Characterising the Social Media Temporal Response to External Events

Peter Mathews

A thesis submitted for the degree of Doctor of Philosophy

School of Mathematical Sciences
The University of Adelaide

April 2019

Abstract

In recent years social media has become a crucial component of online information propagation. It is one of the fastest responding mediums to offline events, significantly faster than traditional news services. Popular social media posts can spread rapidly through the internet, potentially spreading misinformation and affecting human beliefs and behaviour. The nature of how social media responds allows inference about events themselves and provides insight into human behavioural characteristics. However, despite its importance, researchers don't have a strong understanding of the temporal dynamics of this information flow.

This thesis aims to improve understanding of the temporal relationship between events, news and associated social media activity. We do this by examining the temporal Twitter response to stimuli for various case studies, primarily based around politics and sporting events. The first part of the thesis focuses on the relationships between Twitter and news media. Using Granger causality, we provide evidence that the social media reaction to events is faster than the traditional news reaction. We also consider how accurately tweet and news volumes can be predicted, given other variables. The second part of the thesis examines information cascades. We show that the decay of retweet rates is well-modelled as a power law with exponential cutoff, providing a better model than the widely used power law. This finding, explained using human prioritisation of tasks, then allows the development of a method to estimate the size of a retweet cascade. The third major part of the thesis concerns tweet clustering methods in response to events. We examine how the likelihood that two tweets are related varies, given the time difference between them, and use this finding to create a clustering method using both textual and temporal information. We also develop a method to estimate the time of the event that caused the corresponding social media reaction.

Declaration

I certify that this work contains no material that has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree. I give consent to this copy of my thesis, when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time. I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Acknowledgements

Thanks to everyone who participated in my PhD journey.

First and foremost, thanks to my supervisors Professor Nigel Bean, Dr Lewis Mitchell and Dr Giang Nguyen for their guidance throughout the PhD.

Thanks also to my colleagues for the productive academic discussions and enjoyable times throughout my PhD, particularly Yao Li, Dong Gong, Bohan Zhuang, Mingkui Tan, Qinfeng Shi, Jing Liu, Peng Wang, Lingqiao Liu, Xiusen Wei, Shuang Li, Adrian Johnston, Lachlan Birdsey, Dustin Craggs, Luke Keating-Hughes, Brett Chenoweth, Max Glonek, James Walker, Maha Mansor, Caitlin Gray, Angus Lewis and Dennis Liu.

Thanks to the Data to Decisions Co-operative Research Centre (D2DCRC) and the Australian Research Council Centre of Excellence for Mathematical and Statistical Frontiers (ACEMS) for providing financial support during my PhD.

Thanks to all my maths and computer science teachers along the path to my PhD, particularly my high school maths teacher Anthony Harradine.

Finally, thanks to my family for their support.

Publications

The following peer-reviewed conference publications contain preliminary reports of the findings in this thesis:

Peter Mathews, Lewis Mitchell, Giang Nguyen, and Nigel Bean. The nature and origin of heavy tails in retweet activity. In *The 26th International Conference on World Wide Web Companion*, pages 1493-1498, 2017.

Peter Mathews, Caitlin Gray, Lewis Mitchell, Giang Nguyen, and Nigel Bean. SMERC: Social media event response clustering using textual and temporal information. In *The 2018 IEEE International Conference on Big Data*, pages 3695-3700, 2018.

Contents

Chapter 1

Introduction

Since the early 2000s, the global usage of social media has expanded rapidly, leading to it becoming much more than a way for people to connect with their friends [4]. Currently, in 2018, social media is regularly used to receive news (Facebook, Twitter), share photos (Instagram), maintain professional contacts (LinkedIn), share interesting websites (Facebook, Twitter) and comment on the latest videos (YouTube). Importantly, social media has become one of the fastest responding mediums to a wide range of offline events, including sport, terrorist attacks and natural distasters. Many traditional news sources, such as newspapers or nightly television news, have a 24-hour cycle [88]. News organisations with online websites (e.g. CNN) are generally faster, but still have a delay while they write, edit and publish a story about breaking news. Social media is much faster and can react in the time it takes for someone to write and send a tweet, or snap a picture and upload it to Instagram [87]. This is particularly evident during emergencies [8], when information spreads rapidly through social media channels. The largely unregulated nature of social media means there is a high propensity for misinformation spread, such as occurred duing the 2011 London riots [10]. In such situations, it is valuable to understand the temporal dynamics of how this information spreads through social media channels.

The way in which social media reacts in the initial stages after an event tells us much about the event itself. The magnitude and decay of the response gives insight into the longevity of human interest about a topic [90]. As social media responds very quickly, we can make these estimations shortly after the event has occurred. We can estimate the public interest about a social media post by predicting the total size of the information cascade, the number of times that a social media post about the event is shared [60, 83, 97, 180]. The responsiveness of social media also facilitates social sensing, estimating public awareness, in near real time [45].

1.1 Research goals, scope and limitations

The central aim of this thesis is to improve understanding of the temporal relationship between events, news and associated social media activity. We wish to explore and characterise this temporal relationship, and attempt to answer research questions which naturally arise. For example, is social media a leading or lagging indicator of major events? Does it drive the news or is it merely a reflection of traditional media outlets?

This thesis explores such questions through a series of case studies on contemporary issues and datasets. In 2018, there are over 500 million tweets per day worldwide [75] and according to the news aggregation website Twingly, there are over 3.2 million news articles per day [152]. It would be infeasible to collect and analyse this volume of data so we consequently must limit the scope of our work.

We limit our social media platform to Twitter, primarily due to the public availability of data. To facilitate subsequent analysis, for the most part we restrict our choice of topics to politics and sport. Politicians are regularly featured in the news, particularly during election campaigns, while sporting events have set start times and predefined hashtags. Within the genre of sport, we collect tweets from cricket and Australian rules football, two of the most popular sports in Australia.

1.2 Twitter as a data source

Twitter is one of the few social media platforms that has a public API. Although analysis has been done on other social media platforms including Facebook [69], Pinterest [182], YouTube [71] and LinkedIn [155], Twitter remains the most commonly used social media data source for research [166]. Twitter was founded on 21 March 2006 and has consistently grown since this time [141]. The main form of interaction on Twitter is users updating their status, often referred to as sending a *tweet*, a text message of up to 280 characters.

Twitter data can be accessed programmatically via its Application Programming Interface (API) [153], which has a series of functions allowing a program to operate an online Twitter account. Twitter has two main APIs, REST (Representational State Transfer) API and Streaming API [153]. The REST API provides programmatic access to read and write Twitter data. Examples of usage include authoring a new tweet and reading author profile data or follower data. However, the standard (freely available) version of the API has functional restrictions and rate limits, which in particular reduces the ability to obtain older data. The Streaming API gives programmatic access to the global stream of tweet data in selected languages. Tweets can be collected in real time filtered on keywords. If a keyword is sufficiently specific, all data related to this keyword can be extracted. Otherwise, a sample of online traffic is returned, up to 1% of the total feed.

1.3 Literature gap

Within the overall theme of the temporal relationship between events, news and the associated social media activity, we focus our attention on the following identified literature gaps.

The temporal relationship between the volume of social media activity and the news is not well understood [34, 123]. In particular, there are no clear conclusions about the causality relationship between news reports and Twitter activity [123]. There also do not exist methods to predict whether news services will report on a topic by analysing the Twitter reaction after events. The effectiveness of such methods provides evidence about whether Twitter is a leading indicator of traditional news services.

While mathematical modellers have studied information propagation through processes such as rumour cascades, a detailed understanding of the temporal dynamics of these processes remains lacking. For example, the distributions of repost times in social media is commonly modelled as a power law [70, 180], but these works do not conduct in-depth explorations of the appropriateness of such distributions or their fits to data. Furthermore, there is a lack of appropriate techniques to account for the diurnal (daily) cycle of how Twitter activity varies throughout the day.

For effective microblog summarisation and event detection, there is a need for improved techniques to cluster tweets in an unsupervised manner and associate these clusters to an event with an estimated time. Other authors have created methods for specific event prediction from Twitter such as predicting earthquakes and soccer goals [59, 147, 181]. They have also developed methods for microblog summarisation using Twitter [82, 136], primarily based on keyword frequency. However, these methods are not able to take a series of input tweets, cluster them into associated categories, and then estimate the time of the event that caused this social media response.

1.4 Nature of social media analysis

1.4.1 Specific challenges

Although social media is a potentially powerful tool, there are challenges that must be overcome. Many authors have outlined the difficulties of analysing Twitter data, e.g. [142, 162]. There are bots, tweet processing challenges, diurnal effects and complex social networks which are difficult to detect. In addition, social media is in a constant state of evolution, so older results are not guaranteed to still hold today.

Advertisers use bots on Twitter for automated messages. These bots can create a large quantity of output but very little new content or information. They have the potential to adversely impact Twitter analysis which generally has the goal of understanding human sentiment and behaviour. Researchers have developed a variety of methods to identify bots on Twitter, e.g. Clark *et al.* [32], which used a natural language processing approach, or Varol *et al.* [157], which used over 1000 features including friends, tweet content, sentiment, network patterns and activity time series. Twitter bots are not prohibited, provided they adhere to Twitter's terms of service [154]. However, when looking to analyse human responses on Twitter, researchers generally wish to eliminate social media posts created by these bots.

Tweets are short messages of maximum 280 characters (extended from 140 characters on 7 November 2017). They often contain abbreviated words, emoticons and slang terms. Consequently Twitter traffic can be challenging to process [160], as it is difficult to conduct automated text analysis and extract meaningful information from the data. Many techniques have been proposed to deal with these challenges when analysing Twitter data. For example, Marujo *et al.* [101] developed a method for automatic keyword extraction on Twitter. Vosoughi *et al.* [160] developed *Tweet2Vec*, a way to create a vector representation of each tweet for further analysis. Despite these techniques, Twitter data is fundamentally challenging to process, adding to the difficulty of making rigorous statistical conclusions.

The demographics of Twitter users are not the same as the general population. Twitter tends to have users who are younger, richer, more urbanised and more likely to be white [106,111]. Some techniques exist to estimate the demographics of Twitter users, but are imperfect, even for relatively straightforword tasks such as gender detection. Approximately 67% of people use their real names [112], so the gender of a user can be estimated from the social security statistics of their quoted username [106]. Cesare *et al.* [106] also developed a more accurate machine learning based method using an ensemble classifier on supervised input data that can predict the gender of Twitter users with 82.8% accuracy. Even with such techniques available, the bias in the demographics of Twitter makes it difficult to conduct population-level estimation, which would be needed for many tasks.

A final challenging aspect of Twitter that is of relevance to this thesis is diurnal cycles. Twitter activity varies depending on the hour, the weekday and even the season [111]. Each city has its own daily cycle of social media activity, with the shape largely determined by the working hours and culture within the city. People have a tendency to synchronise their activity, with the exact times varying throughout the year as the climate and daylight hours change. The global nature of Twitter adds further complexity with time zones, and chosen topics having varying interest levels depending on the location. We discuss previous work on diurnal cycles further in Section 2.5.

1.4.2 Limitations to conclusions from social media research

Twitter data has been used for a wide variety of research purposes including finance [20, 53, 183], health [36, 85] and politics [25], with varying levels of success. As we shall discuss, some of the most well-publicised claims were later shown to have exaggerated results.

Bollen *et al.* [20] measured public mood from Twitter data and compared this to world events such as stock market movements, the 2008 US presidential election and Thanksgiving Day. They showed that social, political, cultural and economic events have a direct effect on public mood. Following this, many authors have attempted to predict stock market movements from Twitter data and claimed success, e.g. [53,120,183]. However, although some methods do a reasonable job of modelling financial systems in the past, researchers have been unable to create methods able to consistently achieve accurate predictions of future stock market trends [24].

In the field of politics, several authors have claimed to be able to predict the result of elections from social media data [25]. However, these studies are generally post-hoc analysis [58], claiming that a prediction could have been made on previous elections, rather than actually predicting future results. So far, a reliable method to predict future elections is yet to be created [58].

In contrast to these controversial articles, in this thesis we do not attempt to make claims about future prediction capability using social media data. Our research primarily focuses on analysing temporal trends from social media and is justified using established statistical techniques.

1.5 Ethical considerations

When users post on Twitter, the default privacy settings cause their post to be publically available for anyone to observe and potentially store. However, users might not be aware that their social media posts are being used for academic research. Norval *et al.* [117] argued that the conditions of informed consent are not always met, and that research into some areas such as health may lead to undesirable consequences. As an example, in March 2018 there was negative public reaction about data mining company Cambridge Analytica using Facebook data to generate personality profiles in order to influence elections [128].

For social media research, any potential risks need to be carefully considered beforehand. Our research was included in a low-level human ethics application approved by the University of Adelaide (H-2016-281). Our primary interest is in population-level statistics rather than individuals, and when specific example users are required, we choose public figures, such as Donald Trump. When conducting data aggregation, we remove identifying information, such as usernames, in order to protect individuals' privacy.

1.6 Research overview and thesis structure

The outline of this thesis is visually displayed in Figure 1.1.

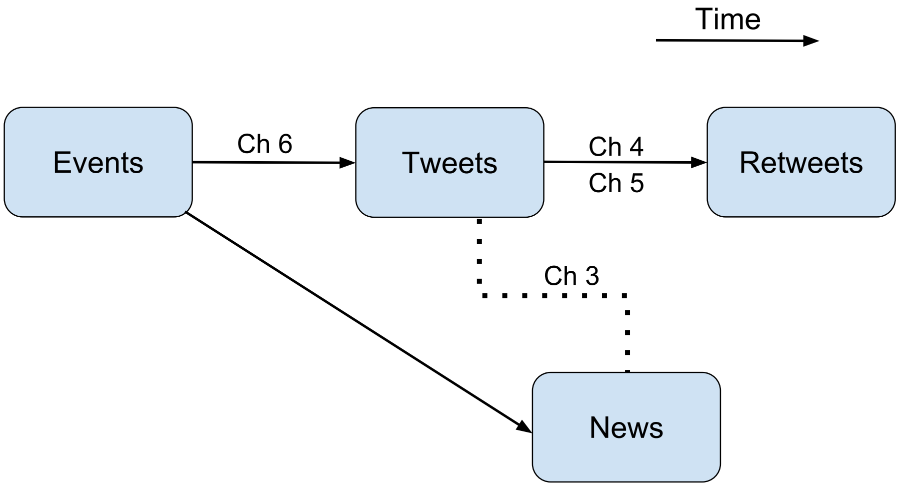


Figure 1.1: The temporal relationship between events, news, tweets and retweets is analysed in this thesis. The associations between chapters and topics are indicated, and the causality relationships are shown with directed line segments. The causality relationship between tweets and news is less clear and is investigated in Chapter 3, represented here as a dotted line.

Social media has disrupted the news industry, providing *tips* of breaking news stories and fast responses from eyewitnesses of events. Global news services are currently in a transition phase between manually reading social media, to automated processing [94]. In Chapter 3 we show that the number of tweets about given famous individuals is highly correlated with the number of news stories about them. We analyse the *Granger causality* between tweets and news and show that tweets Granger-cause news, but not vice-versa. This provides evidence that on average, Twitter responds faster to external events than traditional news services. To further illustrate the close relationship between tweets and news, we demonstrate that the current level of public Twitter activity about a given topic, in addition to other variables, can be used to predict the current number of news stories about a topic.

On Twitter, users have the ability to share the content of other users by *retweeting*. This can lead to a cascade of information through the internet via this social media channel. We are interested in the mechanics of this process and, specifically, the distribution of time gaps between the initial tweet and the retweet. In Chapter 4 we demonstrate, for a selected group of popular accounts, that the distribution of retweets is well-modelled by a power law with exponential cutoff. We show that this provides a better fit than the previously used power law distribution. There is a strong link between the way people behave and their temporal reactions on social media. We provide an explanation of the observed distribution of retweet times using a model of human behaviour, specifically how people execute tasks based on prioritisation. This governs how frequently people check their social media and the subsequent distribution of response times [103].

A well-studied problem in social media analysis is to estimate the total number of retweets from an initial tweet [60,83,97,180]. In Chapter 5 we simulate both individual human retweet behaviour and population level retweet cascades, and develop a technique to estimate the number of total retweets from a seed tweet. Different to existing methods, our retweet count estimation method is based only on observing the times of the initial retweets and, optionally, the category of the author of the tweet. We do not use other information such as tweet text or past history of the author of the tweet. Using only this limited information, we can accurately estimate retweet counts, particularly for tweets of news stories.

When people post on social media, they are often responding to some kind of stimulus [70], whether it is directly observing an event, watching television, or other social media posts. Naturally, the highest rate of reactionary tweets occurs close to this stimulus, while the event is still fresh in people's minds, with the response rate dropping over time. Using manually classified tweet data, in Chapter 6 we calculate the probability that a pair of tweets will be in response to the same stimulus. This allows us to cluster tweets into related groups based on both textual and temporal information, by first calculating a text-based similarity measurement and then combine it with the time distance between tweets. We show that using temporal information correctly removes a high percentage of unrelated tweets from clusters.

When attempting to automatically summarise a microblog, we wish to know both the nature of key events, and the time when they occurred. In Chapter 6 we also create a model for the temporal response between events and the response tweets using a Weibull distribution. We then use this model in reverse to estimate when events occurred based on the set of response tweets. This allows us to automatically cluster tweets into related groups, and predict the time of the event that caused the tweets.

1.7 Key contributions to new knowledge

In this thesis, we make significant contributions to understanding the temporal response on Twitter to real world events and news. We also develop novel and improved methods for social media analysis. We identify the following as our most important contributions to new knowledge (noting that the conclusions apply only to the datasets that we analysed, as we will discuss in the relevant chapters):

- Tweet rates Granger-cause news activity rates, but not vice-versa (Ch. 3).
- The decay of retweet rates over time is well-modelled by a power law with exponential cutoff (Ch. 4).
- The power law observed in retweet rate decay can be explained by human prioritisation of tasking, while the exponential cutoff can be explained by human loss of interest in topics over time (Ch. 5).
- The likelihood of two tweets being related to the same event decays exponentially with the time gap between them (Ch. 6).

We identify the following as our most important new methodological contributions for social media analysis and processing:

- A method to adjust for the diurnal cycle in Twitter activity rates (Ch. 3, 4).
- A method to predict the size of a retweet cascade, particularly effective for tweets of news stories. (Ch. 5)
- An unsupervised method to cluster tweets using both textual and temporal information (Ch. 6).
- A method using reactionary tweets to accurately estimate the time of events (Ch. 6).

Chapter 2

Literature Review and Background

There exists a significant amount of related work about the relationship between events, news and associated social media reaction. In this chapter we discuss key publications upon which our work builds, and illustrate where our research fits into the bigger picture of social media analysis. In Sections 2.1 to 2.4, we discuss literature relevant to Chapters 3 to 6 respectively. In Section 2.5 we discuss diurnal cycles, a topic relevant to several aspects of Chapters 3 and 4. Finally, in Section 2.6, we discuss tools and techniques used throughout this thesis for analysis including decay functions, filters, model selection criteria and prediction techniques.

2.1 The relationship between tweets and news

Here we discuss previous literature related to Chapter 3, *The Temporal Relationship between Tweets and News*.

Researchers have long been interested in the relationship between Twitter and the news. In a 2010 Twitter paper, Kwak *et al.* [84] asked, "What is Twitter, a Social Network or a News Service?" At the time Twitter had only 41 million users, and over 85% of trending topics were headline news or persistent news. Even though the use and dynamics of Twitter have evolved in the eight years since it was published, the important relationship between social media and news identified by this paper is still relevant today. Wu *et al.* [170] analysed Twitter as a news source and discussed how retweet bursts from news-related tweets occur over a shorter period of time than for non-news tweets. They identified news sources as supernodes with high numbers of followers, a finding that is important for our cascade size estimation in Chapter 5.

The temporal relationship between news and tweets is not fully understood. Researchers have speculated about the direction of causality between these two variables, but haven't established clear conclusions. Petrovic *et al.* [123] examined the overlap of news reporting in Twitter and newswire, and whether Twitter reports the news faster than traditional newswire providers. This was implemented by examining the time when key news events were first mentioned on newswire and on Twitter. In some cases, Twitter responded first while in other cases, newswire did, and they could not conclude whether one source leads the other in terms of breaking news. They also examined this causality relationship in further detail based on topic; although some trends were found, no clear causality results were obtained. When analysing the temporal relationship between news and tweets it is necessary to link entities from both sources. One technique to do this is outlined in Onishi *et al.* [119] which performs relevance matching of news and tweets based on keywords. Conway *et al.* [34] looked at whether candidate tweets affected the news conversation in the 2012 US Presidential primaries. They found evidence of correlation between Twitter and news sources, but could not conclude definitively that candidate Twitter feeds directly influenced traditional media.

With the ability to automate tweet processing comes the potential to automate news analysis. Xie *et al.* [173] developed *TopicSketch*, a real-time "bursty" topic detection method using Twitter data. The authors found that the method could detect news events in a short time after they occurred, and significantly before any news reporting on the event. Liu *et al.* [94] developed *Reuters Tracer*, a way to automate news production from Twitter data. It automatically reads large volumes of tweets and is able to quickly detect news, giving Reuters a claimed 8 to 60-minute head start over other news media. The system works by filtering noise (including spam and advertising), clustering tweets, detecting news-worthiness, summarising events, estimating scope and determining the event location.

Our work in Chapter 3 uses data from the 2016 US Presidential Election. In 2019, Bovet and Maske [21] analysed the influence of fake news in Twitter during the 2016 US presidential election. Although the focus was on fake news, this article also analysed the causal relationship between Twitter dynamics and news media sources. The authors used a multivariate network reconstruction of the links between the activity of top news spreaders and supporters of the presidential candidates based on a causal discovery algorithm. They demonstrated Granger causality between the top 100 news spreaders and the rest of the population.

2.2 The distribution of retweet times

Here we discuss previous literature related to Chapter 4, *The Temporal Distribution of Retweets*.

2.2.1 Information diffusion on social media

Information diffusion has unique characteristics when it occurs through social media. While information diffusion and rumour cascades are widely studied topics generally in applied mathematics, here we focus on data-driven studies. Sun *et al.* [144] modelled contagion through the Facebook news feed, and found that social media information diffusion is often different to traditional cascade theory, where it is assumed that information flows are from chain reactions beginning at a few nodes. In contrast, in social media networks where user engagement is high, information is able to enter a system from multiple sources.

The network dynamics of Twitter strongly affect the way information spreads. Nguyen *et al.* [116] analysed influence within Twitter communities and showed that whether a user retweets a message is strongly influenced by the first of his followees who posted that message. Wu *et al.* [171] performed an extensive analysis of the production, flow and consumption of information on Twitter. They found that different user types, and content types, exhibit dramatically different characteristic lifespans. The lifespan of contents of a social media post varies depending on whether it is from media, celebrities, organisations or bloggers. Yang *et al.* [176] developed a *linear influence model* to estimate the influence of nodes in a social network. From this they were able to predict the temporal dynamics of information diffusion. This paper challenged the traditional belief that flows of information on social media could be modelled as diffusion processes over underlying social networks. They claimed that such models were unable to fully capture the complexity of a real social network.

Epidemiology-based approaches have been broadly applied with some success [18, 35,121,168]. Woo *et al.* [168] modelled information diffusion with the SIR (Susceptible - Infected - Recovered) model commonly used in epidemiology. They concluded that such a model performed well at predicting the spread of extreme ideology messages on a Jihadi forum. There are parallels between disease spreading and information propagation: both depend heavily on the population network structure and the infectiousness of the disease / social media post. However, there are also fundamental differences, with the SIR structure being arguably a poor description of how information spreads [43]. Consequently we will not use epidemiology techniques for our own social media analysis and modelling.

