells us, the mean number of offspring events one event can have. If \( \nu < 1 \), the process is in the *subcritical region*, and the branching from one event will die out. If \( \nu > 1 \), it is in the *supercritical region* and will explode exponentially. Moreover, if \( \nu < 1 \) then inductively we get the estimate, via the geometric sum, that an event will generate \( 1/(1-\nu) \) offspring in total on average.

One instructive example of how to interpret the branching factor comes from Filimonov and Sornette [12], where they examine trading by looking at financial data. Their estimation of the branching factor \( \nu \in (0.7, 0.8) \), means that 70%-80% of all the trades in the given data are due to past trades, rather than external events happening.

**Theorem 2.1.** A Hawkes process is stationary if and only if its branching factor satisfies \( \nu < 1 \).

We will now introduce a particular choice of kernel.

**Definition 2.6.** We define
\[
\phi^{(e)}(t) = \alpha \beta e^{-\beta t} ,
\]
as an *exponential kernel* where parameter \( \alpha \geq 0 \) is the *self-excitation parameter*, and parameter \( \beta > 0 \) is the *decay rate*.

Parameter \( \alpha \) thus decides how much an occurred event will influence the rate of new events, while \( \beta \) will decide how long into the future this influence will last as \( \phi^{(e)}(t) \to 0 \), when \( t \to \infty \).

### 2.2.1 Marked Hawkes Point Processes

We now introduce an extension of Hawkes processes, where each timestamp of an event not only contains the event’s location in time, but also some information about that specific event. These types of point processes are known as marked point processes.

**Definition 2.7.** Let \( N^* \) be a point process on \( \mathbb{R}^+ \). We define \( N^* \) as the *ground process*. Let the mark space \( \mathcal{K} \) be a set of *marks*. We then define the point process \( \{t_i, \kappa_i\} \) on \( \mathbb{R}^+ \times \mathcal{K} \) as a *marked point process*, if \( N^* \) is a well-defined point process, i.e., for bounded \( A \in \mathbb{R}^+ \), \( N_g(A) = N(A \times \mathcal{K}) < \infty \).

An example of a marked point process would be a model where we in one dimension have the timestamps of earthquakes occurring, and in the mark space we have the information of the magnitude of each earthquake. The size of magnitude could then influence the intensity of new earthquakes occurring in time.

In Section 6, we will use a marked Hawkes process with a power-law kernel defined as below.

**Definition 2.8.** Let \( N \) be marked point process with history \( \mathcal{H}_{t,m} \) with elements on the form \( (t_i, m_i) \in \mathbb{R}^+ \times \mathbb{N} \). If the intensity of \( N \) is of the form
\[
\lambda(t) = \sum_{(t_i, m_i) \in \mathcal{H}_{t,m}} \phi_{m_i}(t-t_i) = \sum_{(t_i, m_i) \in \mathcal{H}_{t,m}} km_i^\beta (t-t_i+c)^{-(1+\theta)} ,
\]
where \( \kappa, \beta, \theta, c > 0 \), we define \( N \) as a *marked Hawkes process with a mark-proportioned power-law kernel*.

Analogous to the exponential kernel, \( 1 + \theta \) captures how quickly an event is forgotten, while parameter \( c \) shifts the term so that \( \phi_{m_i}(t) \) is bounded when \( t \approx 0 \). Parameters \( \kappa \) and \( \beta \) determine how much an event influences the intensity of future events. Note that that this influence is affected by the mark \( m_i \in \mathbb{N} \), i.e., the larger \( m_i \) is, the larger the influence of that event will be.
2.2.2 Multivariate Hawkes Processes

It is natural to model Hawkes processes in multiple dimensions, where each dimension is a Hawkes process for each type of event, that are possibly interconnected. To continue with the example of earthquakes, let the first dimension in our multivariate Hawkes process count the number of earthquakes in a region, and the second dimension count the number of tsunamis in the same region. The arrival of an earthquake might then give rise to a tsunami, which a multivariate Hawkes process, given the right parameters, can capture.

**Definition 2.9.** We define a marked point process \( N \) with mark space \( \{1, \ldots, d\} \) as multivariate point process. Moreover, for \( i \in \{1, \ldots, d\} \) we denote the \( i \)-th component of the multivariate point process \( N \) as \( N_i \), where \( N = (N_1, \ldots, N_d) \) is a marked point process, and \( N_i \) is a well-defined ground process. We refer to \( d \) as the dimension of \( N \).

**Definition 2.10.** Let \( d \in \mathbb{N} \) be the number of dimensions, and \( \mathcal{H}_{t,i} \) for \( i = 1, \ldots, d \) be the history of events in dimension \( i \). The multivariate point-process induced by the intensities

\[
\lambda_i(t) = \mu_i + \sum_{j=1}^{d} \sum_{t_k \in \mathcal{H}_{t,j}} \phi_{ij}(t - t_k) \quad i = 1, \ldots, d
\]

is then defined as a multivariate Hawkes Process.

![Figure 2](image_url)

**Figure 2:** Realisation of a bivariate Hawkes process \((d = 2)\), with a multivariate exponential kernel. The black dots represent events occurring in one of the two dimensions.

**Definition 2.11.** Let \( d \in \mathbb{N} \) be the number of dimensions. We define the multivariate kernel

\[
\phi_{ij}^{(e)}(t) = \alpha_{ij} \beta_{ij} e^{-\beta_{ij} t} \quad i, j = 1, \ldots, d
\]

(11)
as the multivariate exponential kernel, where $\alpha_{ij} \geq 0$ is the excitation parameter, and $\beta_{ij} > 0$ is the decay rate for which events in dimension $i$ influences the arrival of new events in dimension $j$.

The excitation parameter $\alpha_{ij}$ can be interpreted similarly as $\alpha$ in the one-dimensional case with the exponential kernel, with the exception that this influence on new events in dimension $j$ now may come from previous events in any dimension $i \in \{1, \ldots, d\}$. Analogously, $\beta_{ij}$ is interpreted as the rate of decay that specifies how past events in dimension $j$ can influence the arrival of new events in dimension $i$.

### 2.3 Granger Causality

How to rigorously define causality has been a topic of discussion in western philosophy for over 2000 years, starting with Plato and Aristotle [11], and continuing on with Hume and Kant’s disagreement being one of the fundamental discussions in modern philosophy. The problem is still open. [26].

In light of this, and in some sense to get around the metaphysical complications of proper causality, Clive Granger introduced the concept of Granger Causality relating to stochastic processes. The basic idea is if a variable $X_t$ Granger-causes variable $Y_t$, then the past values of $X_t$ contains information that helps predict future value $Y_{t+1}$ better than just doing prediction on past values of $Y_t$ [16].

Using the following Theorem from Eichler [10], we will test the null hypothesis of the non-existence of Granger causality between events in social and mass media, and vice versa, in the sequel.

**Theorem 2.2.** Let $N(t)$ be a multivariate Hawkes process in $d$ dimensions, with kernels $\phi_{ij}(t)$, $i, j \in \{1, \ldots, d\}$. Then the $j$-th component $N_j$ does not Granger-cause the $i$-th component $N_i$ if and only if $\phi_{ij}(t) = 0$, $\forall t \in \mathbb{R}$.

Thus, when $N(t)$ is a multivariate Hawkes process with an exponential kernel, by Theorem 2.2 the $j$-th component $N_j$ does not Granger-cause the $i$-th component $N_i$ if and only if $\alpha_{ij} = 0$, $\forall t \in \mathbb{R}$.

When examining Granger causality on more than two dimensions, it is natural to look at the following induced graph.

**Definition 2.12.** Let $N$ be a multivariate Hawkes process in $d$ dimensions. We define the Granger causality graph $G_c$ with vertices $V = \{1, \ldots, d\}$, directed edges $(u, v) \in E$ from $u$ to $v$, and the following constraint

$$(i, j) \notin E \iff \phi_{ji}(t) = 0, \forall t, \text{ and } i, j \in V.$$

Via the Granger causality graph, one can naturally talk about indirect Granger causality; assume that there is no edge from vertices $i$ to $j$, i.e., the $i$-th component does not Granger-cause the $j$:th component. The $i$:th component may however affect the $j$:th indirectly, if there exists a path from $i$ to $j$ in the Granger causality graph.
3 Data Handling

In this section we will go through how the Twitter and GDELT data was processed for the analysis in the next section.

3.1 Apache SPARK

The data was handled using Apache Spark\(^2\) which is an open-source engine designed for data engineering, data science, and machine learning on clusters of multiple computers, by implicit data parallelism. Spark is multi-language and supports Scala, Python, R, SQL, Java, C# and F#. While most of the code for this thesis was written in Scala, the ease of switching between languages in the same environment proved quite useful, as we would use libraries written in both R and Python.

On top of Spark core, Spark SQL [2], which introduces the data abstraction of DataFrames, allows manipulation in Scala, Python, and R using the standard SQL language, and the graph-processing framework GraphX [15], allows for network-analysis. To run Spark, the cloud data platform Databricks was used, which provided cloud storage, computing clusters, and a notebook-environment to write and run the code after loading the two main libraries developed for this study, MEP\(^3\) and SPARK-GDELT\(^4\).

3.2 Twitter

3.2.1 Application Programming Interface

To work with and be able to analyse Twitter data efficiently on an arbitrarily large scale, access to Twitter’s Application Programming Interface (API) is needed, and requires Twitter developer credentials, which anyone can apply for. With access to the credentials, one may request and download tweets which can be represented as JSON-files. At the time of writing, two versions of the Twitter API exists. This work was done in the older version 1.

To get a sense of how the data was handled, a brief overview of the relevant fields from the schema of the JSON for a tweet will be presented. For full details, we refer to Twitter’s data dictionary\(^5\)\(^6\). The two most basic objects for a tweet are the User object and the Tweet object shown in Tables 1 and 2, respectively.

Table 1: Some attributes, with their types and description, for the User object.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Int64</td>
<td>The unique integer representation of the user.</td>
</tr>
<tr>
<td>screen</td>
<td>String</td>
<td>The screen name, also known as handle of the user.</td>
</tr>
<tr>
<td>followers</td>
<td>Int</td>
<td>The number of followers the user has.</td>
</tr>
<tr>
<td>friends</td>
<td>Int</td>
<td>The number of users the user follows.</td>
</tr>
</tbody>
</table>

From the User object, as the name suggests, we get access to the metadata of a user. However, note that no direct information about which users follow the user, or which users the user follows, beyond the counts, is accessible from the user object.

---

\(^2\)https://github.com/apache/spark

\(^3\)https://github.com/lamastex/mep

\(^4\)https://github.com/lamastex/spark-gdelt


Table 2: Some attributes, with their types and description, for the Tweet object.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>created_at</td>
<td>String</td>
<td>UTC-time when the tweet was created.</td>
</tr>
<tr>
<td>id</td>
<td>Int64</td>
<td>The unique integer representation of the tweet.</td>
</tr>
<tr>
<td>text</td>
<td>String</td>
<td>The textual content of the tweet.</td>
</tr>
<tr>
<td>in_reply_to_status_id</td>
<td>Int64</td>
<td>If the tweet is a reply to another tweet, the field will contain the tweet-ID of that tweet. Otherwise null.</td>
</tr>
<tr>
<td>in_reply_to_user_id</td>
<td>Int64</td>
<td>If the tweet is a reply to another tweet, the field will contain the user-ID of that tweet. Otherwise null.</td>
</tr>
<tr>
<td>user</td>
<td>User Object</td>
<td>All information of the user of the tweet.</td>
</tr>
<tr>
<td>quoted_status</td>
<td>Tweet Object</td>
<td>If the tweet is a quote tweet, all information of the original tweet will be contained in this field. Otherwise null</td>
</tr>
<tr>
<td>retweeted_status</td>
<td>Tweet Object</td>
<td>If the tweet is a retweet, all information of the original tweet will be contained in this field. Otherwise null</td>
</tr>
</tbody>
</table>

From the Tweet object, we get access to the metadata of a tweet. Via the field “user”, we also get the information of the user behind the tweet, since this is a User object. Moreover, since the fields “quoted_status” and “retweeted_status” are Tweet objects, we get the full information of the original post that has been retweeted or quoted.

Note that the Tweet object in “retweeted_status” points to the original tweet that has been retweeted, if the post is a retweet. It is possible for a user to retweet another user’s retweet, but information on this chain of events is thus not accessible. For example, let user $A$ write a tweet $T$ that gets retweeted by user $B$. Later, user $C$ sees this retweet on user $B$’s timeline and then retweets $T$. Twitter’s API will then only tell us that user $B$ and $C$ have retweeted user $A$, but not the fact that user $C$ accessed this tweet via user $B$. This limitation will provide one of the questions that we will later try to address.

Along with these two objects, there is another object named entities, which contains all the metadata of a tweet’s content, including any URLs, hashtags, twitter handles of users mentioned, and media content (pictures and short video clips).
Figure 3: An overview of Twitter’s API. Source is simply which device (e.g., smartphone, and desktop) a user used to post the tweet. Note that included media in tweets are represented as links in this overview.

3.2.2 Data Set

The data set that was used [13] has 41.8 million collected tweets from 10.1 million unique users regarding the Black Lives Matter-movement, along with the smaller counter movements Blue Lives Matter (pro-police movement), and All Lives Matter. These tweets were collected by filtering on the keywords BlackLivesMatter, BlueLivesMatter and AllLivesMatter. The data contains tweets from the beginning of the movement in 2013 to 30 June 2020. In this work, the focus was on the events occurring during the aftermath of the death of George Floyd on 25 May 2020, thus all tweets before this date were discarded.

3.2.3 Collecting Data

Due to Twitter’s policy, collecting and sharing tweets publicly is not allowed. To share a set of tweets, instead one shares the IDs of each tweet, and to get the full metadata of the tweets, access to Twitter’s API is needed. There is also a limit on how many tweets one may collect per hour, which initially was a problem. To get around this, the python library twarc\(^7\) was used. twarc allowed us to collect tweets from the IDs (a process known as hydrating), in an optimised way with respect to the hourly collection limit.

To be able to work with the data in Databricks and Spark, a Docker-container with python and twarc was set up on a remote machine, that ran the hydration script on small batches of the IDs, collected them as `.json`-files, and then compressed and stored them in our Databricks cloud storage. This procedure took roughly five days. A consequence of retroactively collecting tweets from their IDs is that all tweets that have been removed due to various reasons (such as the users of these tweets getting banned, removing their accounts, or going private) at the time of hydrating, are not accessible and were therefore not collected.

\(^7\)https://twarc-project.readthedocs.io/en/latest/
After hydrating the IDs from the data set, and discarding tweets posted earlier than 24 May 2020, 23.3 million tweets from 7.1 million unique users were left. These were cleaned to be easier to work with using Spark’s Dataframes. We also categorised each tweet as an original tweet, retweet, quoted tweet, etc., and then stored them in the column-based data-storage format parquet on a delta lake [1]. See MEP for details of the collector, pre-processor and categoriser behind the delta lake.

3.3 GDELT

A brief overview of GDELT to appreciate how we handled the data for this work follows. For a more thorough overview, we refer to the documentation and SPARK-GDELT, our open-source library developed for this study, where we built an analytics-ready Delta Lake [1] to handle the data.