Many information propagation characteristics of Twitter have been analysed and modelled. Bild *et al.* [19] showed that lifetime tweet counts are fitted well by a Type-II discrete Weibull distribution. They showed that the tweet rate distribution is asymptotically power law but exhibits a log-normal cutoff over finite sample intervals. They also showed that the intertweet interval distribution for a single user is power law with exponential cutoff. Lu *et al.* [97] developed a method to model the lifetime number of retweets from an originating source, and found the distribution to be a power law with exponent in the range 0.6 to 0.7. They proposed that the probability of being forwarded is proportional to the product of preferential attachment and transmissibility. We provide high-resolution evidence of this relationship over short timescales in Chapter 4, where we examine the decay in retweet rate in the first 24 hours.

Meme propagation research has also contributed to our understanding of information spreading on the internet [88, 164]. Leskovec *et al.* [88] analysed how popularity of memes varies over time and created a model with fluctuations similar to what is observed in real news-cycle data. Our work on retweets focuses on a similar problem, but over much shorter time scales. Memes are likely to be popular for months while Twitter users tend to respond to events or tweets in seconds or minutes.

2.2.2 Relaxation response of a social system

Relaxation response refers to the way a system will return to its original resting state after being excited by some stimulus [35]. From a social media perspective, this may refer to the amount of social media traffic at some given time after an event, and how this level decays.

As Crane and Sornette observed [35], response to events in a social system can be divided into two primary types, *exogenous* and *endogenous*. In the exogenous case, if a network is responding to a large pertubation (e.g. YouTube feature a video on their homepage), the highest rate of activity occurs immediately after the event. Alternatively, in the endogenous case, if a network is responding to small pertubations which spread more slowly (e.g. users sharing a video with other users), the response will grow over time, hit a peak and then decay. Crane and Sornette [35] found that after the initial peak, activity declines as a power law distribution.

In the context of our work in Chapter 3 on news-driven social media activity, we expect retweets in Twitter to be primarily in response to exogenous events. Twitter users have a number of followers who will possibly be aware of a tweet as soon as they log into Twitter. However, there is also an endogenous component to information propagation on Twitter. When users share a tweet by retweeting, further followers will also be exposed to the initial tweet.

Other authors have built upon Crane and Sornette's fundamental work, observing power law decay on social media following peaks of activity. Matsubara *et al.* [105] developed *SpikeM*, a model for the rise and fall patterns of social media influence propagation. This model used a power law to capture the fall pattern after social media spikes, which they claimed was applicable to all classes of social media activity. *SpikeM* requires fitting a high number of parameters, causing it to potentially overfit, and thus reducing its prediction ability. Also, Sadri *et al.* [131] showed that decay in discussions about Hurricane Sandy exhibited a power law decay shape.

Although there exists a significant amount of related work modelling Twitter dynamics, [146], the distribution of retweet times has not been analysed in detail. Hodas *et al.* [70] created a sophisticated method to estimate the probability of whether a user will retweet a given seed tweet. They claimed that the response function was constant for the first two minutes, then dropped as a power law with exponent $\alpha = 1.15$. This analysis was based on fitting a straight line on a noisy data log-log plot, which they showed was roughly linear for the first 10,000 seconds (approximately three hours). Zhao *et al.* [180] performed a similar analysis for retweets from an initial tweet. They plotted retweet times up to 15 hours after the initial tweet and concluded that the observed linear trend on logarithmic axes suggests a power law decay.

As we shall demonstrate in Chapter 4 with more detailed analysis, the rate of retweets has a power law shape for the first three hours, but after this period a simple power law is no longer appropriate. A power law with exponential cutoff provides a better fit to this distribution.

2.2.3 Statistical test for power law

We define power law and other decay functions in Section 2.6.1. Traditional crude methods for detecting whether a dataset is a power law involve binning the data and plotting the bin values against time on a log-log scale, e.g. [3,49]. The slope of the line of best fit, usually determined by least squares fitting, is then taken as the rate of the power law. However, these methods can produce substantially inaccurate results [15] and, in many cases, give no statistical evidence for whether the data is power law distributed.

In 2007 Bauke [15] showed that maximum-likelihood fitting methods are more accurate than methods such as logarithmic binning on the emprical data. We use logarithmic binning only to provide a visual understanding of the distribution of our datasets and use maximum-likelihood fitting methods for all our formal statistical analysis. Following on from the work of Bauke, in 2009 Clauset *et al.* [33] developed a statistical test for determining whether a distribution is a power law. This method uses maximum-likelihood fitting methods to determine the parameter α of the power law, and goodness-of-fit tests based on the Kolmogorov-Smirnov (KS) statistic and likelihood ratios. The basic outline of the Clauset method [33] is as follows :

1. Estimate parameters x_{min} and α of the power law from the empirical data using maximum likelihood estimation.

2. Calculate the goodness-of-fit between the data and the fitted power law distribution using the KS-statistic.

3. Generate a large number of synthetic data sets with power law parameter α and lower bound x_{min}, and calculate the KS-statistics.

4. Compare KS-statistics between the empirical and synthetic data. The p-value is defined to be the fraction of the synthetic KS-statistics that are larger than the empirical KS-statistic.

5. If the resultant p-value is greater than 0.1, a power law is accepted as a plausible hypothesis for the data. Otherwise it is rejected.

We use the Clauset test to determine if our datasets could plausibly be generated from a power law function. The implemetation of the Clauset method for our retweet dataset is outlined in Section 4.3.3. Also worth noting, Alstott *et al.* [5] built upon the work of Clauset to develop *powerlaw*, a Python package for basic fitting and statistical tests for power laws. However, for this thesis we wrote our own code for power law parameter fitting and for most statistical tests.

2.3 Simulating retweet activity and cascades

Here we discuss previous literature related to Chapter 5, *Simulating Retweet Activity and Cascade Size Estimation*.

2.3.1 Causes of power laws in complex systems

Power laws occur frequently in both nature and man-made systems. Examples of phenomena that can be modelled well by power laws include frequencies of words in most languages, sizes of earthquakes, intensity of wars, severity of terrorist attacks, sightings of bird species [33] and citation distributions [23]. Many human activity patterns also exhibit power law distributions [115]. Li *et al.* [91] showed that human correspondence patterns have bursty power law behaviour. Neither the inter-event time nor the response time show Poisson behaviour, as might be expected, but instead both have approximate power law decay.

Doerr *et al.* [44] questioned the applicability of fitting power law distributions to temporal behavioural data, and showed that many processes governing online information spread have a log-normal distribution. They argued that the low exponents found in temporal data militates against preferential attachment, and that while preferential attachment provides an explanation for scale-free degree distributions, it does not provide insight into propagation time distributions. Based on this, they claimed that there does not exist a theoretical model able to explain the observed traces of online human behaviour. This paper considered only preferential attachment as the cause of power laws to online information spread. Although preferential attachment is a common and well-known mechanism for the generation of power laws, it is certainly not the only mechanism.

Power laws can also be caused by sand-pile models, as was shown by Bak *et al.* [9] and also by cascades, shown by Wegrzycki *et al.* [163]. Mitzenmacher [110] and Newman [115] identified 14 causes of power laws, both natural and man-made. Although theoretically possible, many of these causes of power law occur rarely. In the context of our datasets, for our own power law observations on retweets in Chapter 4, it is possible to reject some of these causes quickly. Inappropriate models include random walks, phase transitions, coherent noise [138] or highly optimised tolerance [26]. However, other possible causes of power laws cannot be rejected so readily, and we consider specifically the following three causes:

- Growth by preferential attachment, where new entities attach to existing entities proportional to their current size [13, 14, 76, 100, 178].
- Exponential growth with random observations times (giving the Zipf law where the frequency of an item is inversely proportional to its rank) [126].
- The inter-event time distribution for a single event type where the behaviour that causes events is a consequence of a decision-based queuing process [12, 158].

We discuss these three causes of power laws in more detail below and explain in Chapter 5 why they are candidates for the distribution of retweet times.

Growth by preferential attachment

In preferential attachment, new entities attach to existing entities proportional to their current size. In Polya's Urn model [100] where balls are added to urns with probability proportional to the number of balls in the urn, it can be shown that the number of balls per urn is distributed as a power law. Power laws by preferential attachment occur frequently in nature and in human sciences. Cities tend to grow proportional to their current size [76]. Networks have a tendency to grow by attaching new nodes to those that already have a large number of connections [13]. Preferential attachment is also called the *Yule process*, after it was used to explain the power-law distribution of the number of species per genus of flowering plants [178].

Power law due to combination of exponentials

Combinations of exponentials can lead to power law distributions [115]. Since exponential functions are very widespread in nature and man-made systems, this is a common cause of power laws. As shown by Reed [126], a process that grows exponentially and is sampled at exponentially distributed times, is power law distributed.

Reed argued that the sampling time for distributions should be considered a random variable, and that in many real-world scenarios an exponential distribution is appropriate. For example, the growth of an individual's income may follow geometric Brownian motion. However, the time an individual has been in the workforce until retirement, may follow an exponential distribution. From a set of individuals with the same starting income, the distribution of incomes is a geometric Brownian motion observed after an exponentially distributed amount of time. This leads to a power law distribution of incomes.

Power law due to decision-based queuing process

Barabasi [12] showed that the bursty nature of human behaviour can be explained by a decision-based queuing process, which was further explained by Vazquez *et al.* [158]. When humans execute tasks based on some perceived priority, the waiting time between tasks is heavy-tailed. Consecutive actions from a single user, such as the inter-event times between emails sent, have a tendency to be power law distributed. This is different to the exponential distribution that would occur if human activity was modelled as a Poisson process. Barabasi showed that the timings of five human activity patterns, email and letter-based communications, web browsing, library visits and stock trading, followed non-Poisson statistics.

In Chapter 5 we consider the three aforementioned reasons as possible explanations for the observed power law behaviour in Chapter 4, and deduce whether any of them provides a satisfactory explanation of the phenomena.

2.3.2 Decay of user interest in topics

User interest in topics tends to decay exponentially over time [2,42,90]. Li *et al.* [90] analysed this loss of interest through users' reading history for the purpose of news recommendations. For an exponential decay function, $e^{-\lambda t}$, they assumed the decay parameter λ to be three days, based on a user's reading history. However, the exact decay parameter is dependent on the specific scenario, and attention spans on social media are generally shorter than for other information mediums. Exponential decay forms a component of our explanation of retweet rates in Section 5.2, where we consider user interest in tweets after a period of time. It also forms part of our clustering algorithm, *Social Media Event Response Clustering* (SMERC), in Chapter 6 where we consider the likelihood that tweets are related given their time difference.

Researchers have analysed and modelled specific cases of loss of interest in topics over time [107, 169]. Wu *et al.* [169] modelled the popularity of photos over time, on internet sharing sites such as Flickr, as

$$s = \log_2\left(\frac{r}{d}\right) + 1,$$

where r is the view count of the photo and d is the number of days since the photo was published. The view count to day ratio, r/d, will drop over time, reducing the popularity s of the photo. However, the more widely-used model based on exponential decay is simpler and was found to be more effective for our purposes.

2.3.3 User influence

The *influence* of Twitter users has a strong effect on whether their tweets will be retweeted. However, the concept of influence is difficult to formally define. Cha *et al.* [29] estimated the influence of users by assuming it is comprised of three primary components:

- Indegree influence (the number of followers for a user).
- Retweet influence (the number of retweets mentioning the user's name).
- Mention influence (the number of mentions containing the user's name).

Importantly, they assume the outdegree (number of accounts a user follows) and the number of tweets by a user are not useful measures of influence, as they can be completely controlled by the user. Accounts with high outdegrees and tweet counts have a strong tendency to be bots.

Zhu *et al.* [184] observed that the global distribution of user influence on Twitter has a power law distribution. They attributed this to preferential attachment, with popular users receiving more mentions and retweets, leading to them gaining more followers. This distribution of retweets amongst users has Gini coefficient [27] of $g = 0.9034$, an extremely high value corresponding to the top 1% of users getting more attention than the bottom 99% combined. Information flow in a network is heavily affected by the presence of these highly influential nodes with *super-spreaders* [95] promoting global cascades and *super-blockers* [64] reducing the likelihood of certain types of cascade.

Bakshy *et al.* [11] examined the possibility of cascades through social media structures from *ordinary influencers*. They created a way to predict the influence of users, using user features {# followers, # friends, # tweets, date of joining} along with past influence features {average total influence, minimum total influence, maximum total influence, average local influence, minimum local influence, maximum local influence}. Here *local influence* refers to the direct reshares of social media posts, while *total influence* refers to all reshares, whether direct or indirect. This is an important distinction as for users with a low number of followers, their posts with highest total influence generally occur when a popular user reshares their social media post. A user's influence can be used as a feature to estimate the retweet count.

Having a larger number of followers increases the likelihood that a seed tweet will be retweeted. However, the follower count is not always an accurate measure of a user's influence. Many accounts have artificially inflated numbers of followers due to bots or other techniques such as trading followers [29]. A smaller number of active followers is more impactful and leads to more retweets than a larger number of dormant followers.

2.3.4 Retweet cascade size estimation

We first define a retweet cascade as was done by Vosoughi *et al.* [159], as an unbroken series of retweets from an initial tweet. The size of a retweet cascade is the total number of retweets in this cascade.

The intensity of Twitter activity is often modelled as a self-exciting temporal point process [180,185]. Events and other tweets tend to stimulate other activity, leading to burstiness. The rate of twitter activity is also affected by other factors such as human prioritisation of tasks and diurnal rhythms [55,103].

Estimating the size of a cascade of retweets is a popular research topic, with many different approaches and varied problem formulations [60,83,97,180]. When discussing retweet cascades we use the notation and metrics introduced by Zhao *et al.* [180] with R_∞ denoting the total number of retweets, and the prediction of this value made at time t by $\hat{R}_\infty(t)$.

Kupavskii *et al.* [83] predicted the size of a cascade based on the initial spread using a gradient-boosted decision-tree model. They used social features such as the number of followers, friends and favourites of the user. Content features used included tweet length, number of mentions and hashtags within the tweet. They analysed time-sensitive features such as retweet ratios up to a given time, as well as features of the *infected* nodes (which have been influenced by the information spread) up to the given time. For their experimental setup, they used a training window $[T_0, T_f]$ where T_0 took values of 0, 15 and 30 seconds, and T_f took values of 4 minutes, 15 minutes, 1 hour, and 1 week.

A particularly relevant model for retweet cascade size estimation is the Dynamic Poisson model (DPM) [1] which estimates the retweet rate λ_t as

$$\lambda_t = \lambda_{t_{peak}}(t - t_{peak})^{\gamma}, \tag{2.1}$$

where t_{peak} is the time of peak retweet rate, and γ is the power law parameter. Our model in Section 5.4.3 is similar to Equation (2.1), except we use a power law with exponential cutoff instead of a power law.

SEISMIC (A Self-Exciting Point Process Model for Predicting Tweet Popularity) was a method developed by Zhao *et al.* [180] to estimate retweet information cascades. It uses the theory of self-exciting point processes to develop a statistical model for predicting retweet counts. For the error metric, SEISMIC used absolute percentage error, $\mathrm{APE}(t)$, of the estimation for tweet w at time t after the tweet was posted, given by

$$\mathrm{APE}(t) = \frac{|\hat{R}_\infty(t) - R_\infty|}{R_\infty}. \tag{2.2}$$

Zhao *et al.* [180] also provided a publicly available dataset for testing, which we use in Chapter 5. They collected 3.2 billion tweets from October 7 to November 7, 2011, and kept only tweets with no less than 50 retweets, no hashtags in the text, and written in English. There were 166,076 tweets that met these requirements. One issue with using this dataset is the potential skewing of the optimal cascade size prediction. For example, suppose we have an initial seed tweet from a user with 200 followers and over the first hour it has a total of one retweet. Based on this data, the best possible prediction of the total number of retweets would be less than five total retweets. However, if the data is contained in the SEISMIC dataset, our best guess for the total cascade size would be at least 50 retweets.

Analysing the SEISMIC dataset manually revealed some anomalies in the data. For example, there is a tweet from a user with over 100,000 followers that does not receive any retweets in the first 30 minutes, followed by hundreds of retweets in the next 30 minutes. It's extremely unlikely that such a phenomena would occur, suggesting that at least part of the dataset has been corrupted. We filtered out all tweets where such data corruption appeared to have occurred.

Wu *et al.* [170] provided a method to estimate the cascade size from tweets from news sources, as is our objective in Chapter 5. However, they did not make their datasets or code public, and the paper is insufficiently specific to accurately replicate their approach. They evaluated their method by calculating the correlation between their prediction vector and the actual tweet count vector, which is not frequently used to evaluate the performance of a prediction method [73].

2.4 Automated microblog summarisation and event detection

Here we discuss previous literature related to Chapter 6, *Event Detection and Time Estimation from Twitter*.

2.4.1 Social sensing and microblog summarisation

Social media is a technological tool which provides researchers rich data for social science research. It facilitates social sensing, using crowd response to better understand events in near real time. The progress in our understanding of social sensing has coincided with the rapid growth in social media since the early 2000s [144]. To fully understand social sensing, it is necessary to have accurate models for information flow in large scale social networks [176]. The state of social sensors can be modelled as a probabilistic function of their neighbours, which slowly evolves over time [144]. In addition to posts on social media, other information can be incorporated into social sensing systems, such as ratings on hotel or restaurant review sites, website usage statistics and public social media profile information.

Microblog summarisation refers to techniques to automatically summarise sequences of events from microblogs as they occur in real time [136]. The most common technique overall for automated microblog summarisation is measuring keyword frequency [31, 82, 136], detecting keywords that are used more frequently than expected in a period of time. *TweetMotif* [82] uses this technique, with input tweets tokenised, topics filtered and then scored based on their relative frequency, then merging equivalent topics. Sharifi *et al.* [136] developed the *phrase reinforcement algorithm*, a graph-based method using relative frequency of keywords to determine key topics. Chakrabarti *et al.* [31] developed *SummHMM*, a hidden Markov model based method detecting *bursty* keywords. They used knowledge of previous response patterns, at sporting events for example, to better detect future events.

Our work in Chapter 6, creating a technique to automatically summarise key events and their corresponding times from a social media stream, fits into the categories of both microblog summarisation and social sensing. Compared with previous work, we focus more heavily on the temporal relationship between events and response times.

2.4.2 Event detection

Event detection on Twitter is the ability to detect key events of interest, such as earthquakes or soccer goals. There exist many methods using Twitter to do this, particularly focusing on sporting events [59, 147, 181]. Zhao *et al.* [181] used Twitter response to events in NFL games to identify key events such as touchdowns, interceptions, fumbles and field goals. To determine the nature of events, they measured whether the frequency of content-based keywords was above a pre-defined threshold. The temporal component of their event detection method was based on the rate of relevant posts in specified time windows, using the fact that social media activity increases heavily after key events.

A sequence of tweets constitutes a time series, so the problem of detecting events from Twitter is often transformed into a time series clustering problem [7]. Yang and Leskovec [177] summarised two key components for time series clustering in online media, a distance measure and a clustering algorithm. The most commonly used distance measure is Euclidean distance, which has been used in a variety of works [57]. More advanced measures include *dynamic time warping* [17] and *longest common subsequence* [80]. The most common clustering algorithm continues to be *k-means* [99] despite the limitations of having to specify the number of clusters beforehand, and being sensitive to the starting point. Tweet clustering is often focused on using tweet content and user features, but has limitations based on the amount of information contained in a tweet. Studies typically use either *textual* features such as tracking the number of pre-defined keywords or hashtags, or purely *temporal* features such as the timing of posts.

Similar to techniques for microblog summarisation, several authors have created methods to detect events based on bursty keyword usage. Mathioudakis *et al.* [104] conducted tweet clustering by monitoring keyword frequency, looking for *bursty* keywords. *Twevent* [89] extracts continuous and non-overlapping word segments, and then calculates bursty event segments within a fixed length window. To evaluate their method, the authors used the metrics *precision*, the proportion of detected events related to realistic events, and *recall*, the proportion of realistic events detected from the data set. Due to the difficulty in assessing whether or not a particular tweet, or cluster of tweets, is related to an event or what constitutes a realistic event, such definitions can be hard to measure on real datasets. *Topicsketch* [173] uses a more sophisticated method to bucketize and hash tweets and claims to be able to detect events from shorter bursts than Twevent. Both these methods register events only if there is a bursty keyword response, while smaller events won't be detected.

With most similar objectives and datasets to our tweet clustering work in Chapter 6, Gillani *et al.* [59] developed a way to identify key events in sporting contests by clustering both temporal and textual features. They used a threshold technique to determine whether an event is important, and incorporated the time of posts as a feature, by appending it to a vector of word counts collected within a window. As this method uses k-means for clustering and it is not possible to know the number of key events beforehand, it is perhaps more suited to the problem of *post-hoc* microblog summarisation rather than real-time event detection. Conversely, as we show, our approach in Chapter 6 is more suited to real-time tweet clustering and event detection; by reasoning more probabilistically about the distributions of times between tweets around events, we develop a mathematical model for incorporating temporal information into tweet clustering, instead of simply appending it as part of a feature vector.

Evaluating the performance of approaches for event detection is a known problem, extensively discussed by Atefeh *et al.* [7]. For information retrieval tasks, precision, recall and F-score are commonly used performance metrics, but have the problem of being extremely time-consuming or infeasible to manually calculate. Recall calculation, in particular, requires the manual identification of all events in a large noisy dataset, where it is not always well-defined what constitutes an event. Many works calculate *precision@K* instead of precision, the fraction of correctly detected events out of the top K detected events. Doing this reduces the computational burden and increases precision scores, but often prevents direct comparison between the performance of methods. In addition, Atefeh *et al.* [7] noted the need for public benchmarks to evaluate the performance of different approaches for event detection. We discuss issues of validating methods further in Chapter 6.

2.4.3 Similarity and distance measures

Metzler *et al.* [108] outlined the difficulties in creating similarity measures for small segments of text. They discussed different ways of representing a text string as a stemmed representation in which each word is broken down to its stem, which aids improvement in vocabulary matching. They also gave an alternative representation, in which words are expanded to give their full meaning. For example, text processing is more effective when *Bank of America* and a *river bank* can be distinguished. Google researchers led by Mikolov developed *word2vec* [109], a system of assigning each word to a vector, typically of hundreds of dimensions. The system is designed so that words with a similar meaning are located in close proximity in the vector space. This system allows vector operations on words, for example, King + Woman - Man = Queen. There have also been multiple attempts to extend word2vec to tweets [41,160].

Despite the usefulness of these methods, a drawback is how to deal with words that are not in the dictionary. Our work deals with topics such as sport where the context and meaning of words can differ from other situations. Without dictionaries dedicated to these topics, text vectorisation methods such as word2vec are potentially less useful.

Cha [30] conducted a thorough study of 45 methods to find the distance or similarity score between two vectors. They split the measures into nine categories, which we give in Appendix D.1. Importantly, there is the Minkowski family, involving a norm of some kind, and the inner product family, which considers positive matches only, giving the *closeness* between any pair of tweets. For our purpose this emphasizes the number of common words between tweets. Cosine similarity normalises the measure based on the tweet length, giving a score between 0 and 1. While discussing similarity measures in their survey of clustering algorithms, Xu *et al.* [174] identified that cosine distance is the "most commonly used distance in document area" [sic]. Compared to other measures, such as Euclidean distance, this method emphasizes the impact of words that tweets have in common.

In Chapter 6, we use cosine similarity, which geometrically is the cosine of the angle θ between the vectors $\mathbf{A}$ and $\mathbf{B}$ representing the tweets,

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}}. \tag{2.3}$$

2.4.4 Clustering methods

Jain and Dubes [77] gave the following definition for clustering:

- Instances in the same cluster must be as similar as possible.
- Instances in different clusters must be as different as possible.
- Measurement for similarity and dissimilarity must be clear and have practical meaning.

Xu *et al.* [174] split clustering algorithms into 18 categories, 9 traditional and 9 modern, recorded in Appendix D.2. We use *affinity propagation* [52] as it gives the best performance of all algorithms meeting our requirements, discussed in Chapter 6.