3.3.1 Coding

The idea behind GDELT is that of coding, which is fundamentally fairly simple. Given a record – for example a written news article – go through the text and identify the real world events that are being reported in the record, and identify the actors who are involved in the event. During the Cold war, two coding frameworks dominated: WEIS and the Conflict and Peace Data Bank, COPDAB. Both of these frameworks, being developed and used in a 20th century post-World War II context, were focused on codifying how sovereign states (the actors) interacted through official diplomacy and military threats [31]. For example, in the following sentence:

"President Reagan has threatened further action against the Soviet Union in an international television program beamed by satellite to more than 50 countries",

one would identify the act of threatening as the event, and assign it some integer (decided by the code framework), with the actors being President Reagan (or the United States if the coder is only interested in sovereign states), and the Soviet Union.

This process of coding would historically be done by hand. However, the combination of psychological studies showing that the kind of sustained decision-making involved in coding leads to fatigue, inattention, and heuristic shortcuts, and the technological advancement in computing software and hardware, coding is nowadays automated. The frameworks for codifying has also developed since the cold war, with GDELT using the framework of Conflict and Mediation Event Observations (CAMEO) [20]. Some notable changes being that actors are no longer limited to sovereign states, and include persons, organisations, and companies.

In practice, GDELT is essentially two separate but interlinked databases: The Global Knowledge Graph (GKG), which consists of records and the Event Database, which as the name suggests stores events that are being reported.

3.3.2 GKG

The Global Knowledge Graph (GKG) consists of all records from multiple news sources in the world. As of version 2 of GDELT, new records get added every fifteen minutes. Whenever a record is added, the source text is parsed via natural language processing to identify the events (using coding), locations, persons and organisations, as well as themes mentioned in the text. Moreover, keywords such as “protest“ that are mentioned multiple times get counted. Sentiment analysis is also incorporated to get a value of the tone of the source text (whether the text is positive, neutral or negative). Many other metadata extracts are in each GKG record.

http://data.gdeltproject.org/documentation/GDELT-Global_Knowledge_Graph_Codebook-V2.1.pdf
3.3.3 Event Database

The Event database attempts to record all unique events that are being identified in the parsing process of the GKG database. Each data point is given a unique ID for the event, and contains the date, the actors along with the code of the type of event being identified. The coded event also gets mapped to the Goldstein-scale [14], which seeks to measure the potential impact the event could have on the stability of the country. Moreover, the Event database has metadata on how often the event has been mentioned by records in GKG and the average tone of these records.

3.3.4 Handling of the GDELT Data

Due to the sheer magnitude of data contained in the GDELT database, working with data proved quite a challenge. Our goal was to filter out the events about the protests relating to the Black Lives Matter movement and the counter movements between 25 May 2020 and 30 June 2020. Although the parsing of news records into the GKG database identifies organisations, it did not identify the Black Lives Matter movement as one, probably due to its lack of centralisation.

What we did instead was to filter out all data relating to protests happening in the world. This naturally led to noisy data, since we got reports of protest unrelated to the BLM movement, but we justify this by the fact that no other major protests were happening in the world at the same time. To check this, we filtered the Event database by events with CAMEO root-code 14, i.e., those events coded as protests, over a three months timeline.

![Figure 4: Events coded as protests in the GDELT Event database.](image)

As we see in Figure 4, there is a baseline of roughly 5,000 events per day coded as protests before 25 May. This number then explodes, and there is nothing that suggests that the sudden increase in magnitude of protests are not related to the BLM protests. It is worth pointing out that there is no bijection between the real world protest and the protest data from the Event database. For example, if in one city during one day, large protests are taking place and one group of people are protesting peacefully while another group is rioting, then the coding framework should identify the act of the peaceful and rioting protesters as two different events [31], although they are near each other in time and space. Thus, saying that more than 8,000 protests happened on the 1 June 2020, would be incorrect.
In Section 5 we will look at news reports in mass media, and therefore use data from the GKG database. We did this by filtering by the themes of the records. All records in the GKG database with theme “PROTEST” were filtered out.

Figure 5: Comparison of records from the GKG database with theme “PROTEST”, and events coded as protests from the Event database.

Ignoring the periodic dips in the GKG plot in Figure 5 (which are due to less reporting being done on weekends), the two plots follow a similar pattern. Naturally, there are more records than events, since multiple news sources may report the same event.
4 Analysis of Twitter Data

In this Section, we explore the Twitter data, first via simple querying on the data set, and then by doing network analysis on the induced retweet network. The results from this exploratory data analysis then motivated the choice of using Hawkes processes to model and perform hypothesis tests to shed light on the phenomena of interest in this study – occurrence of tweets in support of the BLM movement and that of mass media reports of street protests.

4.1 Data Observations

4.1.1 Timeline

We started by examining the data over the relevant time-span from 24 May 2020 to 30 June 2020. During this period, 23,346,745 tweets by 7,111,140 unique users were collected using **twarc** on the BLM data set [13].

From Figures 6 and 7, we can see that activity first starts on Twitter, and the reports of protests start to drastically increase on 27 May. We also see a dip in Twitter activity between 31 May and 2 June, while the GDELT data on the number of reports of protests spikes during these days. The explanation of this is simply that the data set lacks tweets on these days. This was found while exploring the data, and noticing that the data set contained retweets of a tweet from this time period, but not the original tweet. Whether these missing tweets disappeared during the collecting of data, or if they are missing in the original data set [13] of the Tweet IDs, remains unclear. To deal with this, we refrained from doing any modelling with tweets from this time period.

4.1.2 Type & Media Content of Tweets

Next, we examined TweetTypes, i.e., the types of status update or interactions in our Twitter data. From Figure 8, the most to least frequent TweetTypes (% of data) were Retweets (55%), Retweets of Quoted Tweets (27%), Original Tweets (7%), Quoted Tweets (7%), Reply Tweets (3%), Reply of Quoted Tweets (1%). Thus, only 18% of the tweets in the BLM-data set were original tweets (either original, or replies to other tweets),...
with the remaining 82% being some sort of retweeted content. This suggests that the re-sharing of other users’ original content is fundamental for how users interact with each other on Twitter, and motivated our choice of examining the retweet network.

4.1.3 Content of Tweets

One initial idea was to focus on URLs to news articles shared by Twitter users, and then link them to the GDELT database. However, we soon discovered that users in general did not share news sources from mass media. Instead highly retweeted tweets often contained original media (i.e., videos and pictures), which were often taken from the protests. For instance, 53% of tweets with over 1000 retweets, as opposed to only 17% of all tweets shared original media.

While any conclusions on why this is, is out of the scope here, it is at least noteworthy that more than half of the tweets with over 1000 retweets, contained some sort of original media.

4.2 Network Analysis

From the last subsection we saw the importance of retweets in the Twitterverse. In this subsection we will formalise this by introducing a network structure on our data set.

4.2.1 Retweet Network

**Definition 4.1.** Let $G_I = (V, E)$ be a directed weighted graph in time interval $I \subset \mathbb{R}_+$, where every vertex $v \in V$ is a unique Twitter user, and every edge $e \subset \{(u, v) \mid (u, v) \in E \subset V^2\}$ is interpreted as user $v$ having retweeted $u$ during time interval $I$. The weight $W(e) = W((u, v)) \in \mathbb{N}$ is the number of times user $v$ has retweeted user $u$. We then define $G_I$ as a retweet network.

Furthermore, we define $G'_I$ as an undirected retweet network if $(u, v) \in E \Leftrightarrow (v, u) \in E$. Thus $G'_I$ ignores whether $u$ retweeted $v$ or vice versa but preserves the information that there is a retweet relation between the two users.
Figure 8: Types of tweets collected in the given time period.

Figure 9: Tweets sharing media

(a) All tweets

(b) Tweets with over 1000 retweets
We chose to look at retweets since a retweet by user \( u \) of an original tweet by user \( v \) is highly likely to mean that user \( u \) agrees with user \( v \). Direct retweets are generally recognized to indicate trust in the communicator and endorsement [19, 22, 5]. The number of times a user has been retweeted also gives a probabilistic interpretation, using the random geometric graph interpretation in [29], that measures how influential a user is on another in terms of the lengths of their most retweeted paths.

Unfortunately, with the given data and how Twitter’s API works, we cannot without using highly data intensive methods, get the network of how the users in the data-set follow each other; this would have opened up for some different methods and algorithms to analyse the data, such as PageRank e.g.

By looking at our retweet network we can already get some information from the Twitter data set; simply by summing the outgoing edges and their weights for every user, we get the most retweeted users in our time interval between 24 May 2020 and 31 June 2020 (Table 3).

Table 3: Ten most retweeted users, sorted by number of retweets. Usernames for non-public users have been anonymized. The communities were identified using the label propagation algorithm.

<table>
<thead>
<tr>
<th>Username</th>
<th>followers</th>
<th>retweets</th>
<th>Community</th>
</tr>
</thead>
<tbody>
<tr>
<td>@JoshuaPotash</td>
<td>142,833</td>
<td>759,572</td>
<td>Pro-BLM</td>
</tr>
<tr>
<td>@YourAnonCentral</td>
<td>5,862,927</td>
<td>529,431</td>
<td>Pro-BLM</td>
</tr>
<tr>
<td>-</td>
<td>1,584</td>
<td>187,065</td>
<td>Pro-BLM</td>
</tr>
<tr>
<td>@elijahdaniel</td>
<td>760,935</td>
<td>161,337</td>
<td>Pro-BLM</td>
</tr>
<tr>
<td>-</td>
<td>22,983</td>
<td>135,698</td>
<td>Pro-BLM</td>
</tr>
<tr>
<td>@MrAndyNgo</td>
<td>799,291</td>
<td>125,898</td>
<td>Anti-BLM</td>
</tr>
<tr>
<td>-</td>
<td>1,232</td>
<td>125,826</td>
<td>Pro-BLM</td>
</tr>
<tr>
<td>@BTS_twt</td>
<td>34,107,446</td>
<td>125,534</td>
<td>K-pop</td>
</tr>
<tr>
<td>@shawnasabi</td>
<td>140,788</td>
<td>106,731</td>
<td>Pro-BLM</td>
</tr>
<tr>
<td>@Drebar_</td>
<td>141,613</td>
<td>103,594</td>
<td>Pro-BLM</td>
</tr>
</tbody>
</table>

One noteworthy user is the sixth most retweeted user @MrAndyNgo. Andy Ngo is an American conservative journalist and a prominent opponent of the Black Lives Matter movement, who in February 2021 published *Unmasked: Inside Antifa’s Radical Plan to Destroy Democracy* [25], where he among other things writes about his experiences from the BLM-protests of 2020. His presence in the most retweeted users will serve as a gateway into the counter-movements of All Lives Matter and Blue Lives Matter. Thus, we need to detect different communities within the retweet network obtained from the dataset, such that each community has more edges or retweets within it when compared to the number of edges between it and another community.

4.2.2 Connected Components

The motivation behind the definition of an undirected retweet network follows in the next step, when we look at the connected components of our graph.

**Definition 4.2.** Let \( G \) be a graph. A sequence of edges \((e_1, \ldots, e_{n-1})\) is called a *path* if it corresponds to a sequence of distinct vertices \((v_1, \ldots, v_n)\), such that \( e_i = (v_i, v_{i+1}) \). Two vertices \( u, v \) are *connected* if there exists a path between them, and if \( G \) is undirected, we call the sub-graph \( H \) of \( G \) a *connected component* if and only if there exists a path between every pair of vertices in \( H \) which contains a subset of the vertices in \( G \).

The reasoning behind invoking the notion of connected components of the undirected retweet network is to, on a high level, make sure that a meaningful discourse between users, in terms of being influenced by and influencing others, exists within the connected component. In practice, we could have a very disconnected network with lots of unconnected components, which would mean that most users only interact and retweet a few selected users. Another interesting case would be if the network would have a few significantly large components; this would suggest the existence of a set of discourses, where the users in their respective component do not interact – perhaps because of political differences reflected in large “echo chambers”. To find all connected components in the retweet network, the GraphFrames framework in Spark was used. The result showed that 6,083,687...
i.e., 85.6% of the 7,111,140 users were in the same connected component. The remaining users were scattered around in smaller connected components, with the largest being 74 users. These users were therefore discarded from further analysis.

4.2.3 Community Detection

While the data set contains tweets using the hashtags of the counter movements #AllLivesMatter and #BlueLivesMatter, in practice, users associated with these movement did not necessarily use these hashtags, but often used the hashtag #BlackLivesMatter either ironically or to get more attention. Thus, just using simple querying on the hashtags in the data set, did not suffice to get a sample of users from these movements. To get a better sense of the relationship between users, we instead therefore used the community detection algorithm known as Label propagation algorithm (LPA). LPA is a semi-supervised machine learning algorithm, which seeks to assign labels to nodes in a network, where each label maps to a specific community inside the network [27]. In Spark’s GraphX framework, the algorithm is implemented using Pregel API [21], which allows for parallel computation when processing graphs. On a high level, Pregel computations are a sequence of iterations, defined as supersteps, where for every superstep, each vertex in the graph runs a user defined function. This local vertex-centric approach where each vertex is processed independently in parallel, in contrast to the more classical iterative graph algorithms where each vertex is visited one by one, naturally induces distributed implementations that can computationally scale to arbitrarily large networks. In distributed LPA, implemented as a Pregel program, each vertex in the graph is initially assigned its own distinct vertex label to represent its initial community label. At every superstep, vertices send their community label to all out-neighbours and update their label to be the mode community label of incoming messages from their in-neighbours. The pseudo-code for the distributed LPA is presented in Algorithm 1. Note that no a priori information is required for the algorithm to deduce the structures of eventual communities, and although the algorithm can have trivial or oscillating solutions without guarantees on convergence, it works well in practice on real data as we found by running LPA on the largest connected component with 10 supersteps and investigating at least the most influential set of users within each community manually.

Algorithm 1: Distributed Label Propagation

| Result: Returns the label $C_v(T)$ for every vertex $v \in G$ |
| $T$ = Maximum number supersteps; |
| $t$ = 1; |
| $\forall v \in G, C_v(0) = v$; |
| while $t \leq T$ do |
| $\forall v \in G,$ |
| Send $v$’s current label $C_v(t-1)$ to neighbours; |
| Receive sent messages from neighbouring nodes; |
| if $\exists$Label L with highest frequency among $v$’s neighbours then |
| \hspace{1cm} $C_v(t) = L$; |
| else |
| \hspace{1cm} Select a label at random from the neighbours labels; |
| end |
| $t$ = $t$ + 1; |
| end |

4.2.4 Exploring Ideological Diversity

By looking at the twenty most retweeted users, we see that eighteen of these fall into the same pro-BLM community, with 155,229 users. Andy Ngo is in a community with 26,624 users. This is interesting when we
Table 4: Sample tweets from the pro-BLM and anti-BLM communities.