Affinity propagation regards all data points as potential centers of clusters. Messages are recursively passed across the edges of the clustering network until a good set of clusters emerges. The algorithm proceeds by alternating two message-passing steps, updating the *responsibility* matrix, which quantifies how well each cluster center is suited to serve as the exemplar, and updating the *availability* matrix, which takes into account other points' preference for cluster centers. The iterations continue until the cluster boundaries remain unchanged over a specified number of iterations, indicating that convergence has occurred. Affinity propagation does not require the number of clusters to be specified beforehand, an important advantage over other clustering algorithms such as k-means.

2.4.5 Natural language processing

Natural language processing is the use of computers to process and analyse large amounts of natural language data [74]. It began in the 1950s with statistical or rule based approaches, but in the late 1980s, the use of machine learning algorithms became popular. More recently, in the 2010s, neural network approaches began to achieve state-of-the-art results in many natural language tasks [61].

Stop words

The concept of stop words was first used by Luhn [98] in 1959 in an attempt to isolate parts of a document containing intelligible information. Luhn originally used 16 basic stop words such as *a*, *of* and *the*; as the concept has developed, a greater number of stopwords have been identified. The idea is that these connector words provide minimal information about the substance of a document, and removing them prevents documents being associated based on the use of these words. For example, it may be erronous to claim that two documents were similar because they both had a similar frequency of the word *the*. In our work we use the Natural Language Toolkit (NLTK) [96] set of English stopwords, containing 153 commonly-used English words. With this stopword list and a standard body of text, about 25% of words are removed as stopwords.

Word stems

Words such as *agree*, *agreeing*, *agrees* and *agreed* have the same stem but different affixes. For the purpose of comparing texts, we consider these all as the same word. To do this, each of the words in a tweet is converted to its stem. Sharma *et al.* [137] performed an extensive study on stemming algorithms and classified them as either being a rule-based approach or a statistical approach. Rule-based stemmers are faster and well suited for English, but have the disadvantages of being time-consuming to create, not being able to handle additional grammatical information, and having a tendency to over-stem in certain situations. The popular *Porter Stemmer* algorithm [79], a rule based stemmer which we use for our text processing, reduces the vocabulary to around one third of its original size.

Bag of words

There are many proposed methods to summarise or "embed" words as vectors. For example, *word2vec* [109], *tweet2vec* [41, 160] and *sentence2vec* [86] are recent methods to convert words, tweets and sentences to vectors respectively. These methods are particularly effective when the structure of a sentence is important, not just the contained words [86]. Alternatively, there are modern and effective ways to conduct natural language processing using *recurrent neural networks* [149].

An older and less sophisticated method to create a vector from a set of text is the *bag-of-words* model [67], where the vector records the number of occurrences of each word in a text. For our tweet data, we found that using the bag-of-words model was more effective than the methods based on word2vec. This is likely because we are attempting to group tweets about selected topics, without being overly concerned about the structure of the tweet. Also, many of our datasets are about sporting events, which are a particularly unstructured category of tweets and potentially less suited to methods to understand their structure.

TF-IDF

Term Frequency - Inverse Document Frequency (TF-IDF) [133] weights the importance of words in a document based on a combination of uniqueness and frequency. It works on the theory that words which occur sparsely are more meaningful in detecting content. For example, suppose in a corpus the word *hippopotamus* occurred in two different documents, but not again in the entire corpus. It would appear likely that these two documents are related. Conversely, if we consider the situation where a word such as *hippopotamus* was occuring in half of the documents, it is likely that the entire corpus is about hippopotamuses and we would be less confident that two documents with the word *hippopotamus* would be related.

TF-IDF is defined by

$$\text{TF-IDF} = TF_t \times \log \frac{N}{DF_t}, \tag{2.4}$$

where TF_t is the frequency of the term t, N is the number of documents in the corpus, and DF_t is the document frequency of t. There are several popular ways to implement the calculation of TF-IDF depending on edge conditions, we use the default *TfidfVectorizer* implementation from scikit-learn [122].

2.4.6 Event time estimation

Event time estimation from social media has received little attention relative to event detection or microblog summarisation. This is likely due to the lack of publicly available datasets and the time consuming task of creating such datasets. Previous literature [132, 181] referred to the first time that events are mentioned on social media, with the implicit assumption that the event occurred some distribution of time prior to this point. Zhao *et al.* [181] found that on average it takes 17 seconds for a Twitter user to report an NFL game event. However, to the best of our knowledge, there have been no attempts to precisely estimate event times from the associated Twitter response.

Our event time estimation method is based on fitting an appropriate response distribution to available tweet times, and then calculating an offset from the distribution intercept. Consequently we must find a distribution that fits our datasets closely, can be used to create a method for event time estimation, and preferably has an explanation consistent with the underlying processes. We consider the use of both the log-normal and Weibull distributions.

2.4.6.1 Causes of log-normal distribution

Log-normal distributions, defined in Section 2.6.1, occur when the growth over a time step is a normally distributed random factor that is independent of the size. For this reason log-normal distributions frequently occur in natural and man-made systems. Voss [161] found that article sizes on Wikipedia are approximately log-normally distributed. They claimed that the likely cause of this is percentage steps of growth, with individual article sizes converging to some *perfect size* which is influenced by topic. This finding was confirmed and extended by Serrano *et al.* [135], who found that in addition to Wikipedia, this result also held for two other collections of web pages.

Sobkowicz [139] analysed the distribution of internet comment lengths, finding that they were also log-normally distributed. However, they asserted that subsequent posts in a discussion are not modifications of each other, with almost 99% of comments original. Consequently they claimed that the model of successive modfication of entries could not be used to explain the observed log-normal distribution.

As we discuss further in Chapter 6, we consider the log-normal distribution as a candidate distribution for the temporal response of tweets from events.

2.4.6.2 Causes of Weibull distribution

Weibull distributions, defined in Section 2.6.1, were originally developed to model particle sizes of powdered coal [129]. However, they are more commonly used in industry for failure analysis [179]. An item with a decreasing, constant, or increasing failure rate will potentially have a Weibull lifetime distribution.

The purpose for which we consider using the Weibull distribution in Chapter 6, the time between events and Twitter response, is quite different to its primary use in industry for failure analysis. However, there have been a few instances of authors using Weibull distributions for online information spread. A Weibull distribution was used by Lande *et al.* [125] to model the number of *likes* of a Twitter message. They used a stochastic process to simulate online information spread, which produced a distribution of likes with the same shape as a Weibull distribution. Also, Jiang *et al.* [78] used a Weibull distribution to model the inter-call durations from a Chinese mobile phone operator. Although they did not explain the underlying mechanics that would cause a Weibull distribution, they found that the tail shape of the Weibull was a closer match to their dataset than either an exponential or power-law distribution.

Similar to the justification used by other authors, the early peak and long tail shape of the Weibull density function is a close match to our observed distributions, particularly when fixing the Weibull shape parameter k between 1 and 2. We thus also consider using the Weibull distribution in Chapter 6 as a candidate distribution for the temporal response of tweets from events.

2.5 Diurnal cycles and adjustment

A diurnal cycle describes the way system behaviour varies throughout the day. For example, people are much more likely to use social media during waking hours, compared to the early hours of the morning when most of the population are sleeping. In addition to daily cycles of activity, human internet usage also exhibits weekly trends, as the population has similar overall tendencies at the same times of the week [143].

Gao *et al.* [56] developed a way to adjust for the daily cycles in social media activity with a concept called *Weibo time*, which is measured in the number of messages that users post on Weibo at that time. They then did an arithmetic conversion from the Chinese time zone GMT+8 to Weibo time to implement their diurnal adjustment.

In this thesis, we use diurnal cycles in Chapters 3 and 4 when looking at how tweet, retweet and news rates vary throughout the day. We make diurnal adjustments to remove the influence of the hour of the day on the overall tweet activity level. Our method has similar goals to Gao *et al.* [56] but using our own techniques. One particular benefit of our probabilistic approach is giving a discrete output (an integer number of tweets) which facilitates follow-on analysis. We also observe that a neural network can effectively *learn* a diurnal cycle.

2.6 Definitions, tools and techniques

Here we outline mathematical and statistical definitions, tools and techniques which are used throughout the thesis.

2.6.1 Decay functions

We use decay functions extensively, to model the distributions of retweet times in Chapters 4 and 5, and the distribution of tweets in response to events in Chapter 6.

Definition 1. *A power law is a distribution that has probability density function*

$$p(x) = Cx^{-\alpha}, \qquad C = \frac{\alpha - 1}{{x_{min}}^{1-\alpha}}, \tag{2.5}$$

with $C, \alpha > 0,\ x \geq x_{min}$.

Importantly, a power law exhibits a *heavy tail*. Compared with functions that decay exponentially, much more of the distribution of the power law is contained at higher values of x. A power law has a well-defined mean only if $\alpha > 2$, and it has a finite variance only if $\alpha > 3$.

Definition 2. *A power law with exponential cutoff is a distribution that has probability density function*

$$p(x) = Ax^{-b}e^{-cx}, \tag{2.6}$$

with $A, b, c > 0,\ x \geq x_{min}$.

In constrast to a simple power law, a power law with exponential cutoff always has a well-defined mean and finite variance.

We use the power law and power law with exponential cutoff in Chapters 4 and 5 to model the distribution of retweet rates.

Definition 3. *A Weibull distribution is a distribution with probability density function*

$$f(t;\lambda,k) = \begin{cases} \frac{k}{\lambda}\left(\frac{t}{\lambda}\right)^{k-1} e^{-(t/\lambda)^k} & t \geq 0, \\ 0 & t < 0, \end{cases} \tag{2.7}$$

where $k > 0$ *is the shape parameter and* $\lambda > 0$ *is the scale parameter of the distribution.*

Definition 4. *A random variable X with log-normal distribution has probability density function*

$$f(t;\mu,\sigma) = \frac{1}{t\sigma\sqrt{2\pi}} e^{-\frac{(\log t-\mu)^2}{2\sigma^2}}, \tag{2.8}$$

where μ *and* σ *are the mean and standard deviation, respectively, of X's natural logarithm.*

We use the Weibull and log-normal distributions in Chapter 6 to model the Twitter response after an event.

2.6.2 Methods to measure error

We measure the performance of our prediction methods using metrics *relative error, mean absolute error* (MAE), *mean absolute percentage error* (MAPE) and *median absolute percentage error* (MdAPE).

Definition 5. *For a value of interest* τ, *the relative error is given by*

$$\mu = \left| \frac{\tau_{predicted} - \tau_{actual}}{\tau_{actual}} \right|. \tag{2.9}$$

The relative error is not defined if the actual value, τ_{actual}, is zero.

Frequently used in assessing prediction performance [73, 167], mean absolute error measures the difference between two paired variables X and Y. One key advantage of mean absolute error is the easy interpretation: it is simply the average distance between values. However, it has the disadvantage of not being scaled based on the magnitudes of the values.

Definition 6. *Given two sets of values $X = \{x_i\}$ and $Y = \{y_i\}$ for $i = 1, \ldots, n$, the mean absolute error (MAE) is given by*

$$MAE = \frac{\sum_{i=1}^{n} |x_i - y_i|}{n}. \tag{2.10}$$

When we wish to scale the errors by the magnitude of the values being predicted, we use mean absolute percentage error (MAPE).

Definition 7. *Given a set of actual variables $\{A_i\}$ and a set of predicted variables $\{P_i\}$ for $i = 1, \ldots, n$, the mean absolute percentage error is given by*

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{A_i - P_i}{A_i} \right|. \tag{2.11}$$

Although easily interpretable, this metric has several disadvantages [73, 151]. First, it cannot be used for zero values, as this would lead to division by zero. Also, MAPE is biased, because if the forecast is too low, the maximum percentage error is 100%, but if the forecast is too high the maximum percentage error is unbounded. Consequently, when MAPE is used to optimise a model it tends to prefer a model that forecasts lower than actual values [151]. In this thesis we generally optimise our models using MAE as it is an unbiased metric.

Another commonly used metric is the median absolute percentage error (MdAPE). This is less heavily affected by outlier values than MAE or MAPE, which can be either an advantage or a disadvantage depending on the application.

Definition 8. *Given a set of actual variables $\{A_i\}$ and a set of predicted variables $\{P_i\}$ for $i = 1, \ldots, n$, the median absolute percentage error is given by*

$$MdAPE = median \left(100 \left| \frac{A_i - P_i}{A_i} \right| \right). \tag{2.12}$$

2.6.3 One-hot encoded variables

One-hot encoded variables are a way of representing categorical variables in a binary form which allows machine learning algorithms to do a better job of prediction. The concept of one-hot encoding originated in electrical engineering [66], but is now regularly used in machine learning. For example, consider the make of a car which could take the possible categorical values {Ford, Toyota, BMW, Mercedes}. In order to input these variables into a machine learning system we need to use a numeric encoding. However, there is no logical ordering of these variables so an encoding such as {0, 1, 2, 3} would potentially cause poor prediction in a machine learning system. A preferable system of encoding is to use *one-hot encoded variables*, splitting the variable into four binary variables representing whether the car is each of the selected brands. As cars have exactly one make, for each car exactly one variable will take the value 1 while the other three will be 0.

2.6.4 Granger causality

In 1969, Clive Granger proposed the *Granger causality test* [62] to determine whether one time series is useful in forecasting another. Traditional statistical tests that determine whether two variables are correlated do not provide information about which variable is causing changes in the other. The test was proposed for economics but has been extended, often contentiously [63], to other areas.

Loosely speaking, a time series X Granger-causes Y if X values provide information about future values of Y. We use the definition of Granger causality from [118].

Definition 9. *Let $X = \{X_t,\ t \in \mathbb{N}\}$ and $Y = \{Y_t,\ t \in \mathbb{N}\}$ be stationary time series. Suppose we regress variable Y_t on its own past values and past values of X_t as follows:*

$$Y_t = \alpha_0 + \alpha_1 Y_{t-1} + \ldots + \alpha_p Y_{t-p} + \ldots + \beta_1 X_{t-1} + \ldots + \beta_p X_{t-p} + \epsilon, \tag{2.13}$$

where α_i and β_i are fitted values while ϵ is the error. Our null hypothesis for no Granger causality is that $\beta_1 = \beta_2 = \cdots = \beta_p = 0$. If this null hypothesis is rejected, X Granger causes Y.

In Equation (2.13), p is the maximum lag value. We test for Granger causality between news and tweet volumes in Chapter 3. Note that Granger causality is much "weaker" than true causality, as even if a variable X Granger-causes Y we cannot conclude that X actually causes Y.

2.6.5 Linear Regression

Linear regression is a technique to model the relationship between two variables by fitting a linear equation to given data. Suppose we have datasets X and $\mathbf{y}$ given by

$$X = \begin{pmatrix} \boldsymbol{x}_1^\top \\ \boldsymbol{x}_2^\top \\ \vdots \\ \boldsymbol{x}_n^\top \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \tag{2.14}$$

where n is the number of samples and p is the number of predictors. Linear regression assumes that $\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}$ where

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix} \quad \text{and} \quad \epsilon_i \sim N(0, \sigma^2). \tag{2.15}$$

When fitting a regression model, we find $\boldsymbol{\beta}$ in order to minimise the L_2-norm $||\boldsymbol{\epsilon}||_2$. Due to its simplicity, linear regression is used extensively in practical applications [175].

We use linear regression in Section 3.5 as a method to predict news activity and tweet activity.

2.6.6 Neural networks

A *neural network* is a non-linear function that maps real-valued vector inputs to real-valued outputs. The vector input is transformed through a series of hidden layers consisting of a set of neurons, which receive a signal from connecting nodes, process it, and then signal adjacent nodes. In recent years, neural networks have revolutionised many fields including computer vision [92,93], machine translation [50,145] and social network filtering [39].

A *multilayer perceptron* is a class of feedforward neural networks [68], artificial neural networks where connections between nodes do not form a cycle. Multilayer perceptrons consist of at least three layers (including at least one hidden layer) of

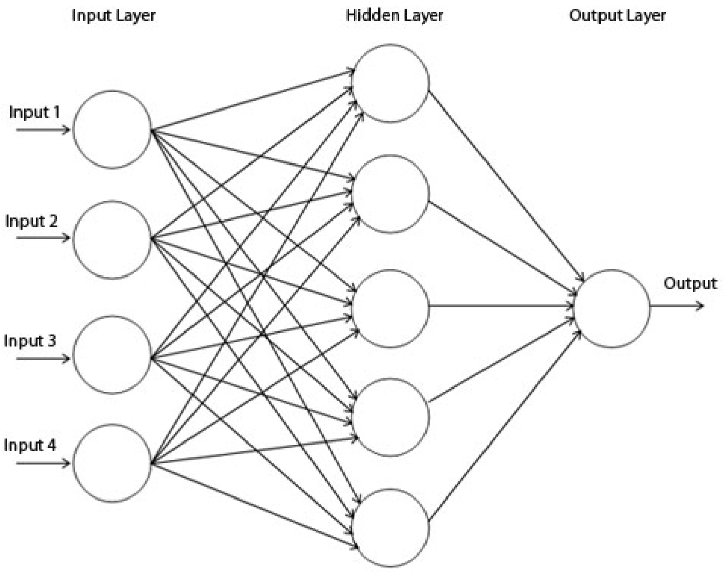


Figure 2.1: Example of a multilayer perceptron with four input variables, one hidden layer with five nodes and one output variable.

fully-connected nodes with non-linear activation functions. As was proven by Cybenko [37], multilayer perceptrons are universal function approximators, so they can be used effectively for regression analysis. An example of a multilayer perceptron is shown in Figure 2.1.

Each node in the neural network consists of a linear combination of the previous layer, fed into an *activation function*. Commonly used activation functions include the sigmoid function or the Rectified Linear Unit (ReLU) function. Let $\mathbf{a}^{[0]} \in \mathbb{R}^N$ be the input to the neural network. We have

$$\mathbf{z}^{[\ell]} = W^{[\ell]}\mathbf{a}^{[\ell-1]} + \mathbf{b}^{[\ell]}, \tag{2.16}$$

with

$$\mathbf{a}^{[\ell]} = \sigma(\mathbf{z}^{[\ell]}]), \tag{2.17}$$

for $\ell \geq 1$, where $W^{[\ell]}$ are the weights of the ℓth layer, $\mathbf{b}^{[\ell]}$ is the linear offset of the ℓth layer, σ is the ReLU activation function [113], $\mathbf{z}^{[\ell]}$ is the output of the ℓth layer before activation, and $\mathbf{a}^{[\ell]}$ is the output after activation. For the final layer in our neural network, we do not use an activation function as we are predicting a real output.

Training the neural network in order to determine the weights $W^{[\ell]}$ requires a loss function to be selected. The best choice of loss function depends on the application, and a common choice for regression is the *mean-squared error*

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2, \tag{2.18}$$

where n is the number of samples, $\hat{Y}_i$ are the predicted values and Y_i are the known values.

One key benefit of neural networks is the ability to model complex relationships with minimal human input required. They work best with very large amounts of input data and extensive computational resources.

Neural networks are trained using backpropagation. During the training phase, inputs are organised in batches and fed into the network, giving a loss result at the output layer. In the backwards pass, the derivative of the error is passed back through the layers of the network, with the gradients for all the learnable weights in the network computed using the chain rule. The weights are then updated using an algorithm such as *stochastic gradient descent*, with a tailored *learning rate* parameter. This process is very computationally intensive and is well suited to being conducted on Graphical Processing Units (GPUs).

Disadvantages of neural networks include the computational resources and time required for training, and also the tendency of overfitting caused by the very high number of weights. The overfitting problem can be minimised by using a technique called *dropout*, where each neuron is dropped from the net with constant probability during the training phase, and then during test phase all neurons are used for prediction. Another disadvantage of neural networks is the tendency to act as a *black box*, not necessarily giving insight into the underlying mechanisms that are being modelled.

We use neural networks in Section 3.5 to predict news activity levels from tweet activity levels, and vice versa.

2.6.7 *k*-fold cross-validation

The technique of *k-fold cross-validation* is used to assess the performance of a prediction method on an independent dataset [16, 81]. The original sample is split into k

equal-sized subsamples. A total of k experiments are run on the dataset. For each experiment, one of the k subsamples being used as the validation set and the other $k-1$ subsamples as the training set. A key advantage of this method is that all observations are used for both training and validation, with every observation used exactly once for validation.

We use k-fold cross-validation in Section 3.5, where we predict tweet activity from news activity, and vice versa.

2.6.8 Savitzky-Golay filter

A *Savitzky-Golay filter* [134] is a method for smoothing digital data points. Given a set of n ordered pairs $\{x_j, y_j\}$ and filter of length m where m is an odd integer, we generate output $\{Y_j\}$ with convolution coefficients C_i according to

$$Y_j = \sum_{i=-(m-1)/2}^{i=(m-1)/2} C_i y_{j+i} \quad \text{where} \quad \frac{m+1}{2} \leq j \leq n - \frac{m-1}{2}. \tag{2.19}$$

A Savitzky-Golay filter has the effect of fitting a low-degree polynomial to the surroundings of each data point. The best choices of the degree of the polynomial and the window size (the number of points surrounding the targeted data point) depend on the raw data. A Savitzky-Golay filter is used to reduce noise in a signal, but also has the potentially unwanted effect of distorting the data by reducing the peak heights. The extent of distortion and improvement in noise reduction both increase with the degree of the polynomial and the width of the filter.

We use a Savitzky-Golay filter in Section 4.2.5 to smooth our diurnal cycle for Twitter activity.

2.6.9 Testing goodness of fit

Given a dataset and a proposed distribution, we often wish to know whether a distribution is a good fit to the data. To do this we use the *Kolmogorov-Smirnov (KS) statistic* to measure the distance between the empirical distribution and the hypothesised model. For a theoretical distribution, $F(x)$, and an empirical cumulative distribution function (CDF), $S_n(x)$, the Kolmogorov-Smirnov statistic, D_n, is defined by

$$D_n = \sup_x |F(x) - S_n(x)|. \tag{2.20}$$

We use the Kolmogorov-Smirnov statistic in Sections 4.3.3 and 6.4.3, where we analyse the quality of distribution fits.

2.6.10 Model selection criteria

In order to evaluate the suitability of models and the number of parameters they contain, we consider the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC).

The AIC is given by

$$\text{AIC} = 2k - 2\ln(L), \tag{2.21}$$

where k is the number of parameters in the model and L is the likelihood function. Functions with lower AIC scores are preferable, so we wish to find the function with the highest likelihood score yet with the lowest number of parameters.

An alternative measure of model fit, the Bayesian Information Criterion (BIC) is given by

$$\text{BIC} = k\ln(n) - 2\ln(L), \tag{2.22}$$

where L is the likelihood function, k is the number of free parameters, and n is the number of observed datapoints. Compared with the BIC, the AIC penalises the number of parameters less strongly.

2.6.11 Bootstrapping

Bootstrapping [46] is used to estimate confidence intervals by random sampling with replacement. Suppose we have sample values $X = \{x_1, x_2, \ldots, x_n\}$. We resample these values, giving $Y_j = \{y_{j1}, y_{j2}, \ldots, y_{jn}\}$, where each y_{ji} is drawn from X with replacement. This is then repeated m times, giving a set of bootstrapped samples $\{Y_j\}$. From each bootstrapped sample Y_j, a measurement of interest will be made. We then have a set of measurements from which we can estimate standard errors and confidence intervals.

The key purpose of bootstrapping is determing the likely error in a prediction, particularly in cases of limited data. Ideally we would like to have multiple samples from the same population. However, when this isn't possible, bootstrapping is an alternative way of determining the underlying variability caused by a limited sample size.

We use bootstrapping in Section 6.4.5 to estimate the error of our method to estimate the time of events based on the Twitter response.

Chapter 3

The Temporal Relationship Between Tweets and News

3.1 Introduction

The expansion in popularity of social media has changed the way people receive news [84]. Young people in particular are far more likely to check their Facebook and Twitter feeds than to read a newspaper or watch an evening news television program [148, 172]. Researchers have questioned the role that social media plays in setting the news agenda [28]. Most major news organisations tweet news articles when they are published [6], therefore it is possible to monitor the activity of news organisations by monitoring their Twitter accounts. As Twitter has a public API [153], this process can be automated, providing an efficient method to monitor current news stories. We discuss prior work on the relationship between the news and social media in Section 2.1.

Human activity on Twitter tends to be in response to some stimulus, whether it be watching television, from other tweets, or directly observing events in person [72]. Researchers can monitor human Twitter activity about certain topics by recording the frequency of tweets mentioning selected keywords related to these topics. One challenge this poses for researchers is the difficulty of knowing the keywords that will be used for future news events. For example, if there is a shooting, news organisations will report the event and people will tweet using the keyword "shooting". However, as researchers, we will not know in advance that this shooting event is going to happen and consequently will not know what keywords to collect.

A way to overcome the problem of not knowing the news in advance is to collect tweets on entities that are regularly featured in the news. One of the best examples is politicians, who regularly generate news stories on a daily basis, particularly during a political race. When Twitter users discuss famous individuals, they often do so by mentioning their Twitter username. As an example, the President of the United States, Donald Trump, has Twitter username @realDonaldTrump.

In this chapter we analyse the relationship between rates of Twitter activity and news about selected famous individuals. In Section 3.2 we outline our data collection methodology and introduce our datasets. In Section 3.3 we give a visual representation of the diurnal cycles of public tweet activity, and how intervals of higher public tweet activity coincide with higher tweet and news counts. We quantify these observations by measuring how the daily news and public tweet counts vary over the day. In Section 3.4 we show that Twitter activity Granger-causes news activity, but not the reverse, providing evidence that Twitter is, on average, a faster responding medium to events than news. Through a novel diurnal adjustment method, we outline a way to conduct automatic event detection using Twitter. When the rate of Twitter activity exceeds a typical value depending on the hour of the day, we can automatically trigger an alert indicating that some significant event has occured. Using diurnal adjustment overcomes the changes caused by the daily Twitter cycle and allows detection of key events at times of lower social media activity, such as in the early hours of the morning. This event-detection method has potential application for news organisations, who wish to know as soon as possible when there is breaking news.

Using the observations about Granger causality as motivation, in Section 3.5 we explore prediction of the current public Twitter activity given other variables such as the recent news activity. This chapter focuses on temporal aspects of social media activity, therefore we deliberately do not use the text content of tweets when investigating causality or making predictions about future tweet volumes. We focus in detail on the text content of tweets, and why using both textual and temporal information is essential for tweet clustering and event prediction, in Chapter 6. This chapter fits into the thesis structure as indicated in Figure 3.1.

This chapter makes the following key new contributions:

- Showing that tweet rates Granger-cause news activity rates, but not vice versa.
- Developing a method to adjust for the diurnal cycle in Twitter activity rates.

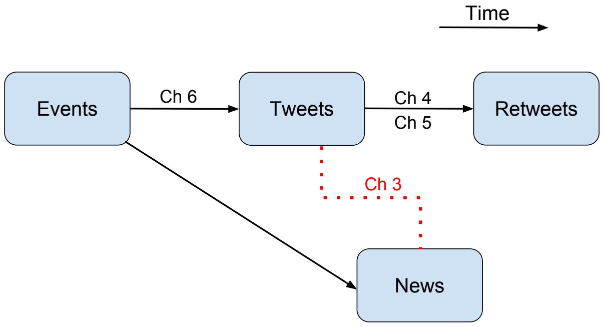


Figure 3.1: The analysis in this chapter relates to the temporal relationship between tweets and news (red dotted line).

3.2 Data collection methodology

We collected data from two major events, the 2016 US Republican Presidential nomination race and the 2016 Australian Federal Election. Even though the Twitter streaming API does not guarantee returning a complete dataset, manually checking tweets indicated that the usernames of the political candidates had a low enough frequency that every tweet mentioning them was recorded. Modern news organisations tweet every new story as they are published and many of them, including those which we use in this chapter, only tweet each news story once. Using a custom Python script, at sixty-minute intervals we collected tweets in the past hour from every selected news service, giving us a complete collection of their published news stories. At sixty-minute intervals, we also collected all tweets in that hour authored by the selected political candidates.

In addition to storing tweets in our database, we also bucketise the tweet counts during collection. Our block size is six-minutes, which generally does not contain many news stories for the political candidates. Clearly, we can combine blocks to analyse the tweet counts in any multiple of this block size. The bucketisation of tweets in this manner was done to improve the performance of analyses; rather than having to conduct a series of potentially slow database searches for every time interval of interest, it is much more efficient to combine the contents of the blocks.

3.2.1 US Republican nomination data collection

For the US Republican nomination race, we collected data between 13 March 2016 to 29 March 2016, and also in the period 11 April 2016 to 20 April 2016. This was in the heart of the race, when Donald Trump was leading in the polls but was still considered unlikely to become the nominee. It was several months before the Republican convention in July 2016 when Donald Trump was announced as the Republican candidate for President. We note that Marco Rubio suspended his campaign during the collection period, on March 20, which affected his news and Twitter volumes.

We counted news events by collecting the tweets from selected news sources that have keywords of interest. For example, for the US Republican race we collected tweets that mentioned usernames of the frontrunner candidates: Donald Trump (@realDonaldTrump), Ted Cruz (@tedcruz), Marco Rubio (@marcorubio) and John Kasich (@JohnKasich).

The thirteen news sources that we considered are the *Associated Press*, *BuzzFeed News*, *CNN*, *Fox News*, *LA Times*, *McClatchy DC*, *NY Times*, *NPR*, *Politico*, *ProPublica*, *Reuters*, *Wall Street Journal* and the *Washington Post*, as were used by the MIT's Electome project [150]. Of these news sources, four are traditional newspapers (*LA Times*, *NY Times*, *Wall Street Journal* and *Washington Post*), two are traditionally television stations but with a modern online presence (*CNN*, *Fox News*), two are multi-source news agencies (*Associated Press*, *Reuters*), four are investigative news agencies (*NPR*, *McClatchy DC*, *Politco*, *ProPublica*) and lastly but importantly we have a digital media news source with a heavy focus on internet trends (*BuzzFeed*). Our news sources are also a mix of politcal leanings, for example, *Fox News* is a right wing television station and *CNN* a center-left leaning station. We consider these news agencies to be a good cross section of news media in the United States. We denote this dataset $A1$.

3.2.2 Australian election data collection

For the Australian election, we collected data collected from 6 June 2016 to 3 July 2016, in the leadup to the election date of 2 July 2016. Australia has a two-party system, the Liberal Party of Australia and the Australian Labor Party, who at the 2016 election were led by Malcolm Turnbull and Bill Shorten, respectively. For news related to the 2016 Australian Federal Election, we collected data from the six news sources *ABC News (Australia)*, *BuzzFeed News Australia*, *Crikey News*, *Guardian Australia*, *Herald Sun* and *Sky News Australia*.

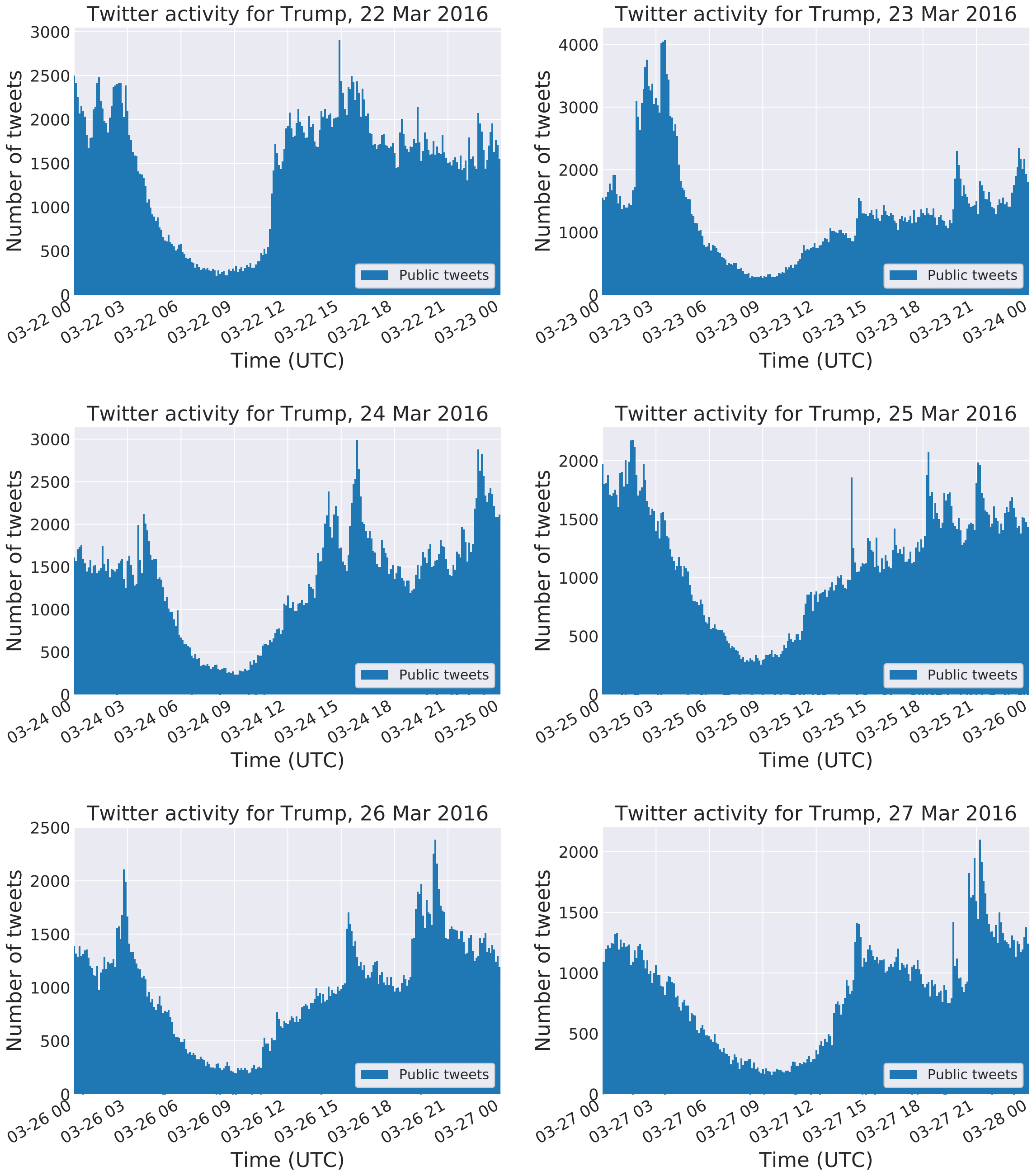
These news sources provide a cross section of news media in Australia. All major newspapers in Australia are owned by either Fairfax Media or News Limited. Consequently, there is a large repetition of articles from major news sources. *Herald Sun* is a Melbourne newspaper owned by News Limited. *ABC News* is a government-owned national news service providing television, radio and online services; it is generally considered slightly left-leaning. *Guardian Australia* is an Australian version of the British *Guardian* newspaper, slightly left-leaning. *Sky News Australia* is a television channel and online news service, which is considered right-leaning. Crikey News is online political commentary and is not mainstream news service. *BuzzFeed News Australia* is also not mainstream and bears similarity to the worldwide *BuzzFeed*.

We use a Twitter collection script to count and record the number of tweets mentioning the Twitter hashtag of four major politicians in Australia: the leader of the Liberal party, Malcolm Turnbull (@TurnbullMalcolm), the leader of the Labor party, Bill Shorten (@billshortenmp), the leader of the Greens party, Richard Di Natale (@RichardDiNatale) and the leader of the Xenophon Team, Nick Xenophon (@Nick_Xenophon). The first two of these politicians are the leaders of the two major political parties in Australia, the second two leaders of minor parties. We denote this dataset $A2$.

3.3 Daily tweet activity analysis

Using dataset $A1$, we first examine the distribution of tweets and news about selected US political candidates for a 24-hour period. As can be seen in Figure 3.2, for Donald Trump there is a clear diurnal pattern of tweet activity with low point around 9 am UTC, corresponding to the early morning in the United States. The six tweet activity plots have similar shapes, particularly overnight. All the plots have spikes of more intense Twitter activity at other times throughout the day.

Figure 3.3 shows the corresponding plots for Ted Cruz. We can see a similar diurnal cycle to that observed for Donald Trump with low point around 9 am UTC, corresponding to early morning in the United States. We collected data over a much longer timeframe than these six days and observed a similar pattern throughout the



Twitter activity for Trump, 22 Mar 2016

Twitter activity for Trump, 23 Mar 2016

Twitter activity for Trump, 24 Mar 2016

Twitter activity for Trump, 25 Mar 2016

Twitter activity for Trump, 26 Mar 2016

Twitter activity for Trump, 27 Mar 2016

Figure 3.2: 24-hour Twitter activity for Donald Trump, 22-27 March 2016. The public tweet data is split into six-minute blocks. The diurnal cycle can clearly be seen with low point at around 9 am UTC, corresonding to 3 am on the East Coast USA.

dataset. During key campaigning events such as debates, there were intense volume spikes.

We now include tweets by the political candidates themselves, along with tweets of news articles about the candidates by news organisations, to observe if these affect the public tweet volumes. Figure 3.4 shows the histogram for Donald Trump tweet and news activity on 26 March 2016. There is a high number of news stories throughout the day. However, there are higher tweet counts when there is a higher number of news stories, such as between 18:00 UTC and 04:00 UTC, corresponding respectively to 11 am and 9 pm in the East Coast of the United States. In addition, there is a higher number of public tweets after a tweet, or series of tweets, written by Donald Trump.

Figure 3.5 shows the equivalent plot on the same day for Ted Cruz. The same patterns apply, with higher public tweet activity coinciding with higher news activity and increased public tweet activity after tweets by the candidate. However, this public response is far less dramatic than for Donald Trump. Also, there are significantly fewer news stories about Ted Cruz than there were about Donald Trump.

3.3.1 Diurnal cycle

Tweet rates vary greatly throughout the day corresponding to changes in human activity [111]. We discuss previous literature on diurnal cycles and adjustment in Section 2.5. In order to quantify our observations about the diurnal cycle, we examine the tweet versus news count for Donald Trump over a 16 day period from 13 to 29 March, 2016. Figure 3.6 indicates that the tweet rate and news count follow the same pattern. The daily spikes can clearly be seen, and the time intervals of highest public tweet counts correspond to the time intervals of highest news activity.

For our collected datasets we calculate the average number of tweets in each hour. We conduct a diurnal adjustment by dividing the number of tweets in a given hour by the average number of tweets within that hour, for a given political candidate over all collected data. For example, if between 8 am and 9 am there are on average 20,000 tweets about Donald Trump, and on a given day there were 25,000, then we record a relative rate of $25,000/20,000 = 1.25$. This normalisation provides a tweet rate that is independent of candidate and the time of day. We repeat the same process for news stories.

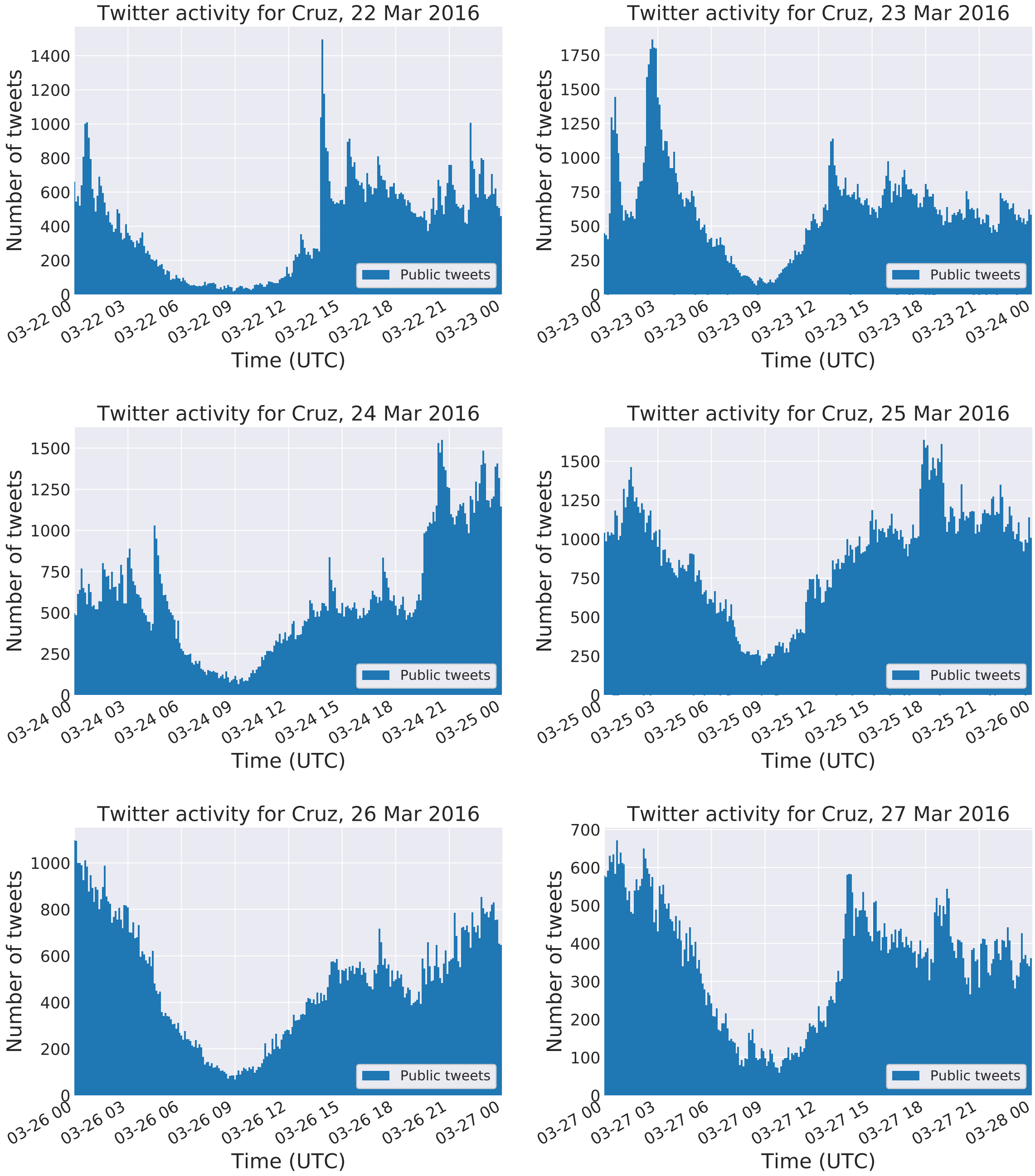


Figure 3.3: 24-hour Twitter activity for Ted Cruz, 22-27 March 2016. The public tweet data is split into six-minute blocks. These plots exhibit a similar diurnal shape as we had for Donald Trump's public tweet data in Figure 3.2. The tweet volumes for Ted Cruz were significantly lower than for Donald Trump

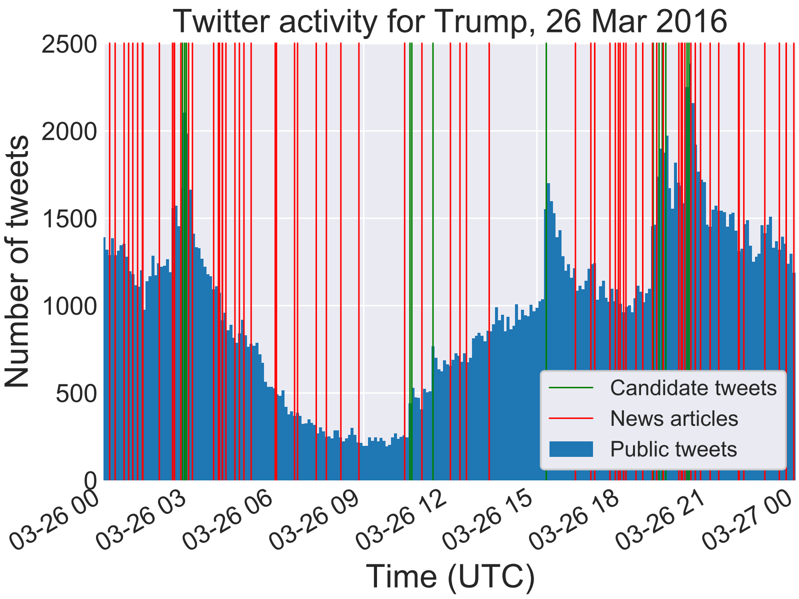


Figure 3.4: 24-hour Twitter activity for Donald Trump, 26 March 2016, with the public tweet data split into six-minute blocks. Higher public tweet activity rates occur around times of news articles or tweets by Donald Trump.

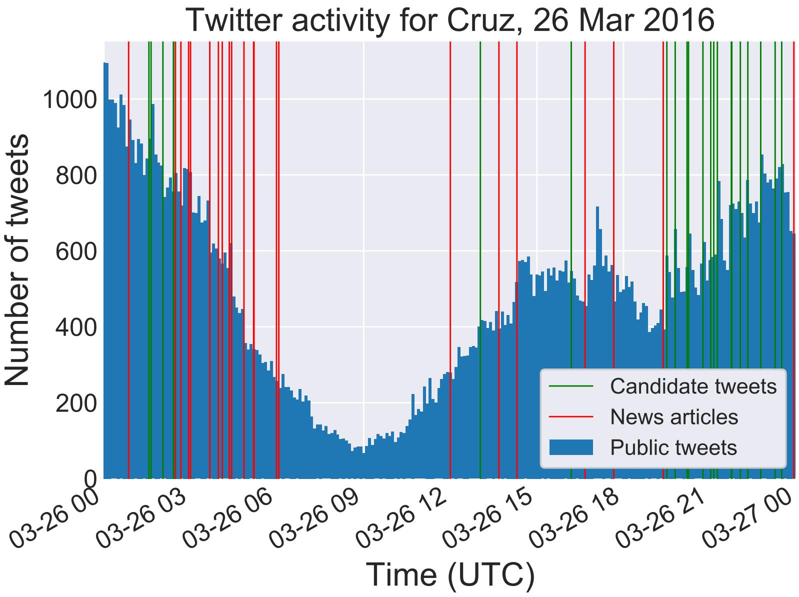


Figure 3.5: 24-hour Twitter activity for Ted Cruz, 26 March 2016, with the public tweet data split into six-minute blocks. Higher public tweet activity rates occur around times of news articles or tweets by Ted Cruz. Tweet rates and news article rates are significantly lower than for Donald Trump.

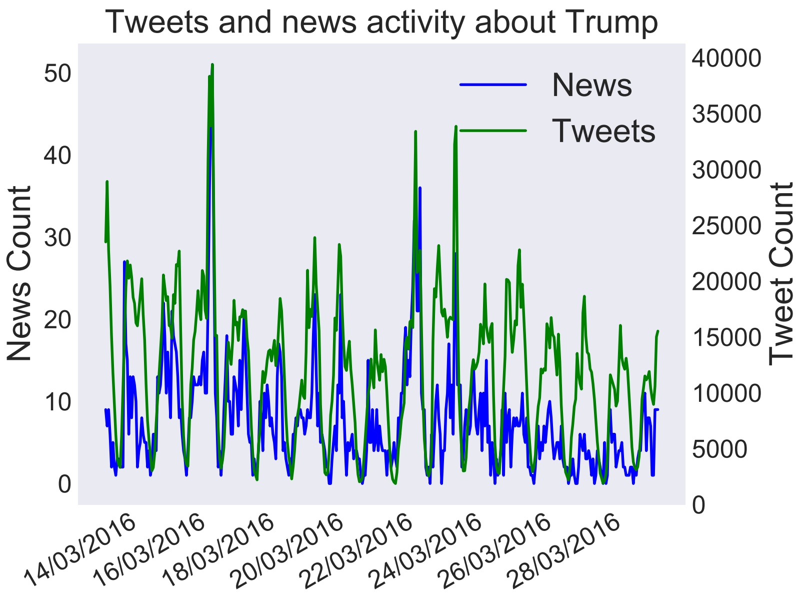


Figure 3.6: News and tweet counts for Donald Trump in 16 days of March 2016. Counts are bucketised into one-hour blocks. The news and tweet shapes show corresponding peaks and troughs in the diurnal cycle.