<table>
<thead>
<tr>
<th>Pro-BLM community</th>
</tr>
</thead>
<tbody>
<tr>
<td>i can’t stand by and continue to live in a world where the color of your skin is an automatic target on my family, friends, and neighbors backs. tri-city we must come together to support our communities. THIS. IS. AMERICA. BE THE CHANGE YOU WANT TO SEE. #blacklivesmatter <a href="https://t.co/XIDSnqgx6Q">https://t.co/XIDSnqgx6Q</a></td>
</tr>
<tr>
<td>Thread of people who took it upon themselves to trivialise the current situation going on and #BlackLivesMatter</td>
</tr>
<tr>
<td>#BlackLivesMatter Houston is hosting a protest march this FRIDAY at 2PM starting at Discovery Green demanding justice for #GeorgeFloyd White allies, y’all gotta do better and this is a place to start. Everyone who’s able should be there. <a href="https://t.co/EbWelBrZuzF">https://t.co/EbWelBrZuzF</a></td>
</tr>
<tr>
<td>#Ayana Jones a 7 YEAR OLD CHILD who was shot in the head by an officer, when the officer raided the wrong house. A 7 year old girl didn’t deserve to be killed because of disgusting reckless officers. Acab and BLM, never forget this girls name! #BlackLivesMatter <a href="https://t.co/HCWzabkFv4">https://t.co/HCWzabkFv4</a></td>
</tr>
<tr>
<td>So protest in Huntsville, TX was small, but that was no surprise. We’re a small town and most things just caught up to the present on the outside...at the end of the protest on my way home, I saw something I never noticed. This is why we do what we do. #BlackLivesMatter <a href="https://t.co/gTuCilB7mi">https://t.co/gTuCilB7mi</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Anti-BLM community</th>
</tr>
</thead>
<tbody>
<tr>
<td>Black people are 80 times more likely to kill white people in England/Wales than the reverse! And yet, #BlackLivesMatter more than others? EXPLAIN... Check the stats: <a href="https://t.co/DmPDVVGbSo">https://t.co/DmPDVVGbSo</a> <a href="https://t.co/qxXmuNIh2X">https://t.co/qxXmuNIh2X</a></td>
</tr>
<tr>
<td>#BlackLivesMatter should now be classified as an extreme political hate group... Simple... <a href="https://t.co/mfbs7gCpe9">https://t.co/mfbs7gCpe9</a></td>
</tr>
<tr>
<td>#DontTakeTheKnee #DontTakeTheKnee please get this trending Sick &amp; tires of the #ScumMedia telling us what we should do! Well I say #DontTakeTheKnee #BLM is a terrorist organisation. Do your homework! #AllLivesMatter #WhiteLivesMatter #ISTANDwithDominic Raab @SkyNews</td>
</tr>
<tr>
<td>Then someone gets stabbed and they want the police back after running them out of town. Ha you couldn’t make it up #BlackLivesMatter #blm #blaug #brixton <a href="https://t.co/1aVXQ63UT2">https://t.co/1aVXQ63UT2</a></td>
</tr>
<tr>
<td>Just saw a video of #BlackLivesMatter protest in #Reading - looks like 3 white people have been stabbed and in a bad way! Now if this turns out to be a race attack, I’m going to blame the #Media. They’ve been stoking up tensions between blacks and whites for weeks now!</td>
</tr>
</tbody>
</table>

remind ourselves from Table 3 that he is the sixth most retweeted user, and if we assume that most of his retweets come from his relatively small community, it suggests that he has a very loyal set of core followers. The questions that arises then are if we can identify this core set of followers, and moreover if we also can identify similar core followings in the pro-BLM community. In the same community where we find Andy Ngo, we also have prominent conservative commentators such as Candance Owens, Glenn Beck, Steven Crowder, Paul Joseph Watson, Dave Rubin, and also Republican senator Ted Cruz, and Raheem Kassam from the Reform UK-party (formerly known as The Brexit-Party), along with others. It is worth mentioning that all of the twenty most retweeted users in this community are users with largest followings (over 25,000 followers). Thus, the phenomena of users with small followings reaching a larger audience does not exist to the same extent in this community when compared to the pro-BLM community. The last of the twenty most retweeted users is the official account of the South Korean pop (K-pop) group BTS, who has their own community. The communities for the top ten mostretweeted users are presented in Table 3 and a sample of tweets from the pro-BLM and anti-BLM communities are presented in Table 4. Note how the textual content of the tweets from the two communities differ. By going through the label propagation algorithm we seem to have identified the two different political camps. Moreover, we note that usage of the hashtag #BlackLivesMatter is prominent in the anti-BLM community. Thus, we can conclude that just filtering by the anti-BLM #AllLivesMatter and #BlueLivesMatter would not have sufficed to identify these communities. Thus, through the use of (1) retweet network, which encodes retweets, one of the clearest signals of directional ideological concurrence of the retweeter with the tweeter, (2) distributed label propagation on such a retweet network to detect communities of users who are in ideological concurrence within each community, and finally (3) listing the top $K$ most retweeted tweets within each such community, we have a simple yet effective mechanism to explore the ideological diversity that is representative of the communities, independent of their sizes and activity levels, i.e., the number of users and intensity of interactions in Twitter. We found this simple three-step process to be an effective approach to identifying the pro/anti-BLM tweets before further analysis.
4.3 Summary

In this chapter we have examined how the users in the data set have interacted by defining and looking at the retweet network. We noted that some of the users, despite their relatively small following, managed to become the most retweeted users. This motivates the questions:

- What is the nature of the information diffusion process for retweets in the data set in general? Do tweets from different movements spread in the same way?
- In particular, how does the diffusion process look for a viral tweet when it is initialized by a user with a small following?

We also implemented a community detection algorithm to find groups of users who share similar values. This proved quite successful, by looking at the users, and sampling the textual content of the tweets from these communities. The fact that the sixth most retweeted user Andy Ngo belonged to a relatively small community, raises the question:

- What role does influential users play in the spread of a tweet? Can such influential users be found for all viral tweets?

These are the main questions that motivates the choice of models in Section 6.
5 Joint Media Modeling

In this section we examined the interplay between the Twitter and GDELT data sets by looking at the Granger causality between them. For this we proposed simple two-dimensional Hawkes processes with an exponential kernel. The timeline for this joint modeling was three days after the death of George Floyd over the 24-hours-long period between midnight of 28 May and midnight of 29 May, which is when the protests had just started to spread nationwide across the US, and also become violent.

5.1 Model and Data

In dimension one we had the Twitter data. To control the magnitude of the data we only considered original tweets, i.e. all retweets were filtered out, that had at least one retweet, to filter out tweets made by users with a negligible following. Moreover, we examined the 20 largest communities and identified one anti-BLM (the same community identified in the previous section), and filtered out all tweets made by users from that community, so that we only considered pro-BLM tweets. This left us with 10,774 tweets.

In the second dimension we had records from the GKG-database from GDELT. The records were first filtered on mentioned themes, and only those reporting events of protests were selected. This naturally lead to some noise in the data, due to not being able to precisely filter out only the events mentioning protests relating to the Black Lives Matter-movement. To reduce this noise, we also filtered on records that mentioned George Floyd. While in theory a record could report a BLM related protest without mentioning George Floyd, we reasoned that since our timeline of interest was three days after his passing, most records should mention George Floyd to give the reader some context for the reported protest. To handle that the GKG-database updates in intervals every 15 minutes, every record got a randomised timestamp in the fifteen minute interval prior to it being added into the database, to get the records in continuous time. With this query in the selected time interval, 3,341 records were found.

Given this data, we jointly model events in social and mass media by fitting the multivariate Hawkes process in Definition 2.11. We want to test whether or not Granger causation exists between dimensions 1 and 2 representing events in Twitter and events in mass media from the GDELT project, respectively. As per Theorem 2.2, parameter $\alpha_{12} = 0$ if and only if mass media events do not Granger cause Twitter events, and vice versa for $\alpha_{21} = 0$.

5.2 Results

The data was fitted using python library tick\(^9\). tick requires that the decay parameters $\beta_{ij}$ are given as constants beforehand, which then allows highly efficient fitting of the remaining parameters $\mu_i$ and $\alpha_{ij}$, using accelerated gradient descent [4]. The problem of fitting the decay parameter $\beta$ in the exponential kernel is well-known [30], and is due to the fact that while the baseline parameter $\mu$ and excitation parameter $\alpha$ can be efficiently computed via convex optimisation, this is not always true for $\beta$. With this in mind, we proposed three different models where the decay parameters $\beta_{ij}$ were handled differently:

- $\mathcal{M}_0$: $\beta_{ij} = 1, \forall (i, j) \in \{1, 2\} \times \{1, 2\} =: \{1, 2\}^2$
- $\mathcal{M}_1$: $\beta_{ij} = \beta \in (0, \infty), \forall (i, j) \in \{1, 2\}^2$
- $\mathcal{M}_2$: $\beta_{ij} \in (0, \infty), \forall (i, j) \in \{1, 2\}^2$

To compare the different models, we looked at the Akaike information criterion, the relative likelihood, and the likelihood-ratio test statistic $\lambda_{LR} = -2\ln(\hat{L}_p/\hat{L}_q)$, where $\hat{L}_p, \hat{L}_q$ are the maximum likelihood for models $p$, and $q$, to find the p-value.