To check whether the diurnal cycles coincide, we bucketise data into one-hour intervals and average the number of news stories and tweets within each bucket. Figure 3.7 shows the resultant average diurnal cycles for news and tweet counts. The shape of the diurnal cycles are remarkably similar, with the time of highest tweet count occurring at around 1 am UTC (8 pm EST, US Eastern Standard Time), and the lowest tweet rate at around 9 am UTC (4 am EST), when most people in the United States are sleeping, indicates correlation (and possible causality) between the two sets of variables.

We now use this diurnal adjustment to normalise tweet and news counts, removing the effects of the daily cycle. Figure 3.8 shows a plot of news count against tweet count with diurnal adjustment. We divide the tweet and news counts by the average tweet and news counts respectively at each hour of the day. This allows us to see changes in volumes not caused by changes in daily tweet activity. This plot also shows that at times of higher news, there is a tendency for a higher tweet count. Plotting the tweets and news against each other with this diurnal adjustment allows easier detection of key events. The fluctuations in the plot are no longer caused by daily activity variations, but are instead caused by events.

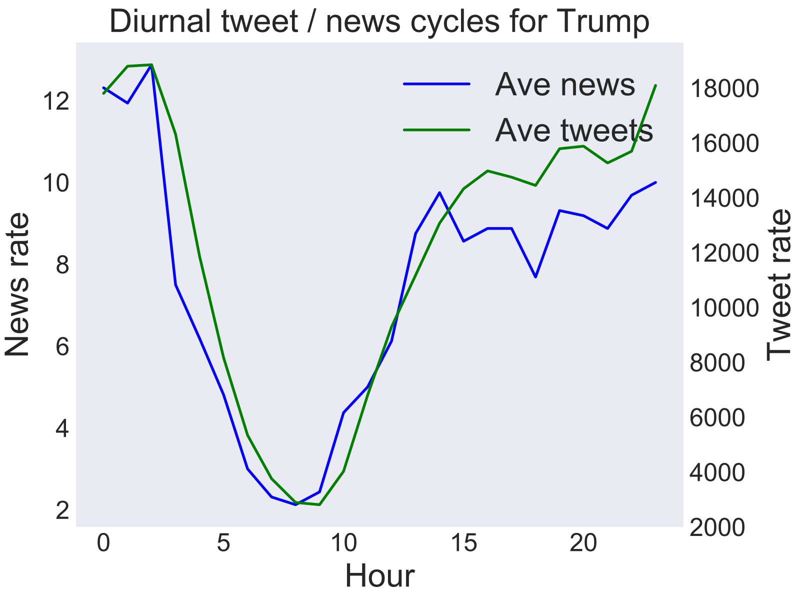


Figure 3.7: Average hourly (UTC) tweet and news rates for Donald Trump (dataset A1). The two sets of rates show remarkably similar diurnal shapes.

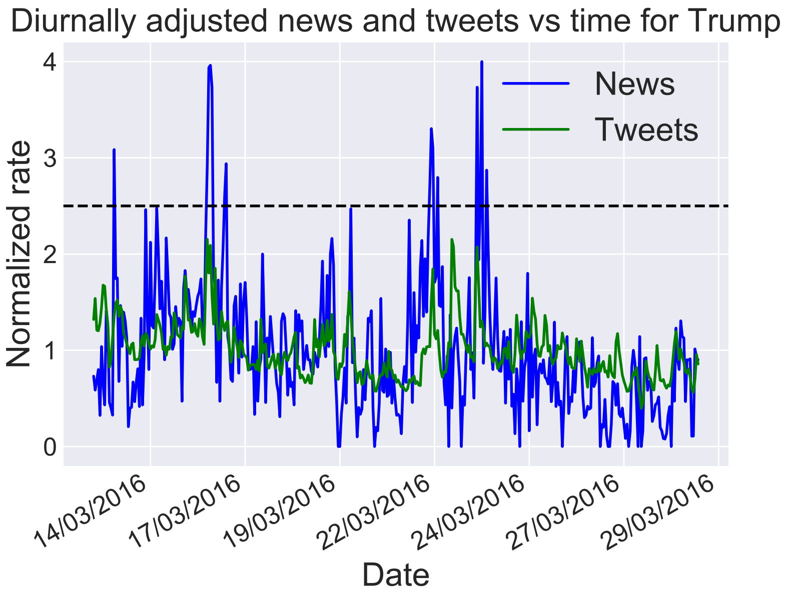


Figure 3.8: Diurnally adjusted and normalised news and tweet counts for Donald Trump in 16 days of March 2016. Counts are bucketised into 60-minute groups. The horizontal black line shows a possible threshold level for Twitter activity, above which an alert could be raised. As can be seen, in this timeframe, the theshold was not reached. The correlation score between the tweets and news was 0.7005.

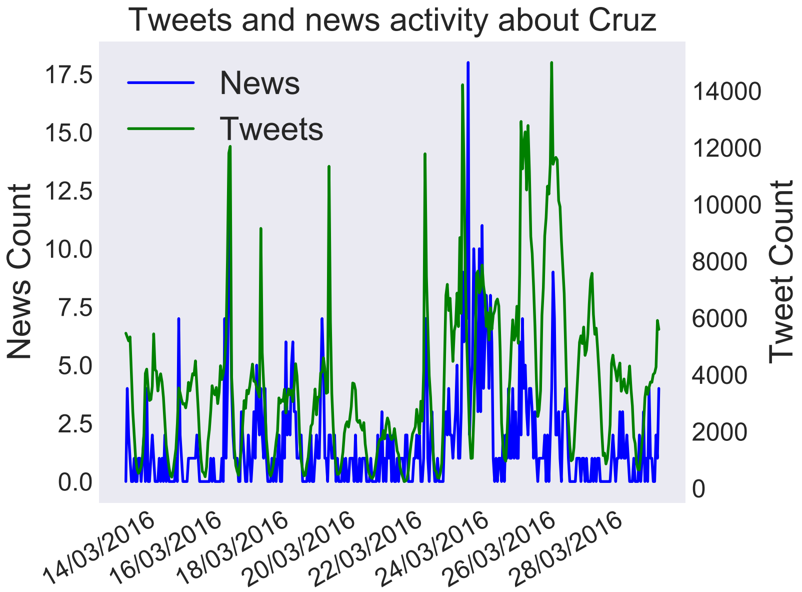


Figure 3.9: News and tweet counts for Ted Cruz in March 2016. Counts are bucketised into 60-minute groups. The two counts show similar shapes, but the lower volumes causes the plots to be less similar than for Donald Trump. The correlation score between the tweets and news was 0.4783.

To check whether this property is unique to one individual, we repeat the same experiment over the same time period for Ted Cruz and show in Figure 3.9 the tweet versus news count. Again, the hours with higher tweet counts correspond to hours with more news. The patterns here are not as clear, mostly due to the much lower news count for Ted Cruz: there were many hours when there were no news stories about Ted Cruz. Figure 3.10 shows the diurnal cycle of Twitter activity and news for Ted Cruz. Here we can see that the two diurnal cycles follow the same shape, which is also very similar to the diurnal cycle shape for Donald Trump. We again note that the closeness of these two plots is striking, providing further suggestion of a relationship between these variables.

Figure 3.11 shows an example plot for the tweet and news counts for Ted Cruz, with a diurnal cycle implemented. The overall tweet rates for Ted Cruz are lower than for Donald Trump, so the patterns have more noise and key events demonstrate a much more pronounced spike. An example of such a spike is on 23 March 2016, when an unsubstantiated sex scandal about Ted Cruz was reported by news organisations [165].

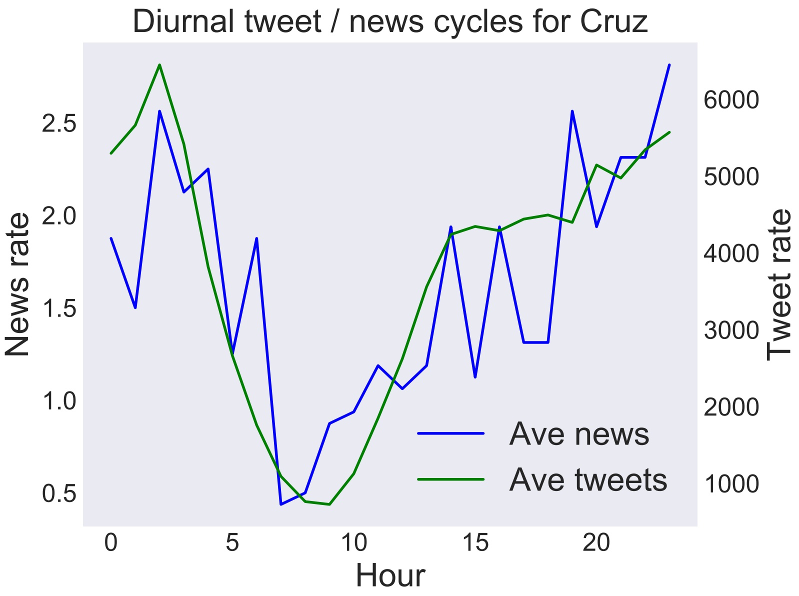


Figure 3.10: Average hourly (UTC) tweet counts for Ted Cruz in 16 days of March 2016. News and tweets show a matching diurnal shape, which is similar to the diurnal shape we observed for Donald Trump in Figure 3.7.

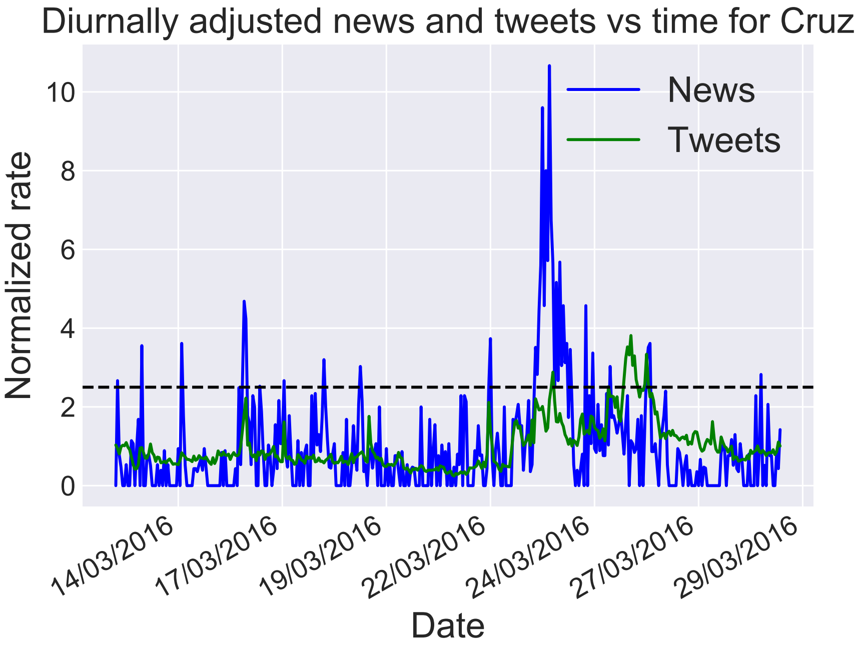


Figure 3.11: Diurnally adjusted and normalised news and tweet counts for Ted Cruz in March 2016, with counts bucketised into one-hour intervals. The horizontal black line shows a possible threshold level for Twitter activity, above which an alert could be raised.

	Block size	
Politician	**30-min**	**60-min**
Donald Trump	0.6434	0.7005
Ted Cruz	0.4314	0.4783
Marco Rubio	0.8065	0.9252
John Kasich	0.8098	0.8796

Table 3.1: Correlations of public tweet count and news count for US politicans, 13-29 March 2016. Data is bucketised into 30- or 60-minute blocks. All values show a clear positive correlation indicating that high tweet activity corresponds to high news activity. Correlation scores increase for the larger time block.

We repeat this analysis between news and tweets for dataset $A2$, using the two leading candidates in the Australian Federal Election, with output plots shown in Appendix A. The overall shape is similar to that for the United States political candidates, but naturally the shape is offset by approximately 13 hours, the time difference between Australia and the United States. However, a difference occurs when looking at the normalised news and tweet ratios. During the evening in Australia, the normalised tweet ratio is much higher than the normalised news ratio for both political candidates. This is likely indicative of the cultural differences between the countries. Whereas US news organisations regularly produce large quantities of news articles throughout the day, including in the evenings, Australian news journalists tend to produce most of their output during the standard 9 am to 5 pm workday.

3.3.2 Correlation between tweets and news

Our analysis in Section 3.3.1 suggested a close relationship between tweets and the news. To test this observation, we calculate the correlation between tweet and news counts for the US candidates, Trump, Cruz, Rubio and Kasich, and display the output in Table 3.1. The values range from 0.4314 up to 0.9252, a clear positive correlation between tweet count and news count. Changing from a 30-minute to a 60-minute time block either increases the correlation scores, or leaves it approximately unchanged, for all four candidates. We attribute this to the proportion of randomness for news counts decreasing over longer time periods. Increasing the block size gives us more data in each block and reduces the relative noise of news counts.

We repeat the same correlation analysis for politicians in the 2016 Australin Federal Election. The output is shown in Table 3.2. The correlations are lower here than for

	Block size	
Politician	**30-min**	**60-min**
Malcolm Turnbull	0.4749	0.5328
Bill Shorten	0.3751	0.4373
Richard DiNatale	0.3582	0.3496
Nick Xenophon	0.5289	0.6996

Table 3.2: Correlations of public tweet count and news count for Australian politicans, 6 June to 3 July 2016. Data is bucketised into 30- or 60-minute intervals. Correlations are less strong than for US politicians, but are still positive. Correlations generally increase for the larger time block.

US politicians, although are still all positive. The correlations for 60-minute blocks are generally higher than for the 30-minute blocks.

3.3.3 Automated event detection through diurnal adjustment

To implement automated event detection using diurnal adjustment, we detect an event as being when the adjusted tweet count has exceeded a selected threshold. The level of the threshold can be chosen based on the significance of the event that a user would wish to detect. For example, the threshold level could be selected as 2.5 times the diurnally adjusted rate of the average tweet count. If over a six-minute period the analysed number of tweets exceeds this value, an alert would be automatically raised. Such automated event detection would be valuable for news organisations who want to know as soon as possible when a significant event is occuring.

Let d_i be the diurnally adjusted average tweet count for the ith six-minute interval of the day, $0 \leq i \leq 240$, and let c be the chosen threshold factor. We denote the measured rate in the ith six-minute interval of the day as a_i and diurnal adjustment function as f. We can raise an alert if

$$f(a_i) > c \times d_i. \tag{3.1}$$

Examples of this threshold are shown in Figure 3.8 for Donald Trump and Figure 3.11 for Ted Cruz, with $c = 2.5$. In the given time period, the threshold for public tweet volumes mentioning Donald Trump wasn't reached, but was reached for Ted Cruz several times between 23 March and 26 March, 2016, during discussion of the unsubstantiated sex scandal. This would indicate to news journalists or other groups that there is significantly more Twitter activity about these selected policitians.

	News → Tweets			Tweets → News	
Lag	Test statistic	p-value		Test statistic	p-value
1	1.81	0.177		74.27	6.79×10^{-18}
2	0.53	0.766		39.61	2.50×10^{-9}
3	3.94	0.219		42.60	2.98×10^{-9}
4	3.79	0.358		49.33	4.96×10^{-10}
5	4.19	0.192		55.05	1.27×10^{-10}

Table 3.3: Donald Trump Granger causality for 30-min intervals

This diurnal adjustment alert mechanism would be expected to stay valid while the underlying social dynamics remain constant. However, if a change occurred, such as Donald Trump getting a large increase in followers from another time zone, then the diurnal adjustment would likely need to be recalculated.

3.4 Granger causality

Although it is infeasible to test true causality between news and tweet counts, we can test whether one of the variables leads the other by measuring the *Granger causality* [62], as introduced in Section 2.6.4. We select lag values to examine the effect that chosen older values have on current values. For example a lag of 4 combined with a block size of 30 minutes tests the effect of variables from 120 minutes ago on current values. We conduct Granger causality tests for the news and tweet counts over various time blocks, with results for tweets about Donald Trump over 30-minute intervals shown in Table 3.3. Testing whether news Granger-causes tweets, every lag value ranging from 1 to 5 gives relatively small test statistics and p-values of greater than 0.05. We consequently cannot reject the null hypothesis; in other words, we cannot conclude that news Granger-causes tweets. However, testing whether tweets Granger-caused news gives consistently high test statistics for all lag values, and p-values far less than 0.05; so we reject the null hypothesis and conclude that tweets Granger-cause news.

Repeating Granger causality tests for Ted Cruz produces a similar pattern, as seen in Table 3.4. Testing whether news Granger-causes tweets gives p-values greater than 0.05, so we cannot reject the null hypothesis. However, testing whether tweets Granger-cause news gives p-values less than 0.05 so again we reject the null hypothesis and conclude that tweets Granger-cause news.

	News → Tweets			**Tweets → News**	
Lag	**Test statistic**	**p-value**		**Test statistic**	**p-value**
1	0.30	0.580		50.97	9.33×10^{-13}
2	2.99	0.223		35.42	2.03×10^{-8}
3	3.94	0.267		29.42	1.82×10^{-6}
4	3.79	0.434		31.82	2.07×10^{-6}
5	4.19	0.521		33.32	3.25×10^{-6}

Table 3.4: Ted Cruz Granger causality for 30-min intervals

Our conclusion is that for selected high-profile politicians in the 2016 Republican Nomination race, tweet counts Granger-cause news counts. To our knowledge, there have not been any previously published definitive conclusions about Granger causality between tweet counts and news counts.

3.5 Modelling and prediction of tweet and news activity

We now examine a more challenging aspect of the relationship between tweets, news and other variables, modelling and predicting the overall volumes. We do this to investigate how accurately such a prediction can be done, while recognising that we have already shown relevant Granger causality results in Section 3.4. (An introduction on linear regression and neural networks, both used for prediction, is given in Section 2.6.)

Using both datasets $A1$ and $A2$, we split our tweet and news counts into one-hour time periods. We have a total of 7812 data points, each representing an hour of data for a selected individual. We outline the input and output variables for tweet volume prediction in Table 3.5, with the target account of interest taken as a one-hot encoded variable (defined in Section 2.6.3).

We select $k = 2$, using the past two hours of data for modelling relationships between variables and future predictions. Including the hour of the day potentially allows the neural network to learn diurnal cycles and the day of the week allows learning of the weekly cycle. Including the day of the year allows recording any longer term trends, such as an individual's Twitter profile increasing in popularity over time. As we are only looking at data over a single year, it is not necessary to include longer term variables. We compare the performance of our neural network prediction with linear regression and a baseline prediction method of the variable of interest in the

Variable	Description
X_1 to X_8	Target account of interest (Turnbull, Shorten, DiNatale, Xenophon, Trump, Cruz, Rubio, Kasich)
X_9	Number of news stories in prediction hour
X_{10}	Number of tweets by the target user in prediction hour
X_{11}	Hour of the day
X_{12}	Day of the week
X_{13}	Day of the year
X_{14} to X_{13+k}	Number of public tweets in each of the previous k hours before prediction hour ($k \geq 1$)
X_{14+k} to X_{13+2k}	Number of news stories in each of the previous k hours before prediction hour ($k \geq 1$)
X_{14+2k} to X_{13+3k}	Number of user (candidate) tweets in each of the previous k hours before prediction hour ($k \geq 1$)
Y_1	Number of public tweets mentioning the user's account in the hour

Table 3.5: Input (X_i) and output (Y_j) variables for tweet volume prediction.

previous time period.

We use the LinearRegression class from Scikit-learn [122] in Python 3.6 to implement the linear regression. We use 5-fold cross validation to test the accuracy of the output, with the mean absolute error (discussed in Section 2.6.2) used as the metric. For each iteration of the cross validation, we consequently train on 80% of randomly selected data and test on the remaining 20%.

For our neural network, we have 19 input variables, which we feed into two hidden layers with 12 and 8 nodes respectively, a neural network structure that was determined through experimentation to find a model that performs well. Larger models tended to overfit the data, while simpler models had a tendency to underfit and struggle to model the non-linear components of the output such as the diurnal cycle. Both of the hidden neural network layers have a *ReLU* activation function and random uniform initialisation. As we are predicting a floating point output, the final node is fed into a single variable with no activation function. We use 5-fold cross-validation in order to test the accuracy of the output, with the mean absolute error (MAE) used as the metric. Our neural network was implemented with a *Keras* wrapper over a *Tensorflow* backend, using Python 3.6. The package Scikit-learn [122] was used to implement the k-fold cross-validation. The neural network training was done using an Nvidia GTX 1070 Ti GPU on a system with a i7-6700k processor and 16GB of RAM.

	Mean of 5-fold validation MAE values	**STD of 5-fold validation MAE values**
Previous hour's tweet count	931.20	98.32
Linear regression	847.55	63.87
Neural network	726.50	67.28

Table 3.6: Estimation of tweet counts (MAE = mean absolute error. STD = standard deviation).

The results are given in Table 3.6. For linear regression, the mean absolute error is 847.55 with standard deviation 63.87. The neural network performs better with mean absolute error of 726.50 with standard deviation 67.28. For reference, the average number of tweets per hour in the dataset is 3834, suggesting prediction error rates of roughly around 20%. However, the number of tweets depends heavily on the candidate. For example there were an average of 9829 tweets per hour mentioning Donald Trump's account name, with minimum and maximum of 892 and 66,196 respectively. The mean absolute error using linear regression is 16.7% higher than using a neural network, giving approximate prediction error rates of 23%. This isn't surprising as the neural network is able to predict much more complicated functions than a linear model. Neural networks perform best when they have huge amounts of data [68]. We are only training on several weeks worth of data, so the comparison between the methods is close. If we added much more data it is likely that the neural network would outperform the linear regression by a higher margin.

We also perform our prediction in the opposite way, using public tweets and other variables to predict the number of news stories in each hour. We conduct news activity prediction in the same manner as the tweet activity prediction model, using the same variables and using both linear regression and a neural network to predict the output. Altering the model to predict news volumes, we change the previous output variable to an input variable:

X_9: Number of public tweets mentioning the user's account in the hour

and the output variable becomes:

Y: Number of news stories in the prediction hour.

This is a substantially different problem for two key reasons. First, the news story count is a small discrete variable compared to the large public tweet count. Consequently this variable can change by a substantial percentage due to random fluc-

	Mean of 5-fold validation MAE values	STD of 5-fold validation MAE values
Previous news count	2.267	0.282
Linear regression	1.715	0.093
Neural network	1.583	0.071

Table 3.7: Estimation of news counts including public tweets as input features (MAE = Mean Absolute Error. STD = Standard Deviation).

tuations. Second, based on the Granger causality results of Section 3.4, we expect previous tweet counts to contain information useful to predict current news counts. The first of these factors makes this a harder problem than tweet activity prediction, while the second makes this an easier problem.

Results for the news prediction, along with using the baseline measure of the previous hour's news count, are shown in Table 3.7. The neural network outperforms linear regression with a mean absolute error of 1.583 compared to 1.715, a 9.2% improvement. The number of news tweets per hour was extremely variable and dependent on candidate. For Donald Trump the number of news articles per hour ranged from 0 to 116. There were an average of 7.45 news articles with a standard deviation of 8.43. However, overall there was a much lower average of 3.12 news articles per candidate per hour.

3.6 Discussion and conclusions

The strength of the correlation between the news and corresponding Twitter activity on the same topic, as shown in Section 3.3, is striking, and indicates that the two quantities of Twitter discussion and news discussion are very closely linked. Our result in Section 3.4, that the Twitter activity Granger causes the news activity, is also a noteworthy result which provides insight into the timing relationship between these two variables.

There are several possible explanations for the observed Granger causality results. A first possible explanation is the lag that occurs in news publication. Events that occur on television, such as debates, are discussed on social media in near real time. Events that occur in densely populated public places, such as the Boston Marathon bombing, will also have a near real-time response. However, news organisations generally publish after the events have finished, creating a lag. A second possible explanation is that many news reports are reactive and are written only after a story has broken in another newspaper. Even if an initial news report is released first in one media source, social media may then react before the rest of the media outlets piggyback off the initial release and publish their own stories. This can also lead to the Granger causality analysis showing that Twitter reaction occurs before news reaction. A third possible explanation, is that social media actually causes news organisations to react. This would happen when news organisations become aware of the public attention towards a topic and consequently want to leverage off this attention. Such a concept is difficult to quantify, and would require close examination of news article trends to see if there are references to Twitter activity.

We focused on news sources and individuals from two Western countries, the United States of America and Australia. It would be possible to extend this work to check whether the relationships hold for non-Western countries. The underlying mechanisms may be different in countries with a government-controlled media. We also focused on politicians as they are regularly reported in the news, particularly during a political campaign. It's likely that the results would be similar for celebrities in other areas, but investigating this is outside the scope of this research. It is less insightful to conduct streaming collection of famous people who are not being continually discussed in the news; the data is sparse so the collection must occur over a much longer timeframe and correlations are more difficult to demonstrate. One difference in the underlying social mechanics is that politicians, especially those in power, have an enduring public profile. In contrast, the focus on other celebrities or content creators is more likely to vary with social trends. Future work could test such relationships and determine whether the same conclusions would apply to famous people in other areas.

Chapter 4

The Temporal Distribution of Retweets

4.1 Introduction

Modelling population-level phenomena such as social contagion and information diffusion are contingent upon a detailed understanding of the underlying information sharing-processes [35, 144, 176]. On Twitter, an important aspect of this occurs with retweets, where users rebroadcast the tweets of other users. To improve our understanding of these processes, we analyse the distribution of retweet times. For a given seed tweet, we wish to know the distribution of times for retweets that follow. As we shall show, the highest density of retweets occurs immediately after the initial tweet and then decays. In addition to estimating the rate of decay itself, we also wish to know what factors affect the rate of decay. Prior work relevant to this chapter is discussed in Section 2.2.

Retweeting is an extremely straightforward process; a user only has to hit a single button on their graphical user interface, causing the seed tweet to be broadcast to their followers. In particular, this is much simpler than the user having to draft and write a new tweet in response to an event. As we shall see by comparing the results here to Chapter 6, this causes retweet times to have a different distribution than the times between events and reactionary tweets.

Section 4.2 outlines how we use the Twitter Streaming API [153] to collect complete retweet datasets, an essential step for the subsequent statistical analysis. We focus on dense datasets from personalities with popular Twitter accounts such as Donald Trump, since having more data points gives more precise distribution fits. We then

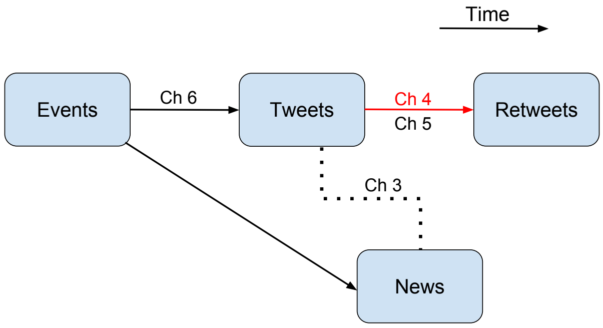


Figure 4.1: This chapter analyses the temporal relationship between tweets and retweets (red link).

analyse our collected datasets to better understand retweet rates. We initially look at the first three hours after the inital tweets and examine retweet rate decay for a set of popular tweets. We show that in this short time period, the data shows power law characteristics and is not strongly affected by diurnal effects. We then look at a longer time period, up to 24 hours, for our sample tweets and show that a power law with exponential cutoff better explains the data than a power law does. We also demonstrate diurnal effects in this time period and present a stochastic method, built upon the approach from Chapter 3 to adjust for these effects.

Next, in Section 4.3, we statistically test our observations on a much larger and more diverse dataset. We determine the percentage of datasets that pass Clauset's test [33] for a power law. We then examine how the power law parameter varies based on categorisation of the tweet's author. We finish the analysis by statistically demonstrating a new key result, that a power law with exponential cutoff better explains the distribution of retweet times than the widely-used power law. We do this by using the Akaike Information Criterion (AIC), to show that the improvement in the quality of the fit justifies the additional parameter in the model.

A significant portion of the work in this chapter was peer-reviewed and published in Modeling Social Media (MSM) 2017, a workshop of WWW2017 [103]. As is indicated in Figure 4.1, this chapter fits into the thesis structure by analysing the temporal relationship between tweets and retweets.

This chapter makes the following key new contributions:

- Showing that the decay of retweet rates over time is well-modelled by a power law with exponential cutoff.

- Developing a method to stochastically adjust for the diurnal cycle in Twitter activity rates.

4.2 Analysis of example seed tweets

We first analyse the retweet time distributions arising from some example seed tweets in depth to better understand these distributions. This enables us to make hypotheses about the population of retweet data sets, that we then test on a larger and more diverse dataset with statistical analysis.

4.2.1 Data collection and processing methodology

It is crucial to select sample users carefully to maximise the insights to be gained from our analysis. In order to obtain large and diverse retweet sets, we would ideally have a sample of users who tweet frequently, are heavily followed and have diversity in the times and topics of their tweets. For an initial illustration of retweet behaviour, we first focus on retweets from seed tweets by the American businessman and politician Donald Trump during the 2016 US Republican Primaries, nine months before he became President of the United States. To demonstrate that the findings are not unique to this individual user, an additional corresponding analysis of tweets by the American politician Ted Cruz is presented in Appendix B.1.

We use the Twitter REST API [153], discussed in detail in Section 1.2, to access retweet data. The *GET statuses/retweets/:id* request returns a collection of the 100 most recent retweets of the specified tweet. The Twitter API allows 15 retweet GET requests per 15 minutes, one per minute on average. However, accessing the 100 most recent retweets of older seed tweets, which already have more than 100 retweets, is not sufficient for our purposes. For thorough analysis of a retweet distribution we need the times of every retweet that occurs from an initial seed tweet.

To obtain the desired datasets, we created and ran a custom Python 3.6 script which checks for a new tweet from a particular user every 60 seconds. When this initial tweet occurs, the script then begins periodically asking for the most recent 100 retweets. In order to avoid hitting Twitter's rate limit, we stop the collection of any retweet set that has a retweet rate greater than 60 retweets per 60 seconds, an average of one per second. New retweets that have not been observed previously are added to the list of collected retweets. This occurs for the next month from the time of the inital seed tweet, in order to collect all retweets in the period. To maximise the number of tweet collection scripts running at any time, we slow the rate of API requests for the 100 most recent retweets as time progresses. This allows the collection of retweets from many seed tweets in parallel, from a single automated Twitter account.

Our collection software was set up in 2016 to record seed tweets and corresponding retweets from a number of users, including Donald Trump. Let $T_i = \{t_{i1}, \ldots, t_{in}\}$, $i \in \{\text{A, B, C, D, E, F}\}$, be the retweet set from the ith dataset, where t_{ij} is the number of seconds between the seed tweet and the jth retweet for the ith dataset. The example set in Table 4.1 was selected to give a clear demonstration of the distribution of retweet times, and is typical of the population of retweet data sets. These seed tweets were taken from 7-9 February 2016, around the time of the New Hampshire Republican primary. At the time, Trump was the frontrunner in the polls to become the Republican nominee for President of the United States [124]. He had a strong following on Twitter but far less than that obtained after winning the candidacy and eventually becoming President [65].

It is useful to consider the first few hours after the initial tweet separately, as this period is less likely to be affected by the daily diurnal cycle [56]. We also analyse a 24-hour period after the initial tweet, which includes most of the eventual retweets. In this time period we investigate how the diurnal cycle affects the retweet rate distribution. Finally we examine longer time periods, up to a month after the initial tweets.

From the seed tweets in Table 4.1, we note that tweet F occurred at 04:06 UTC, corresponding to 11:06 pm on the East Coast of the USA. Even though we collect retweets globally, most interest in the Republican race for Presidential candidate is from the United States. Consequently many potential retweeters would not see (and potentially retweet) this tweet until the following morning.

4.2.2 First three hours after initial tweet

We analyse the first three hours of retweet decay after six initial seed tweets from Donald Trump. We first look at histograms of tweet times and plot rates on a log-log scale to gain a visual understanding of retweet behaviour.

Label	Tweet text	Tweet date (UTC)
T_A: Trump A	I will be on State of the Union @CNN with @jaketapper at 9 am. Enjoy!	2016-02-07 13:19:33
T_B: Trump B	Great to meet everyone while having breakfast @ChezVachon this morning! #FITN #VoteTrumpNH	2016-02-07 16:29:09
T_C: Trump C	My two wonderful sons, Don and Eric, will be on @foxandfriends at 7:02 - now! Enjoy.	2016-02-08 12:01:01
T_D: Trump D	Thank you for your support at this mornings Town Hall- in Salem, New Hampshire. #FITN #NHPrimary https://t.co/4m6dabtxCV	2016-02-08 16:29:53
T_E: Trump E	Thank you, New Hampshire! #FITN #NHPrimary #VoteTrumpNH Voting questions? https://t.co/BmZyKQOZJJ https://t.co/1tZfqVETrX	2016-02-09 02:20:29
T_F: Trump F	Thank you, New Hampshire! #FITN https://t.co/uZItWkqQZa	2016-02-09 04:06:58

Table 4.1: Sample tweet details from Donald Trump (Twitter: @realDonaldTrump). We collected these tweets during the Republican nomination process, just after the New Hampshire primary.

We create eighteen linearly spaced 10-minute bins, between the time of the intial retweet and three hours afterwards. In this timeframe, the diurnal cycle has smaller effects on the resulting distribution. Figure 4.2 shows the retweet distribution for a single tweet in the first three hours. The shape of all six plots is similar, and as expected the retweet density slowly decreases over time.

We then split the data into log-spaced bins, starting at $t_0 = 60$ seconds. Calculating the log of the retweet rate in each bin (number of tweets / bin width) gives Figure 4.3. The data points appear approximately linear with some noise. To demonstrate mathematically how a straight line on a log-log plot leads to a power law, consider tweet F where we have line of best fit $y = -0.674x + 3.022$. Consequently, for retweet rate $R(t)$, t in seconds, we have

$$\log(R(t)) = -0.674 \times \log(t) + 3.022, \tag{4.1}$$

which implies that

$$R(t) = e^{-0.674 \log(t) + 3.022} = 20.53t^{-0.674}. \tag{4.2}$$

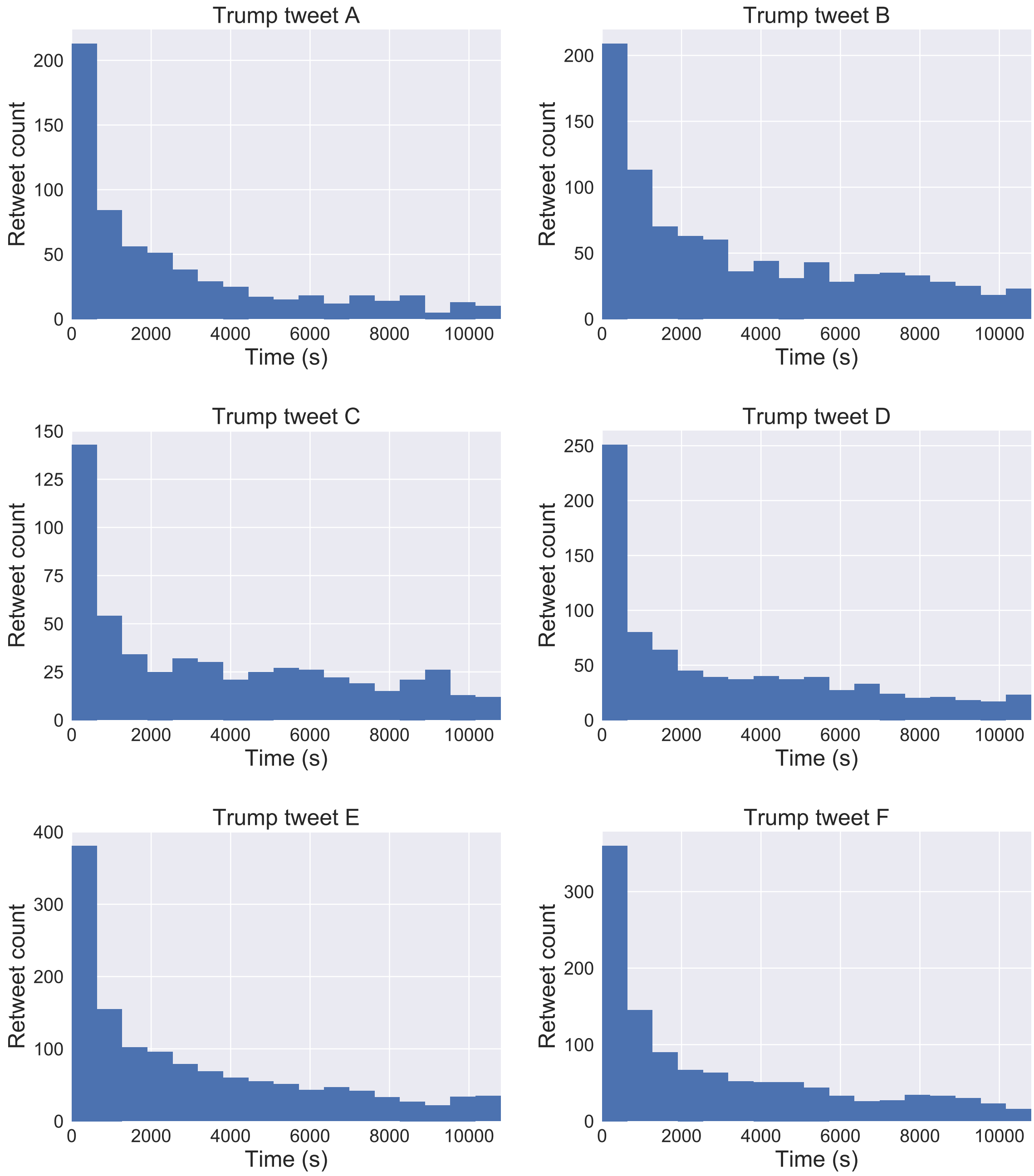


Figure 4.2: Donald Trump seed tweets: First three hours of retweet distribution histogram. The retweet rate decays over time, with some level of noise. Note that the vertical scales vary between the tweets; the shape remains similar but the magnitude varies depending on the popularity of the tweet.

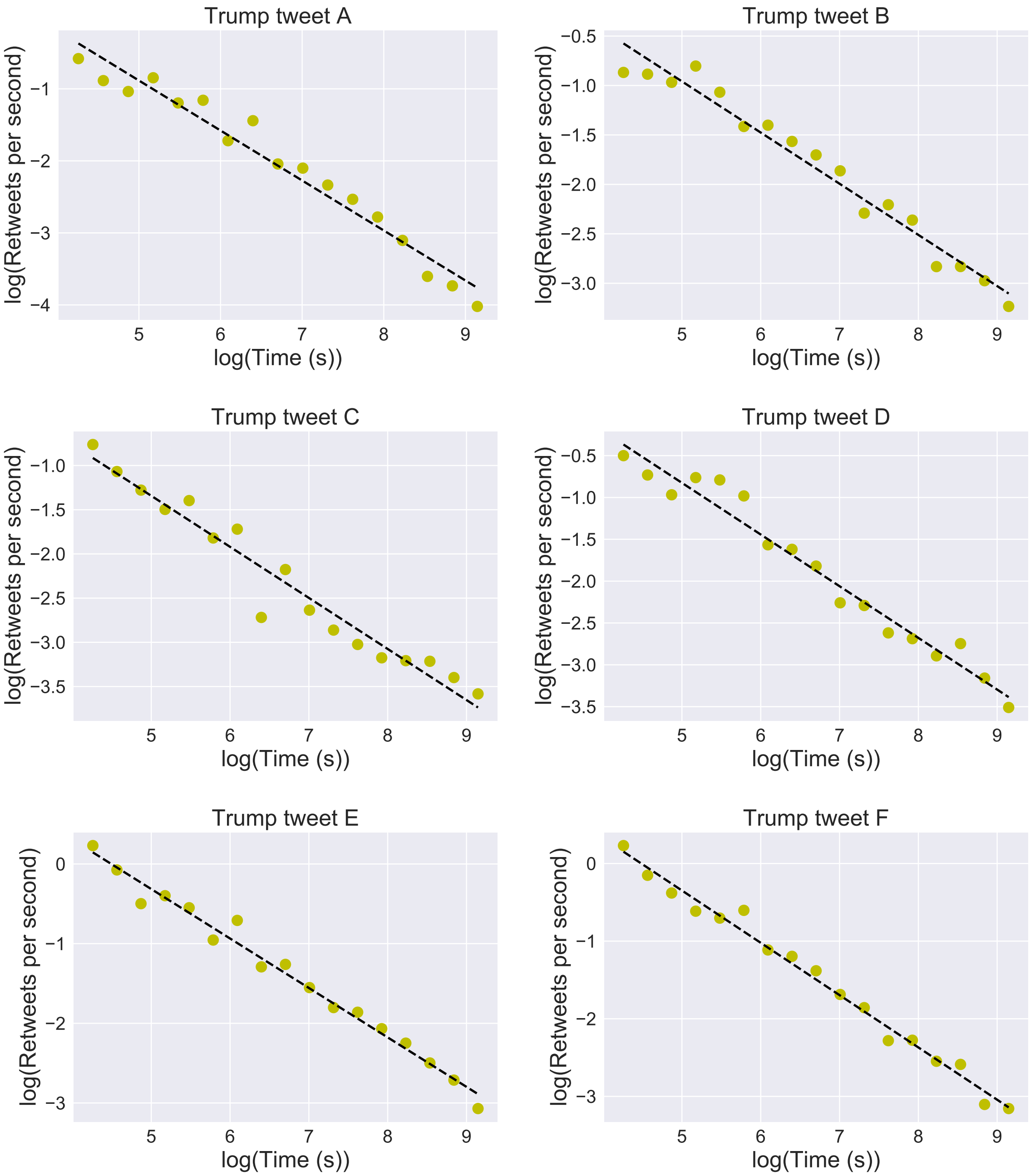


Figure 4.3: Donald Trump seed tweets: First three hours of retweet distribution on a log-log plot. The linear shape of the curve on a log-log plot indicates that the retweet rate distribution behaves as a power law in this time period.

Dataset	α	R^2
A	0.691	0.933
B	0.508	0.919
C	0.570	0.909
D	0.605	0.904
E	0.618	0.972
F	0.674	0.987

Table 4.2: Donald Trump: Power law parameters for three-hour retweet collection. Most of the values of the power law parameter α are around 0.6 with high R^2 values above 0.9, demonstrating approximate linearity of the log-log relationship.

Table 4.2 summarises the power law parameter α and R^2. The values for α are in the range 0.5 to 0.7, a very slow decay rate for a power law [126], with all R^2 values above 0.9 indicating a close fit.

4.2.3 First 24 hours

We now look at the first 24 hours after the initial tweet. Similar to Section 4.2.2, we split the data into eighteen bins, both on a linear and log-log scale. As we are observing an entire day, we expect the retweet rates in this timeframe to be heavily affected by the diurnal cycle. Histograms of retweet frequencies for the first 24 hours are shown in Figure 4.4.

The retweet rate continues to decay over time. Tweets E and F were sent at 02:20 UTC and 04:06 UTC respectively, corresponding to 9:20 pm EST and 11:06 pm EST (US East Coast time). For these tweets we can clearly see diurnal effects. Tweet E shows a spike about 9 hours after the initial tweet, corresponding to approximately 6 am EST. Tweet F shows a spike about 7 hours after the initial tweet, again corresponding to approximately 6 am EST. A logical explanation for this is that Twitter users check their accounts after waking up, see the tweet and then decide whether to retweet.

We again plot this distribution on a log-log graph to see if the data is well-modelled by a power law. As can be seen in Figure 4.5, the linear curve no longer fits the data for longer times. Given the fall away from the linear line of best fit, we consider fitting functions that begin as a power law then decay more rapidly. One example of such a function is a power law with exponential cutoff, which has density function $R(t) = At^{-b}e^{-ct}$, where b is the power law parameter, c is the exponential cutoff parameter and A is a constant.

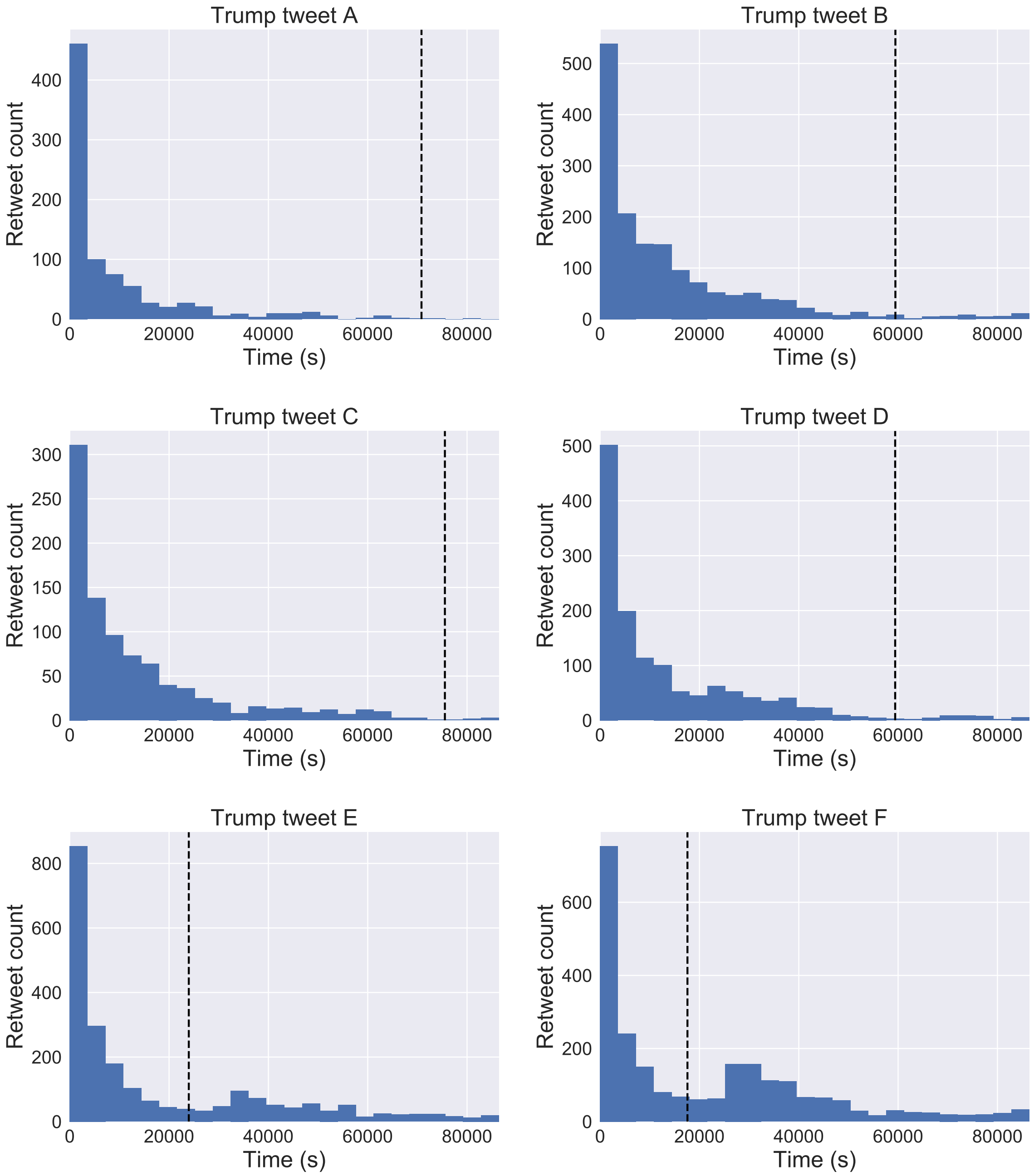


Figure 4.4: Donald Trump seed tweets: First 24 hours of retweet distribution histogram. The varied shapes of the plots (particularly for E and F) are caused by diurnal effects, with higher relative retweet rates when people are awake. The vertical dotted black line indicates 9 am UTC, the time of lowest Twitter activity in the United States.