\(^9\)https://x-datainitiative.github.io/tick/
The quantity \( \exp((\text{AIC}_p - \text{AIC}_q)/2) \) is interpreted as the probability that model \( M_q \) minimises the estimated information loss, compared to model \( M_p \).

### 5.2.1 Comparison between \( M_0 \) and \( M_1 \)

Setting \( \beta_{ij} = 1 \) for all \( i, j \) in model \( M_0 \) gave us the log-likelihood value of 372.981, and \( \text{AIC} = -733.963 \) (where \( k = 6 \) for the two estimated baseline parameters \( \mu_i \) and the four excitation parameters \( \alpha_{ij} \)). For model \( M_1 \), we did a sequential grid-search over \( \beta \)'s, by using the convex optimiser in \texttt{R} to quickly obtain the most likely \( \mu_i \) and \( \alpha_{ij} \)'s for each fixed \( \beta_{ij} = \beta \), to find the most likely parameter \( \hat{\beta} = 6.17 \), with the maximum log-likelihood value of 384.771 and \( \text{AIC} = -755.542 \) (where \( k = 7 \) since we now also estimate \( \beta \)).

The relative likelihood of the models was \( 2.0624 \times 10^{-5} \), i.e., model \( M_0 \) was \( 2.0624 \times 10^{-5} \) times as probable as model \( M_1 \) to minimize the information loss. Since \( M_0 \) is nested in \( M_1 \), i.e., the parameter space of \( M_0 \) is a proper subset of that of \( M_1 \), we do a likelihood ratio test and reject \( M_0 \) in favour of \( M_1 \) (\( \lambda_{LR} = 23.5781 \), p-value < \( 10^{-7} \)).

### 5.2.2 Comparison between \( M_1 \) and \( M_2 \)

Model \( M_1 \) and \( M_0 \) assume that the decay parameters \( \beta_{ij} \)'s are identically \( \beta \in (0, \infty) \), i.e., the decay parameter within each dimension and between every pair of dimensions is given by the same value. The real-world interpretation of this is that tweets and mass media reports stay relevant for the same amount of time into the future, which seems like a major assumption as mass media dissemination and social media communication are fundamentally different in nature. To account for this, we introduced model \( M_2 \), where each \( \beta_{ij} \) can vary freely in \( (0, \infty) \).

We did a sequential grid search over the 4-simplex, similar to the one-dimensional case of \( M_1 \). We found the most likely values to be \( \hat{\beta}_{11} = \hat{\beta}_{22} = 16.170, \hat{\beta}_{12} = 3.702, \text{ and } \hat{\beta}_{21} = 8.638 \), at the maximum log-likelihood value of 384.772, with \( k = 10 \) and \( \text{AIC} = -749.544 \). Note that despite having three additional parameters, the maximum log-likelihood of \( M_2 \) is close to that of \( M_1 \), with the relative likelihood of the models, likelihood-ratio test statistic, and p-value being 0.04984, 0.002121, and 0.9971, respectively. We therefore do not reject \( M_1 \) in favour of \( M_2 \) and choose \( M_1 \) for further analysis.

### 5.2.3 Fitting the Data using \( M_1 \)

To find whether Granger causality between the two dimensions exists, we were interested in whether parameters \( \hat{\alpha}_{12}, \hat{\alpha}_{21} \) are equal to 0 or not. Fitting the data using model \( M_1 \) with estimated decay parameter \( \beta = 6.1700 \) gave us the following estimated parameters \( \hat{\mu}_1 = 1.000, \hat{\mu}_2 = 0.998, \hat{\alpha}_{11} = 0.986, \hat{\alpha}_{12} = 0.0327, \hat{\alpha}_{21} = 0.0216, \hat{\alpha}_{22} = 0.921 \). Note that the point estimates satisfying: \( \hat{\alpha}_{12} > \hat{\alpha}_{21} > 0 \), implies that there exists Granger causality between reported protests and tweets regarding the BLM-movement, provided we account for the errors in their estimation, i.e., their confidence intervals. We address this next using non-parametric bootstraps.
5.2.4 Hypothesis Testing

The following null hypotheses were proposed:

- $H_{0,12} : \alpha_{12} = 0$, i.e., reports of protests in mass media do not Granger-cause communication events in Twitter related to the BLM-movement.
- $H_{0,21} : \alpha_{21} = 0$, i.e., communication events in Twitter related to the BLM-movement do not Granger-cause reports of protests in mass media.
- $H_{0} : \alpha_{12} = \alpha_{21} = 0$.

To get the confidence intervals for $\alpha_{12}, \alpha_{21}$ we did a non-parametric bootstrap by sampling the observed data with replacement, and then estimating the parameters on the bootstrapped data under model $\mathcal{M}_1$. This was repeated 1000 times.

For $\alpha_{12}$, i.e., the influence of mass media on Twitter, the 99-th percentile bootstrapped confidence interval is (0.000, 0.09405), and therefore we cannot reject the null hypothesis $H_{0,12}$ that $\alpha_{12} = 0$ by the Wald test. Thus, the reports of street protests in mass media do not Granger-cause the pro-BLM interactions in Twitter.

On the other hand, the 99-th percentile bootstrap confidence interval for the parameter $\alpha_{21}$ that models Twitter’s influence on mass media is (0.01479, 0.02949), and therefore we reject the null hypothesis $H_{0,21}$ that $\alpha_{21} = 0$ by the Wald test. Thus, the pro-BLM interactions in Twitter Granger-cause the reports of street protests in mass media. We therefore also reject the common null hypothesis that there is no Granger causality whatsoever between social and mass media events around the BLM-movement, i.e., $H_{0} : \alpha_{12} = \alpha_{21} = 0$.

To estimate type I error, i.e., the probability of rejecting the null hypothesis $H_{0}$, when it is true, we simulated data from the null hypothesis $H_{0}$, i.e., from the most likely parameters in $\mathcal{M}_1$, while restricting $\alpha_{12} = \alpha_{21} = 0$. For each such simulated data, we then performed the Wald test using non-parametric bootstraps by sampling the data with replacement 1,000 times. Only one out of 100 such simulations from $H_{0}$ was rejected giving 0.01 as the Monte Carlo estimate of the Type I error.

5.3 Summary

In this section we jointly model and test hypotheses about causal relationships between interactions in social media and the reports in mass media during the Black Lives Matter (BLM) protests following the death of George Floyd, by employing self-exciting Hawkes processes and their Granger causal inference machinery. We reject the null hypothesis that there is no causal relationship, and show that communication events in Twitter, surrounding tweets that supported the BLM movement, Granger-caused the reports of street protests in mass media from the GDELT project. However, we cannot show that the reporting of street protests in mass media Granger-caused the corresponding communication events in Twitter.

We thus establish a verifiable causal relationship between social media interactions in Twitter that are supportive of the global BLM social movement on one hand, and global mass media reports of street protests in solidarity with the movement on the other.
6 Modelling Retweet Cascades

In this section we will model the diffusion process of a retweet cascade, given one initial tweet. For this we will use marked Hawkes processes with the power-law kernel introduced in Section 2.