Dataset	A	b	c
A	10.15	0.6435	3.992×10^{-5}
B	3.536	0.4498	3.647×10^{-5}
C	3.166	0.5049	3.396×10^{-5}
D	6.880	0.5571	2.965×10^{-5}
E	2.142	0.6661	1.024×10^{-5}
F	1.941	0.6654	4.177×10^{-6}

Table 4.3: Donald Trump seed tweets: Parameters for power law with exponential cutoff curves of best fit for first 24 hours.

For very small values of t, $e^{-ct} \approx 1$ so the density function is approximately equal to a power law. We fit this curve to each of the six originating tweets, using the *curve_fit* function from the scipy.optimize package in Python 3.6. This curve fits the data much closer than a linear relationship. Tweet sets E and F, which are affected by the diurnal cycle more heavily than the other data sets, show some noise near the tail of the graph. The curves of best fit have parameters given in Table 4.3. As with the power law without cutoff, the power law parameter b is again quite low, less than one in every case. The exponential cutoff parameter c is within an order of magnitude of $3 \times 10^{-5} \approx 1/(9 \times 60 \times 60)$, contributing a factor of $1/e$ after approximately 9 hours.

It is also of note that tweet A and tweet C both have curves with sharp decay. Both were about the candidate's appearance on TV shows in the near future. It is natural for these tweets to be of less interest after the TV show has occured, hence the faster decay. However, a power law with exponential cutoff still provides a good fit to the data within this time frame.

Subsequently in Section 4.2.5, we consider the same 24-hour period with a diurnal adjustment implemented.

4.2.4 Longer time durations

The majority of retweets occur in the first 24 hours after an initial tweet. When an unread tweet becomes several days old, users will tend to not see it when scrolling through their Twitter feed. Consequently the mechanics of how retweets occur for very old tweets are potentially different than for newer tweets. However, for completeness, we now consider a much longer time period, the first month after the initial tweet. Over this time period, we find the power law with exponential cutoff model no longer holds. The tail of the distribution is heavier than would occur from

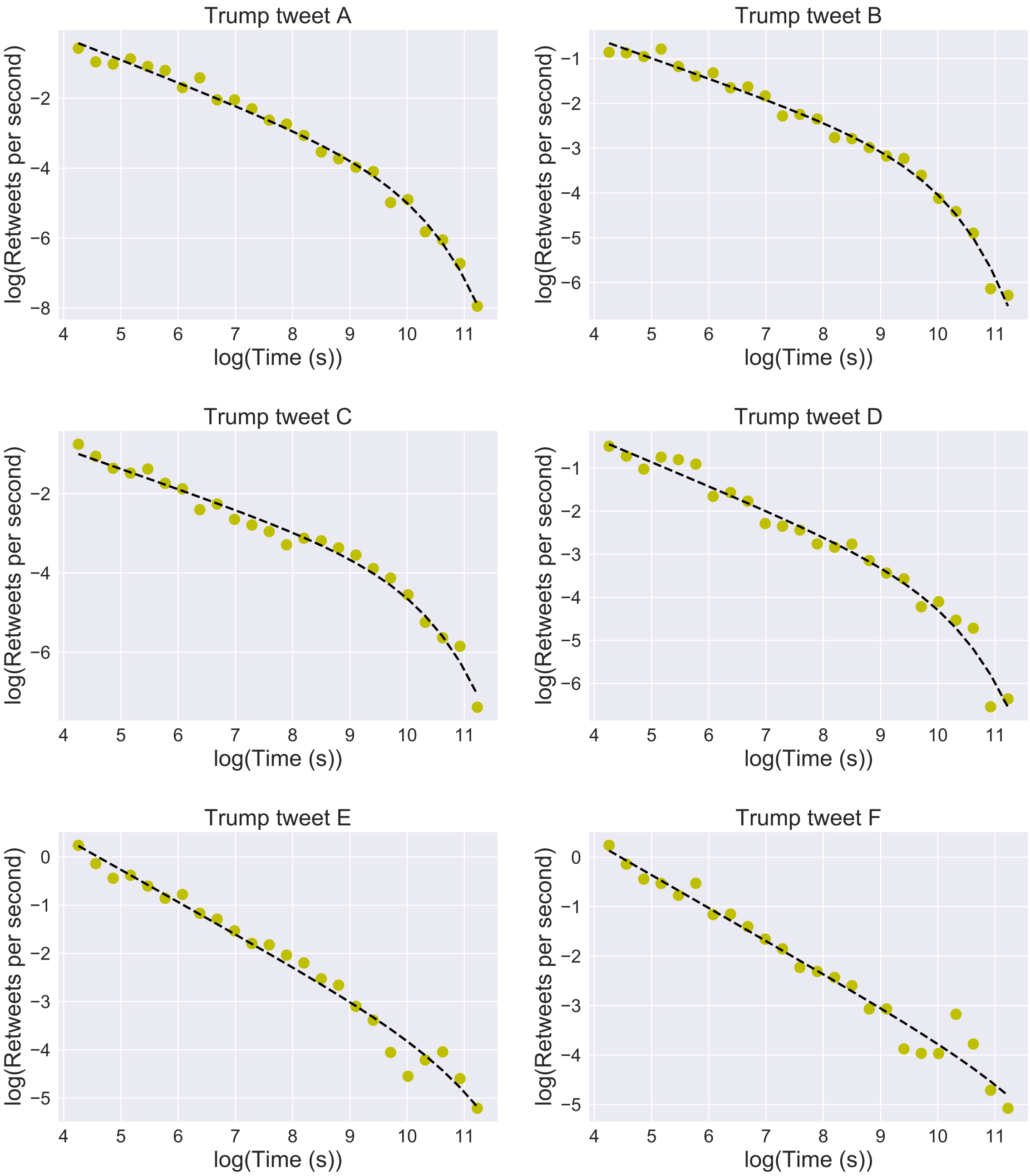


Figure 4.5: Donald Trump seed tweets: First 24 hours of retweet distribution on a log-log scale. A power law with exponential cutoff provides a better fit to the curve than a power law.

Dataset	A	b	c	d
A	3.925	0.8605	1.498×10^{-5}	0.0348
B	10.98	0.6205	1.911×10^{-5}	0.0138
C	12.39	0.7185	1.141×10^{-5}	0.0706
D	9.059	0.6051	2.324×10^{-5}	0.0183
E	16.06	0.6287	1.500×10^{-5}	0.0589
F	7.575	0.5083	2.116×10^{-5}	0.0419

Table 4.4: Donald Trump seed tweets: Parameters for curves of best fit for one month retweet collection.

a distribution with an exponential cutoff. In addition, the rate of retweets a month after the initial tweet becomes too low to accurately model. We suggest a model which could potentially be used, given sufficiently large data volumes.

A model which visually does a reasonable job of fitting the data is

$$R(t) = At^{-b}((1-d)e^{-ct} + d), \tag{4.3}$$

where typically $d \approx 0.03$. For values of t less than 3 hours, this function behaves as a power law. For values of t approximately between three and 48 hours, it behaves like a power law with exponential cutoff. For much larger values of t, this model behaves as a power law again. Examples of this function fitting the data are shown in Figure 4.6. The function does a reasonable job of fitting the data, but not nearly as well as the power law with exponential cutoff in Section 4.2.3 did for the first 24 hours.

Parameters for the curves in Figure 4.6 were determined with the curve_fit function from the scipy.optimize package. The parameters giving the best fit are shown in Table 4.4.

We note that the retweet rates in the period between 48 hours and one month after an initial seed tweet are extremely low, below e^{-10} tweets per second, corresponding to less than one retweet per six hours. This rate is exceptionally low that the datasets could be heavily affected by bot activity, a user systematically retweeting a series of tweets from a chosen author, or even just random variability. Consequently we do not claim that Equation (4.3) is an appropriate model for all tweets. However, we include this analysis for completeness, in order to demonstrate the retweet behaviour over this long time period.

The different models in the 3-hour, 48-hour and one-month time ranges correspond to the different human behavioural tendencies which would lead to a user reading

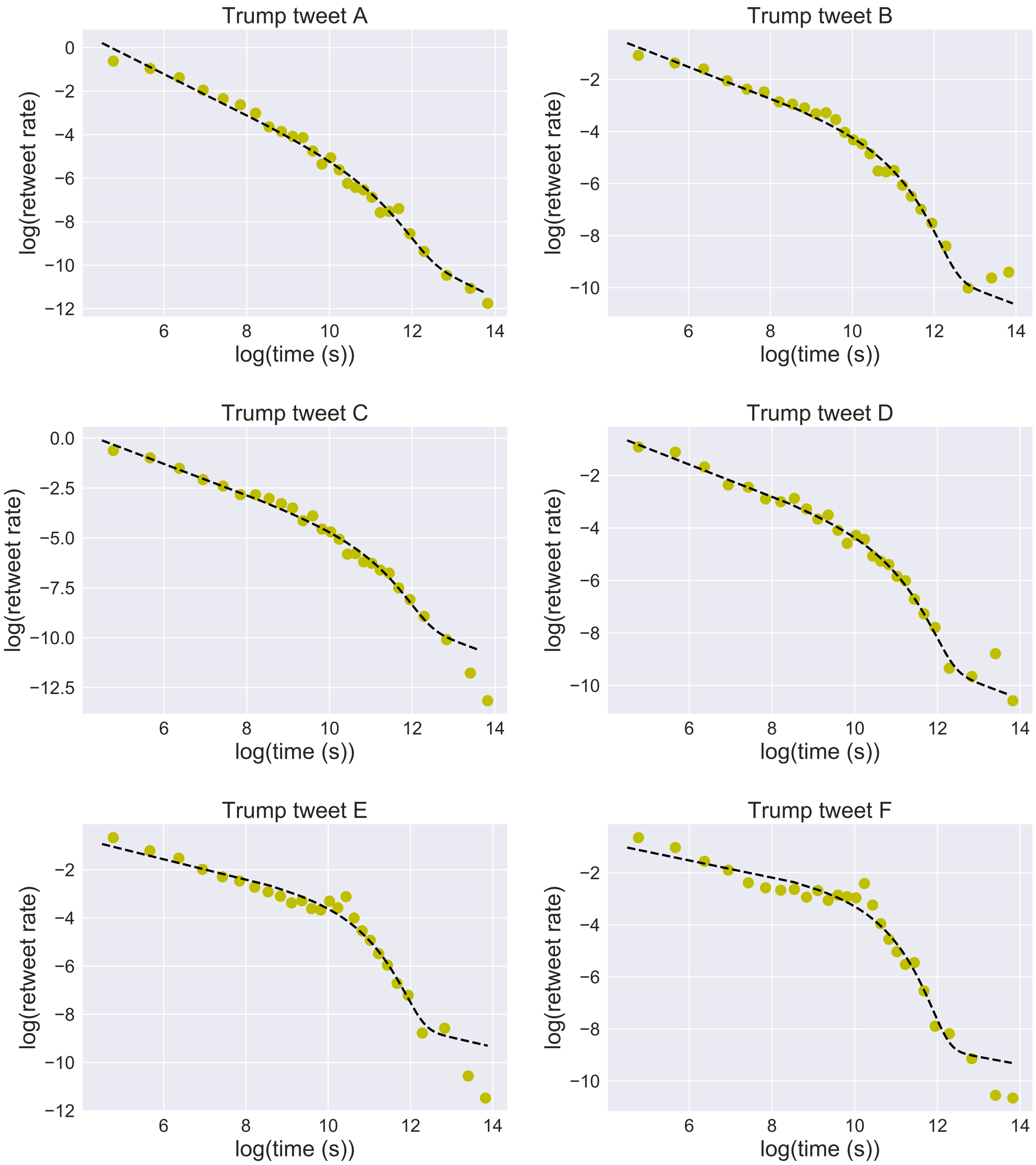


Figure 4.6: Donald Trump seed tweets: Long time duration retweet distribution. For time durations beyond one day and up to a month, the power law with exponential cutoff behaviour stops holding.

a tweet of a given age, then choosing to retweet. In Chapter 5 we give a proposed explanation of the power law and exponential cutoff, which lead to the first two models. The one-month time frame retweet rate model would have a different underlying mechanism, which we don't analyse in detail.

4.2.5 Diurnal effects and adjustment

For an initial tweet from our selected politicians in the United States, it appears that the retweet densities are somewhat correlated with the likelihood that people are awake in the US at that time. For example, for tweets by Donald Trump, the lowest rate of retweets occur at around 4 am US East Coast time.

To adjust for diurnal effects, we measure the overall tweet rate for tweets about a specific user, in this case Donald Trump, at six-minute blocks throughout the day, for a one-month period. We normalise these values and smooth the curve using a Savitzky-Golay filter [134], giving us the normalised tweet frequency, shown in Figure 4.7. To remove the diurnal effect for a retweet set, we "scale" the retweet count by the corresponding point on this curve. For example, if a retweet occurred at 05:00 UTC with a normalised retweet rate of 0.86, we would record $1/0.86 = 1.163$ retweets at this time. As we cannot have a fractional number of tweets, we record one tweet at this time and a second tweet with probability 0.163. This stochastic method allows us to account for changing activity during the day. Compared to other methods [56], ours has the advantage of giving a resultant set of discrete retweet times as output which we can subsequently use for statistical tests.

Figure 4.8 shows the diurnally adjusted data for the first 24 hours after our selected seed tweets for Donald Trump. The shapes of the curves are more uniform than without diurnal adjustment, with diurnal *dips*, in particular, being reduced. Table 4.5 gives parameters of best fit for the 24 hour retweet distribution, with diurnal adjustment. The exponential decay parameter is more consistent than without the diurnal adjustment.

4.2.6 Additional retweet datasets

We conduct a similar analysis for US politician Ted Cruz (shown in Appendix B.1) as well as US politician Marco Rubio and National Security Agency (NSA) whistle-

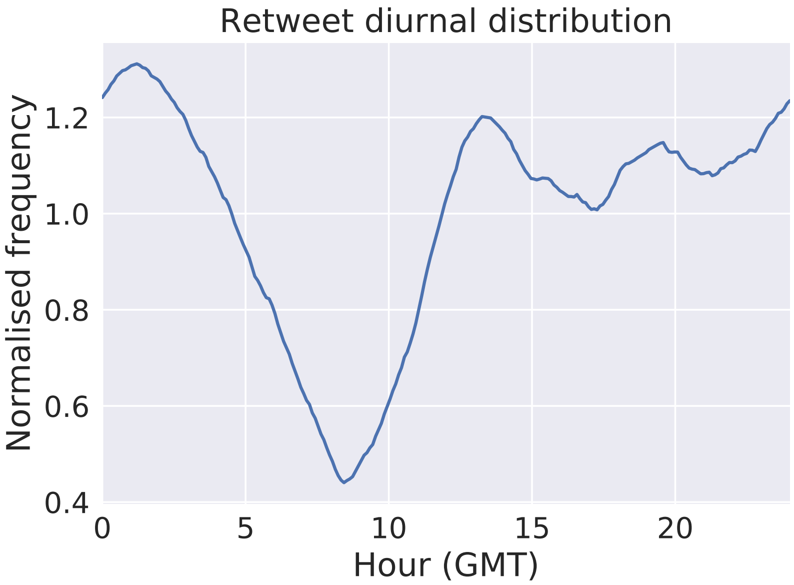


Figure 4.7: Retweet diurnal distribution for Donald Trump. The highest peak occurs in the evening in the United States (corresponding to 1 am UTC). The lowest point occurs in the early morning in the USA (9 am UTC).

Dataset	*A*	*b*	*c*
A	23.386	0.7559	3.041×10^{-5}
B	3.402	0.4711	2.791×10^{-5}
C	14.172	0.6700	2.665×10^{-5}
D	7.175	0.5913	1.985×10^{-5}
E	4.843	0.4986	1.347×10^{-5}
F	3.590	0.4334	1.774×10^{-5}

Table 4.5: Donald Trump seed tweets: Parameters for curves of best fit for 24 hour retweet collection with diurnal adjustment. The exponential decay parameter c is more consistent than we had without the diurnal adjustment in Table 4.3.

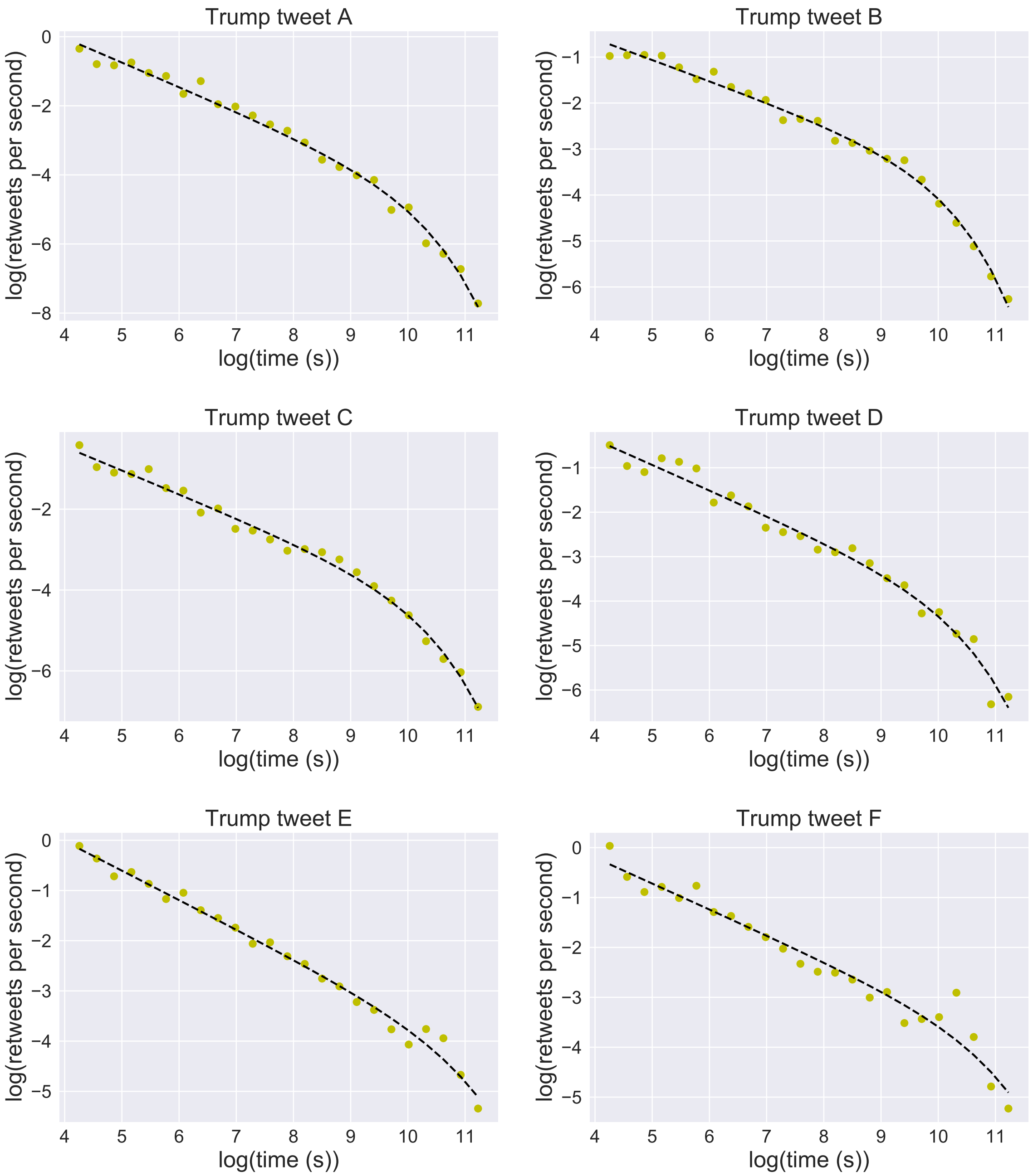


Figure 4.8: Donald Trump seed tweets: First 24 hours of retweet distribution, with a diurnal adjustment, on a log-log scale. The diurnal adjustment leads to more consistency in parameters, particularly the exponential decay parameter.

blower Edward Snowden. The resultant output distributions are similar, again indicating that the retweet rate density is well-modeled by a power law over the first three hours, and a power law with exponential cutoff for the first 24 hours. From these general trends we hypothesise that this retweet distribution behaviour applies more broadly, a claim which we statistically test in the following section.

4.3 Large scale data analysis

Graphically analysing selected examples gives visual insight into real world phenomena. However, in order to make population-level conclusions, we need to use more precise statistical methods on a larger, more diverse dataset. From analysing our example seed tweets in Section 4.2, we have two hypotheses:

1. The density of retweet rates decays as a power law over the first three hours.
2. A power law with exponential cutoff provides a better fit to the distribution of retweet rates than a power law.

4.3.1 Large scale data collection

In order to test our hypotheses on a large dataset, we collect retweets from the 100 Twitter users with the most followers [51] using the Twitter API. Details of these users can be found in Appendix B.3. We choose these popular Twitter users as their tweets will generally be retweeted frequently, providing more dense data. We used our Python scripts, described earlier in Section 4.2.1, to monitor and record the times of retweets from 314 seed tweets authored from 5 April to 9 April, 2016, using the Twitter REST API. In total, we collected 58,704 retweets, which are used for subsequent analysis. To enable replication of this work, tweet ids are available at the GitHub repository for this thesis[1]. Similar to analysing individual seed tweets in Section 4.2, let $D_i = \{t_{i1}, \ldots, t_{in}\}$ be the retweet set from the ith dataset, where t_{ij} is the number of seconds between the seed tweet and the jth retweet for the ith dataset.

The retweets from a new randomly chosen seed tweet were collected when our software identified that it had spare collection capacity. Consequently the tweets that we collected were not sequential, but were an assortment of tweets from the target

[1] https://github.com/pete1729/phd-thesis

users. Our datasets are consequently more likely to be from users who tweet more frequently. Dataset D_i is disregarded if tweet i was deleted within 24 hours after being published, if there were less than 50 collected retweets, or if the retweet rate was ever higher than 60 retweets within 60 seconds. The most popular authors of tweets, such as Taylor Swift, almost never have less than 50 total retweets, and regularly exceed 60 retweets within 60 seconds. Consequently this process of rejecting tweets creates a level of systematic effects in the seed tweets that were rejected from further analysis. For our collection process, even if a retweet dataset is immediately rejected, the dataset is labeled as *rejected* but the counter i is still incremented.

4.3.2 Fitting parameters to a power law by maximum likelihood estimation

For statistical analysis of a large dataset, we wish to use more precise methods for estimating power law parameters than the logarithmic binning-method used in Section 4.2. Here, we outline how to fit parameters to a power law with maximum likelihood estimation.

We now need to write likelihoods for power laws with lower bounds and potentially also with upper bounds. Following on from our definition of a power law in Section 2.6.1, a continuous power-law distribution has probability density $p(t)$ such that

$$p(t) = Ct^{-\alpha} \tag{4.4}$$

where C is a normalisation constant and $\alpha > 0$.

Power law with lower bound

As $t \to 0$, $p(t) \to \infty$. As the rate of a real event must be bounded, Equation (4.4) cannot hold for all $t > 0$. Hence for a real dataset, there must be some lower bound to the power law behaviour, t_{min} [33,131].