The motivation for using a marked Hawkes process stems from the following properties in a retweet cascade, that we seek to capture:

- **Word-to-mouth spread**: When a user shares a tweet, the tweet will organically find its way into new a set of new users, and so on.
- **The magnitude of influence**: Users with more followers tend to get more retweets.
- **Memory over time**: Most of the retweeting by users happen when the users first see it in their timeline.
- **Content quality**: The better a tweet is, vaguely speaking, the more retweets it will get.

Let us now look at the intensity of a marked Hawkes process with a power-law kernel on $\mathbb{R}^+ \times \mathbb{N}$, where each point $(t_i, m_i)$ is a retweet at time $t_i$ where the mark $m_i$ is the number of followers the user who retweets has.

$$
\lambda(t) = \sum_{(t_i, m_i) \in \mathcal{H}_{t,m}} \kappa m_i^\beta (t - t_i + c)^{- (1 + \theta)}.
$$

The property of word to mouth is naturally captured by the self-exciting nature of Hawkes processes; when a tweet is shared, a new set users gets access to this tweet in their time-line, and the intensity of the process, i.e., the probability of a new retweet, will increase. The magnitude of influence is captured by the fact that we are implementing a marked Hawkes Process. Since we let $m_i$ equal the number of followers a user who retweets has, it follows that users with larger followings will contribute to a larger jump in intensity, scaled by parameter $\beta$.

Since $(t - t_i + c)^{- (1 + \theta)} \to 0$ as $t \to \infty$, the kernel is monotonically decreasing, and the property of memory over time is taken care of. $\kappa$ scales the quality of a tweet; larger values of $\kappa$ results in larger jumps in intensity. The motivation for the requirement of this property is that we could have a relatively large retweet cascade that lacks any user with a significantly large following, but still gets a large spread.

All of these properties are captured by other marked kernels, and we did some fitting using a marked exponential kernel. However, the marked power-law kernel gave best results, which is is also backed by relating works [23, 32].

6.1 Fitting Marked Hawkes Processes

The marked Hawkes processes with a power-law kernel has four parameters $\theta = \{\kappa, \beta, c, \theta\}$. We estimate these by computing the maximum likelihood. The log-likelihood for the intensity is of the following form:

$$
\mathcal{L}(\kappa, \beta, c, \theta \mid \mathcal{H}_t) = \log \mathbb{P}\left(\{(m_i, t_i), i = 1, ..., n\}\right)
= \sum_{i=1}^{n} \log (\lambda(t_i)) - \int_0^T \lambda(\tau) d\tau
= \sum_{i=2}^{n} \log \kappa + \sum_{i=2}^{n} \log \left(\sum_{t_j < t_i} \frac{m_j^\beta}{(t_i - t_j + c)^{1+\theta}}\right)
- \sum_{i=1}^{n} \int_{t_i}^{T} \kappa m_i^\beta (t - t_i + c)^{-(1+\theta)} dt
= \sum_{i=2}^{n} \log \kappa + \sum_{i=2}^{n} \log \left(\sum_{t_j < t_i} \frac{m_j^\beta}{(t_i - t_j + c)^{1+\theta}}\right)
- \kappa \sum_{i=1}^{n} m_i^\beta \left[ \frac{1}{\theta c^\theta} - \frac{(T + c - t_i)^{-\theta}}{\theta} \right].
$$

(12)
The term $\int_0^T \lambda(\tau) d\tau$ is a normalisation factor that we get by integrating the event rate over the time interval $(0,T)$. This is non-linear and solved numerically. For this we used the R library `evently`\(^{10}\), which builds on `AMPL`\(^{11}\). Due to the requirement of `AMPL`, we were not able to implement `evently` in Databricks using Spark. Instead a remote machine was set up, and via a Docker container with R and all required packages installed, the fitting was done.

### 6.1.1 Results

The remote machine that was set up was quite powerful with 64GB of RAM. Regardless, when trying to fit retweet cascades with more than roughly 3,000 retweets, the machine ran into memory issues. Note that the largest retweets cascades in our data set had around 300,000 retweets. These were certainly rare, but would have been interesting to study in more detail. We should however point out that cascades of around 3,000 are still relatively big, and certainly big enough to have a diffusion processes of interest.

Fitting was done on 20 retweet cascades initialised by users from both the BLM and anti-BLM communities identified in Section 4. By just comparing the plots of the intensities over time, and the fitted parameters, no clear distinction between cascades from the two communities could be made.

![Figure 10: Cascade initialised by a relatively highly influential user](image)

However, in both communities, retweet cascades of the type (Figure 11) where the intensity suddenly spikes when an influential user joins the diffusion process, were found.

This phenomena is the most probable explanation for how some of the largest cascades in the data set that were initialised by users with relatively small followings managed to diffuse to a large group of users.

\(^{10}\)https://github.com/behavioral-ds/evently
\(^{11}\)https://AMPL.com/
6.2 User Influence

Taking with us the results from fitting different types of cascades, and discovering the type of information diffusion where a user with a large following joins the cascade at a later time, we here present an idea on how to recreate a probable branching process of the retweet diffusion, using the fitted parameters of the kernel in our marked Hawkes process.

Definition 6.1. Given a retweet cascade, let a diffusion scenario $G$ be a directed tree, where for each $v_i, v_j \in G$, $v_i$ has an edge to $v_j$ if the retweet $v_j$ is a direct retweet of $v_i$.

We remind ourselves that we via Twitter's API have no information on how the actual branching process initialised, as all retweets point only to the original tweet. The idea comes from Rizoiu et al. [28].

Definition 6.2. Let

$$\phi^p(m_i, t) = \kappa m_i^\beta (t + c)^{-(1+\theta)},$$

be the marked power-law kernel. We define the probability of direct retweet as

$$p_{ij} = \begin{cases} \frac{\phi^p(m_i, t_{ij} - t_i)}{\sum_{k=1}^{i-1} \phi^p(m_k, t_{jk} - t_k)} & i < j, \\ 0 & i \geq j. \end{cases} \quad (13)$$

The probability of a direct retweet $p_{ij}$ thus tells us how likely it was that the $j$:th retweet in the cascade was a direct retweet from the $i$:th retweet, and thus a direct descendant in the diffusion scenario $G$. Since we have the marked kernel from the Hawkes processes in the definition, naturally retweets close to each other in time will have a higher probability of being direct retweets. The definition also takes into consideration that users with more followers will have a larger probability of getting direct retweets.
Definition 6.3. We define the pairwise influence as

\[ r_{ij} = \begin{cases} 
    \sum_{k=1}^{j-1} r_{ik} p_{kj} & i < j, \\
    1 & i = j, \\
    0 & i > j.
\end{cases} \]

The pairwise influence gives us a measure of how much influence tweet \( v_i \) exerts over retweet \( v_j \). This influence can either be via a direct retweet in a diffusion scenario, but also when \( v_j \) is an indirect retweet of \( v_i \), i.e., when there exists a path from \( v_i \) to \( v_j \) in a diffusion scenario.

Definition 6.4. We define the total influence of a tweet \( v_i \) in a retweet cascade as

\[ \varphi(v_i) = \sum_{k=1}^{n} r_{ik}. \]

We now have a measure of how influential users are in a retweet cascade. Note that while we are talking about the influence of a tweet, we are actually taking the heuristic shortcut and assuming that this is the influence of the user of that tweet; the reasoning behind defining the influence in terms of tweet influence, is that technically a user may retweet the same tweet more than once, and these retweets will then be given different total influence.

In Table 5 we present an example from a retweet cascade.

<table>
<thead>
<tr>
<th>Time</th>
<th>Number of followers</th>
<th>Total influence</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00</td>
<td>1,475</td>
<td>195.00</td>
</tr>
<tr>
<td>621.00</td>
<td>142,881</td>
<td>161.28</td>
</tr>
<tr>
<td>565.05</td>
<td>16,527</td>
<td>143.13</td>
</tr>
<tr>
<td>165.38</td>
<td>2,285</td>
<td>118.81</td>
</tr>
<tr>
<td>304.53</td>
<td>591</td>
<td>27.39</td>
</tr>
<tr>
<td>737.97</td>
<td>27,081</td>
<td>24.18</td>
</tr>
<tr>
<td>1,550.91</td>
<td>51,256</td>
<td>22.69</td>
</tr>
<tr>
<td>544.08</td>
<td>546</td>
<td>20.32</td>
</tr>
</tbody>
</table>

Table 5: The 8 most influential users in a retweet cascade.

Note that the user who joins the retweet cascade at time 1,551 is given an influence score of 22.69, although this user has quite a large following of around 51,000 users. Compare this to the user who joins the cascade at time 565, with a following of roughly 17,000 users. Due to the fact that this user joins the cascade earlier, their influence score is much higher at 143.13, even though their following is three times as small.

6.3 Summary

In this section we modelled retweet cascades using R library `evently`. Due to the limitations of hardware and lack of efficiency for the fitting of these models, we were only able to look at retweet cascades with magnitude of roughly 3,000 retweets. From the fitting of retweet cascades, two distinct patterns in the diffusion process were identified; one where the intensity of Hawkes process starts high and peaks close to the initialisation of the retweet cascade, and one where the intensity reaches its peak at point later in time after its initialisation. Influenced by these two patterns, we introduced a heuristic method of measuring influence score of users in a retweet cascade, by recreating a probabilistic diffusion process retroactively via the fitted Hawkes process. While no major conclusions from this will be drawn in this thesis, our hope is that these models can be used as tools in other fields, e.g., social sciences, for doing comparative research. In an ideal future with a better scalable estimator for the parameters at hand, we would be able to fit all retweet cascades in our data set,
which would allow us to do more interesting work, such as looking at the distribution of the parameters and also identify the most influential users for the whole data set, by summing and normalising the users influence from all cascades.

6.4 Related Work

The idea of using Hawkes processes to model information diffusion and in particular retweet cascades is well established, often in the context of predicting, the final size of a retweet cascade given the first initial tweets in a given time interval. In this subsection we present brief overviews of some of the articles we read during writing of this work, and found interesting.

A well known model for prediction is the SEISMIC (Self-Exciting Model of Information Cascades) model by Zhao et al. [32]. The SEISMIC model also implements a marked Hawkes process with a power law kernel, but unlike our approach, they fix parameters for the kernel for all retweet cascades. Instead they extend the Hawkes process by introducing a infectiousness parameter \( p_t \), which aims to model how likely a post is to be re-shared at time \( t \).

The intensity for the point process is of the form

\[
\lambda(t) = p_t \cdot \sum_{(t_i, m_i) \in H_t, m} \phi_{m_i}(t - t_i),
\]

which is a Cox-process (i.e., a double stochastic process), since the infectiousness \( p_t \) is stochastic. The main idea for SEISMIC is then to, given a retweet cascade at some fixed time \( t \) (e.g., \( t = 60 \) minutes) find an estimate of the infectiousness \( \hat{p}_t \) (which is bounded by \( \hat{p}_t < \frac{1}{\nu} \), to not explode exponentially and where \( \nu \) is the branching factor of the Hawkes process) and then from \( \hat{p}_t \) estimate the final size of the retweet cascade.

Mishra et al. [23] article – which was our main source of inspiration for the modelling done in this section – also presents models for prediction, and argues that the kernel of the Hawkes process should not be fixed, since retweet cascades may die out quickly or slowly and still end up with roughly the same number of retweets. Therefore, they fit each cascade in a similar fashion as was done by us using a non-linear solver. Moreover, they add predictive layer using the estimated parameters as features and then training a Random Forest regressor, to predict the final size of the cascade after a given initial time. They also introduce a purely feature driven predictor with basic user features (number of followers, number of posts etc.), temporal features (waiting time for posts in the cascade), volume (i.e., the size of the cascade at the given time), past user success (average size of cascades previously initialized by the given user), and – perhaps more interesting in the context of this work – a hybrid model, which combines the Hawkes process with a predictive layer, and the purely feature driven predictor.

All three of these models outperform the SEISMIC model when tested on two data sets (where the first one is the same data set used in Zhao et al. for the SEISMIC model) containing cascades of size greater or equal to 50, with 30,463 and 110,954 cascades respectively, and statistics mean length of 160 and 158, and median of 95 and 90, respectively. Note that most cascades are therefore rather small compared to the cascades we managed to fit from the BLM-data set. Moreover, all predictors based on Hawkes processes will fail when the conditional intensity at the time of prediction is too high, i.e., when the branching factor \( \nu \geq 1 \) in Mishra’s models, and when the infectiousness parameter \( p \geq \frac{1}{\nu} \). This suggests that there is an upper bound on the magnitude of retweet cascades, when modelling via Hawkes processes is successful.

In Zhou et al. [33] a multivariate Hawkes process is implemented to discover the social influence of users in a network. In the article, a multivariate Hawkes process in 500 dimensions, each representing a popular website, is fitted on a data set containing timestamps of the event that one of the 500 sites creates a hyperlink to another site, to find the most influential sites that are quickest in detecting trends on the internet, and moreover the community structure of these sites.

The main idea is to let every user \( u = 1, \ldots, U \) in the data set be represented by one dimension in a multivariate Hawkes process in \( U \) dimensions. The estimated baseline intensities \( \mu_u \), and excitation parameters \( \alpha_{uv} \) (which
captures the influence of excitation from dimension \( v \) to \( u \) are collected into matrix form \( \mathbf{\mu} = \mu_u, \mathbf{A} = \alpha_{uv} \),
where \( \mathbf{A} \) is defined as the infectivity matrix. To estimate the parameters in the Hawkes process, a low-rank and sparse regularisation is imposed on \( \mathbf{A} \). The motivation for this is that the sparsity of the infectivity matrix \( \mathbf{A} \) captures the fact that most users only influence a small number of users, while there can be a few very influential users. The low-rank structure of \( \mathbf{A} \) is meant to capture the structure of communities in the network of users, where we interpret a set of linear dependent column vectors as a community. An algorithm for solving the optimisation problem of estimating \( \mathbf{A} \) so that it is both low-rank and sparse is presented in the article.

This approach should in theory work for Twitter data and its restrictions given by the API, and could then be another way of identifying influential users and community structures. The data would then be tweets represented as a tuple of a timestamp and user, with no regard to whether they are original tweets or retweets. Other features such as how many followers a user has would also be discarded. The estimated influence of a user \( u \) in the data set, would then be given by the \( u \):th column in the low-rank and sparse estimated infectivity matrix \( \mathbf{A} \).
7 Conclusion

In this work we have analysed Twitter and news data relating to the Black Lives Matter-movement during a one month time span following the death of George Floyd. For this, we implemented open-source pipelines through MEP and SPARK-GDELT to process the data, i.e., extract, load, transform, explore, from scratch and at scale, on cloud infrastructure, and by employed self-exciting Hawkes processes and their Granger causal inference machinery.

We identified both pro- and anti-BLM tweets through a network analysis of the Twitter data to identify communities of users who have a shared ideology among an ideologically diverse set of communities.

We rejected the null hypothesis that there is no causal relationship, and showed that communication events in Twitter, surrounding tweets that supported the BLM movement, Granger-caused the reports of street protests in mass media from the GDELT project. However, we could not show that the reporting of street protests in mass media Granger-caused the corresponding communication events in Twitter.

Lastly, we modelled retweet cascades using marked Hawkes processes, to study the diffusion processes of viral tweets. Two different cases were identified, and the key difference between them were whether or not a highly influential user partook in the retweet cascade at the beginning or later stages of the diffusion process. From this we presented a probabilistic diffusion scenario, where each user in the process is also given a measure of influence in the given cascade.
References