The probability density function for a power law with a lower bound is given by

$$p(t) = \frac{\alpha - 1}{t_{min}} \left(\frac{t}{t_{min}} \right)^{-\alpha}. \tag{4.5}$$

For a dataset from seed tweet i containing observations $\{t_{ij}\}$ with $t_{ij} > t_{min}$, we wish to find the decay rate parameter α that is most likely to have generated the data. The likelihood of observing the data, given the model, is given by

$$p(\{t_{ij}\}|\alpha) = \prod_{j=1}^{n} \frac{\alpha-1}{t_{min}} \left(\frac{t_{ij}}{t_{min}}\right)^{-\alpha}. \tag{4.6}$$

As the logarithm function is strictly increasing, the maximum of this likelihood function occurs at the same value of α as the maximum of the logarithm of the likelihood function. Taking the logarithm in this manner avoids arithmetic underflow. We have

$$\begin{aligned} \mathcal{L} = \ln p(\{t_{ij}\}|\alpha) &= \ln \prod_{j=1}^{n} \frac{\alpha-1}{t_{min}} \left(\frac{t_{ij}}{t_{min}}\right)^{-\alpha} \\ &= \sum_{j=1}^{n} \ln \left(\frac{\alpha-1}{t_{min}} \left(\frac{t_{ij}}{t_{min}}\right)^{-\alpha} \right) \\ &= \sum_{j=1}^{n} \left(\ln(\alpha-1) - \ln(t_{min}) - \alpha(\ln(t_{ij}) - \ln(t_{min})) \right). \end{aligned} \tag{4.7}$$

We then find the maximum likelihood estimate (MLE) by setting $\delta\mathcal{L}/\delta\alpha = 0$, or numerically with an appropriate computing package. For our analysis we use the *minimize* function from the Python 3.6 package scipy.optimize.

Power law with lower and upper bounds

Some distributions only obey a power law distribution over a certain range, or we have an upper cutoff of data collection. In these cases we wish to also set an upper bound on the power law distribution. We have

$$\int_{t_{min}}^{t_{max}} Ct^{-\alpha}\, dt = 1 \tag{4.8}$$

which gives

$$C = \frac{\alpha-1}{t_{min}^{1-\alpha} - t_{max}^{1-\alpha}}. \tag{4.9}$$

Substituting into (4.4) gives

$$p(t) = \frac{\alpha - 1}{t_{min}^{1-\alpha} - t_{max}^{1-\alpha}} t^{-\alpha}. \tag{4.10}$$

Similarly to the case where we only have a lower bound, we have log-likelihood function

$$\mathcal{L} = \sum_{j=1}^{n} \ln \left(\frac{\alpha - 1}{t_{min}^{1-\alpha} - t_{max}^{1-\alpha}} t_{ij}^{-\alpha} \right). \tag{4.11}$$

Again, for our analysis, we find the MLE with the *minimize* function from the Python 3.6 package scipy.optimize.

4.3.3 Clauset's test for power law distribution

We now test whether our retweet data meets a statistical test for a power law over the first three hours. Other authors have observed the power law behaviour in this period [3,49], but have not conducted thorough statistical tests.

We follow the procedure outlined by Clauset *et al.* [33] (discussed in Section 2.2.3), for determining whether a set of data can be considered to be drawn from a power law distribution. We conduct testing on datasets $\{D_i\}$, $1 \leq i \leq 314$. For each dataset D_i, we estimate the power law parameter α_i using maximum likelihood estimation, and also calculate the Kolmogorov-Smirnov (KS) statistic KS_i. We then generate synthetic power law datasets, $D_i^j, 0 \leq j \leq 100$, with scaling parameter α_i. For each synthetic dataset, we fit a power law model and determine the scaling parameter $\hat{\alpha}_i^j$ using maximum likelihood estimation. We calculate the KS statistic for each power law dataset based on its own model.

Much of the subsequent analysis in this chapter involves cumulative distribution functions (CDF), and testing the closeness of an empirical CDF to the CDF of a fitted theoretical function. An example of an empirical and theoretical cumulative distribution function for a sample retweet distribution is shown in Figure 4.9. On this plot, the KS statistic is the maximum difference between the empirical and theoretical distribution. As can be seen, the CDF of a power law with exponential cutoff is a closer fit to the empirical CDF than the CDF of a power law, and consequently has a

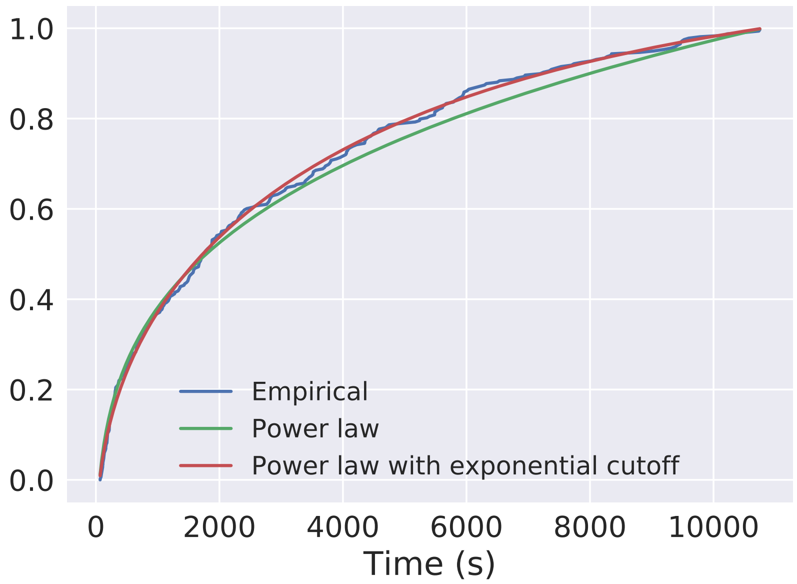


Figure 4.9: Example cumulative distribution function of retweet distribution for a sample tweet. The power law with exponential cutoff provides a closer CDF than a power law. The KS statistic for the power law is 0.0519 while the KS statistic for the power law with exponential cutoff is 0.0334.

lower KS statistic.

Our null hypothesis is that the data was generated by a power law distribution. To calculate a p-value, we set p to be the proportion of times when the KS statistic from the synthetic data is greater than from the empirical data. We follow Clauset's [33] choice of rejecting a power law if $p \leq 0.1$. This rejection can be interpreted as meaning that over 90% of the simulated datasets more closely matched the power law than our original empirical dataset.

We filter out datasets that are rejected for too high or low input rates, seed tweet deletion within 24 hours, or if the attempted power law fit fails. This leaves 158 datasets for the three-hour window and 157 datasets for the one-hour window. For the three-hour time window, only 66 of the 157 (42.0%) datasets passed the power law test. Reducing the time window to one hour, 100 of 158 (63.3%) datasets passed the test. Detailed numerical results of these statistical tests can be found in the GitHub page for this thesis[2].

These tests show that even in a relatively short time window after the inital tweet,

[2] https://github.com/pete1729/phd-thesis

Window size	Input datasets	Passed	% Passed
One hour	158	100	63.3%
Three hours	157	66	42.0%

Figure 4.10: Results of Clauset's test [33] for whether retweets occur as a power law. The datasets are not consistently passing the test. The one-hour dataset passes more often than the three-hour dataset.

sets of retweet times do not consistently pass the theoretical power law test. This is not unexpected as we are looking at real-world online data which typically contains more noise than a theoretical or simulated distribution. This finding is consistent with recent work from Broido and Clauset [22], who found that perfect power laws are rare in the real world.

4.3.4 Improvement of fit for power law with exponential cutoff

Finally, we test whether a power law with exponential cutoff provides a better fit to the distribution of retweet times than a power law. We use all collected retweets, up to a month after the inital seed tweet, and do not adjust for diurnal effects. In order to have sufficient data for thorough statistical testing, we use our Python scripts to monitor and record the times of retweets from an additional 1362 seed tweets using the Twitter REST API. These seed tweets were posted from 10 April to 4 May, 2016, and contain a total of 251,168 retweets. Combining these with the data collected in Section 4.3.1 gives a total of 309,872 retweets from 1676 tweets, which are used for analysis in this section. We label the new datasets D_{315} to D_{1676}. Of the initial 1676 seed tweets, approximately half were rejected, most commonly for the retweet rate being too high, leaving us with 808 datasets on which to conduct tests. For each dataset D_i, we determine the KS statistic for a power law, KS_i, and for a power law with exponential cutoff, KSE_i. We then calculate the difference between these values $KS_i - KSE_i$, and determine whether this is statistically significant.

A histogram of KS_i and KSE_i values is shown in Figure 4.11. The mean KS statistic value with exponential cutoff is 0.0508 with standard deviation 0.0230. This is significantly lower than the mean KS value of 0.0745 without the cutoff (32% improvement) and demonstrates an improvement in the quality of fit. Running a paired t-test on the two sets of data gives a p-value of 2.26071×10^{-157}. We therefore reject the null hypothesis that the samples have the same mean and conclude that the power law with exponential cutoff fits the data better.

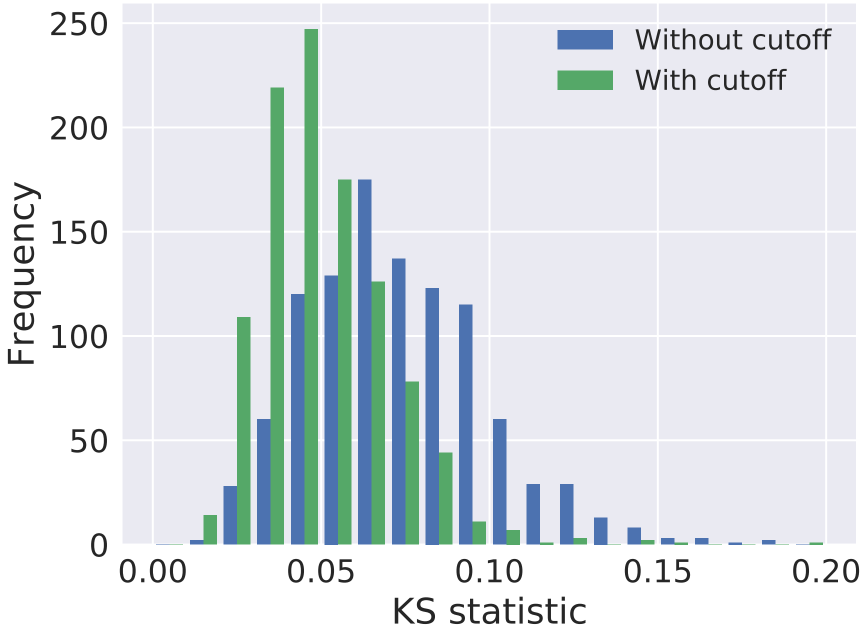


Figure 4.11: Histogram of KS-statistics. The power law with exponential cutoff generally has a lower KS-statistic than the power law without exponential cutoff.

We use the AIC criterion (defined in Section 2.6.10) to determine whether the improvement in the fit justifies the addition of the extra parameter. We calculate the average improvement in AIC score and percentage of datasets when we see improvement.

In order to achieve a smaller AIC value, adding an additional parameter requires an improvement in log-likelihood score of 1 to justify its inclusion. We consider the log-likelihood scores for the power law and power law with exponential cutoff and observe the increase in log-likelihood score.

Figure 4.12 shows the improvement in log-likelihood score by adding an exponential cutoff. Some datasets are well-modelled by a power law and only show a very small increase in likelihood score. However, other datasets benefit significantly by adding the cutoff. Changing from a power law to a power law with exponential cutoff improves the likelihood score by more than 1 in 558 of 808 tested datasets, 69.1% of the time. It improves the likelihood score by a mean value of 4.239. Consequently, adding an exponential cutoff improves the AIC score by a mean value of $(4.239 - 1) \times 2 = 6.478$.

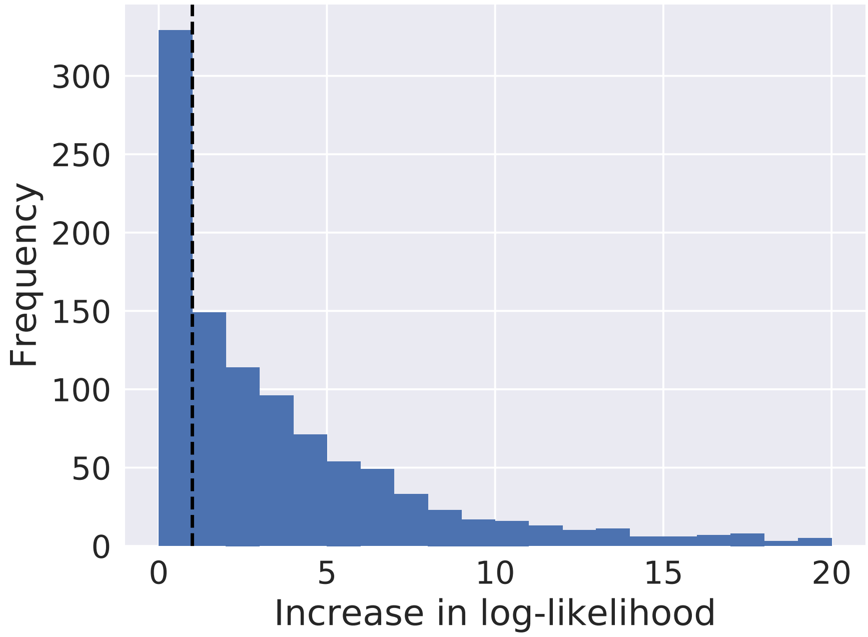


Figure 4.12: Histogram of improvement to log-likelihood changing from power law to power law with exponential cutoff. The black dashed line represents an increase in log-likelihood of one.

We therefore conclude that a power law with exponential cutoff is generally a better model for the rate of retweets than the previously used power law.

4.3.5 Power law parameter by topic

Human attention spans are of great interest for advertisers, who wish to understand if users will be interested in certain topics, and the duration of their interest. Our analysis of retweets provides insight into how human attention spans vary depending on the category of the topic, which in turn affects information propagation.

We analyse how the power law parameter varies by topic. Table 4.6 gives our lists of categorised Twitter users from the top 100 by follower popularity. The assignments are not necessarily unique and contain some level of subjectivity. The account @espn for example could have been put in the *Sports* or *Television* category (or both). Datasets are taken from $\{D_i\}, 1 \leq i \leq 1676$, as discussed in Section 4.3.4. For this analysis we use MLE to fit a power law over the first three hours of data to each dataset which met the requirements for at least 50 retweets and was not deleted in

Category	**Twitter username**
People / Groups	@priyankachopra, @MohamadAlarefe, @narendramodi, @SrBachchan, @khloekardashian, @iHrithik, @SnoopDogg, @shakira, @coldplay, @blakeshelton, @jimmyfallon, @BarackObama @Oprah, @kourtneykardash
Institutions	@instagram, @NASA, @twitter, @vine, @YouTube, @google
News outlets	@CNN, @nytimes, @TheEconomist, @BBCWorld, @cnnbrk, @BBCBreaking
Television	@MTV, @SportsCenter, @espn, @TheEllenShow
Sports	@NBA, @FCBarcelona, @NFL, @realmadrid

Table 4.6: Categories of selected Twitter users from the most popular 100 accounts.

Dataset type	**Number of retweet sets**	$\bar{\alpha}$	$\sigma(\alpha)$
People / Groups	92	0.7456	0.1676
Institutions	44	0.8125	0.0900
News outlets	482	0.8987	0.1178
Television	100	0.9335	0.1504
Sports	280	0.9405	0.1954

Table 4.7: Retweet power law parameter α from Twitter user categories.

the given time period. We have $t_{min} = 60$ and $t_{max} = 3 \times 60 \times 60$. Over this time period we do not expect that the curve shape will be significantly affected by diurnal cycles, nor do we expect to observe exponential cutoff behaviour.

Table 4.7 gives the average power law parameter over the category of user who wrote the seed tweet. As can be seen, the *Sports* category and *Television* category had the highest average power law parameters, while the *People / Groups* category had the lowest average parameters. We explain this high average parameter for the *Sports* category and *Television* category due to the fast moving nature of the genres. If the Barcelona Football Club (@FCBarcelona) tweet a halftime score of a soccer game, for example, that tweet will become largely irrelevant after the end of the game, and will have very few subsequent retweets. The category of *People / Groups* conversely has a much longer attention lifespan. If an influential personality, such as former US President Barack Obama (@BarackObama), tweets about reducing gun violence, this message will potentially be relevant for a much longer period of time. This potentially offers a method for classifying users, without resorting to expensive feature engineering or sophisticated machine learning methods (e.g. BotOrNot [38]).

4.4 Discussion and Conclusions

In this chapter we showed that the rate of retweets can be well-modelled by a power law with exponential cutoff, providing a better fit than a standard power law distribution.

Our retweet datasets did not consistently pass Clauset's test [33] for power law. However, Clauset's test is a very strict interpretation of what constitutes a power law and has a tendency to fail datasets that are elsewhere generally considered to be power law [40]. We conclude that a power law is a good model for the distribution of retweets over the first three hours, as has been found by other authors, even if it does not pass the strict statistical test.

Our stochastic method to conduct diurnal adjustment gives an output set of retweet times which can be then used for follow-up statistical analysis. This is a new technique which can potentially be used for many other purposes, when adjusting data for diurnal cycles or other cyclical trends.

Our analysis used the retweet times from the 100 Twitter users with the most followers. A natural question is whether similar retweet rate distributions would hold for all other Twitter users. This would be more difficult to statistically analyse as less followed accounts tend to have far fewer retweets. Future work could extend our analysis in Section 4.3.5, further investigating how power law and exponential cutoff parameters vary based on author, tweet topic or other factors. This will allow prediction of a tweet's propagation. We could also look at population-level social questions, e.g. how do decay parameters vary over the long term? As a society, are we growing more or less engaged with news from social media? As the tweet/retweet mechanism provides a continual source of information propagation data, it is possible to test theories that have been proposed in the social science literature using this experimental environment. In addition, there is an extensive amount of possible further work on analysing details of diurnal cycles, such as how they vary based on geography or the demographics of Twitter followers.

Chapter 5

Simulating Retweet Activity and Cascade Size Estimation

5.1 Introduction

The popularity of a social media post is a reflection of current social trends, and a key theme in social media research is understanding the popularity of online posts such as tweets [180]. Marketers aim for their material to become "viral" and want to understand the factors affecting information spread. The news media want to understand whether the public are interested in the content they create, and whether additional news stories on a given topic will be of interest [94]. One proxy for this is through measuring retweet cascades. (Prior work on retweet cascades is discussed in Section 2.3.)

In this chapter we simulate various aspects of the creation and spread of retweets, and then develop a method to estimate the total size of a retweet cascade. First, in Section 5.2, we determine the underlying mechanics for an appropriate human behavioural model that can explain the distribution of retweet times. We do this by exploring possible causes for the distribution of retweet rates being a power law with exponential cutoff and provide an explanation based on human prioritisation of tasks and loss of interest in topics over time. Consequently this chapter provides a link between social sciences and social media, as we show how human behavioural characteristics affect information propagation through the internet. In Section 5.3, we simulate the human behavioural process leading to retweets and show that a power law with exponential cutoff retweet distribution can be produced by a priority-based queuing model.

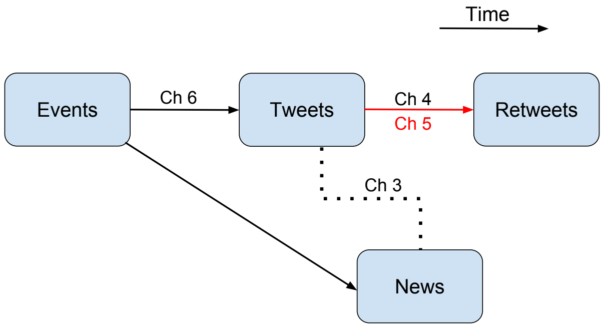


Figure 5.1: The analysis in this chapter relates to the temporal relationship between tweets and retweets (red link).

Using observations from our simulations coupled with additional social media theory on the Twitter distribution of followers, in Section 5.4 we create a method to estimate the size of the retweet cascade, focusing on retweets of news stories. Our method uses tweets from an initial time period and the topic categorisation of the tweet as possible input information. This is less information than is available, for example we do not use text details of the tweet or information about the followers of the user who created the initial tweet. To evaluate our method, we compare our predictions to the actual number of retweets on both our collected dataset and on a publicly available dataset. We show that our model can achieve accurate results for predicting the cascade size of tweets of news stories, with under 10% median absolute percentage error (MdAPE), using only the first hour of retweet data for training. This chapter relates to the temporal relationship between tweets and retweets and fits into the thesis structure, as shown in Figure 5.1.

This chapter makes the following key new contributions:

- Outlining how the power law observed in retweet rate decay can be explained by human prioritisation of tasking, while the exponential cutoff can be explained by human loss of interest in topics over time.
- Demonstrating through simulation that a priority-based tasking model can lead to a power law with exponential cutoff distribution of retweet rates.
- Developing a method to predict the size of retweet cascades, that is particularly effective for tweets of news stories.

5.2 A model for the distribution of retweet times

Among the process types leading to a power law discussed in Section 2.3.1, we argue that a priority-based queuing process is the most relevant to temporal retweet behaviour. The action of checking Twitter and deciding whether to retweet can be considered a task prioritised against other daily activities. The other causes of power laws discussed, preferential attachment or combination of exponentials, are not as relevant for our dataset, which is created by decision-based human activity. Although preferential attachment [14,44] is one of the more commonly discussed causes of power laws and ubiquitous in network science, the mechanics of this process are not aligned with retweet times. Although human loss of interest in topics over time is often modelled as exponential decay [2, 42, 90], there does not appear to be a plausible way to combine this with another exponential function to generate power law behavior.

We now propose a priority-based tasking theory of how people use Twitter in order to explain retweet times. Users will implicitly assign priorities to tasks in their lives and execute these tasks according to their internal perceived priorities [12]. Checking social media can also be considered a daily task, which competes against other life activities. For most users, checking social media is a low-medium priority task, out-prioritised by more urgent activites in life. Higher priority tasks will tend to be executed as soon as possible, leading to an exponential inter-event time distribution. Meanwhile, inter-event times between lower priority tasks will have a much heavier tail, as they often will be delayed while more urgent tasks are executed. The arrival of tweets to a user's account can be modelled by a Poisson arrival process. Consequently, based on the theory of prioritisation of tasks [12], the time between a tweet arriving and a user checking their Twitter account may have a power law distribution.

The second factor affecting the retweet distribution is loss of interest in topics over time, which as discussed in Section 2.3.2, can be modelled by exponential decay [2, 42, 90]. If the topic of the tweet is less relevant than when it was tweeted, it is less likely that it will be retweeted. The third and final component that affects the likelihood of a retweet is the proportion of users who decide to retweet. For our explanatory model, we assume that a constant proportion of users who see the tweet at a time when they think it is still relevant will decide to retweet.

If the distribution for an individual is power law with exponential cutoff, and a similar distribution holds for the entire population, the global distribution of retweets will be power law with exponential cutoff. Consequently it is possible to use our understanding of human prioritisation and the theory of heavy-tailed distributions to explain the phenomenon observed in Chapter 4. There might be alternative explanations for the cause of the power law with exponential cutoff. However, our explanation is simple, directly related to human behaviour, and explains the components of the phenomenon that we have observed in the empirical data.

5.3 Simulation of retweet activity

5.3.1 Priority-based tasking

We create a generative simulation model for retweets based on the priority-based tasking model. We assume that throughout the day, users have different tasks which they need to complete, and they therefore implicitly assign a priority to each of these tasks. At any time, the highest-priority uncompleted task will be undertaken and a queue of lower priority tasks may develop. For example, during the day a diligent student may have task categories of shopping, cleaning the house, completing university assignments and checking their social media. They may prioritise these events from highest to lowest as 1) assignment, 2) shopping, 3) social media and 4) cleaning. There may be more than one task in each category, for example multiple assignments which need completing. As the day progresses, more tasks may be added to the student's day.

Suppose at the start of this hypothetical day, the student has two assignments, two shopping tasks, three social media checking tasks and one cleaning task. They begin by processing the tasks in priority order, first completing the assignments, then shopping and then social media. Suppose that while completing the first social media task, the individual realises that they have another assignment that requires completion. As this student is diligent, the assignment task would be pushed to the front of the queue and executed before the remaining social media tasks. Due to the number of assignments, the cleaning task may not be executed on this day. This example leads to the sequence of events {assignment, assignment, shopping, shopping, social media, assignment, social media, social media}. The remaining cleaning task is not executed due to lack of time.

High-priority tasks such as assignments are completed soon after they arrive. Low-medium priority tasks such as checking social media are generally completed, but only if there are no competing higher priority tasks. This potentially leads to longer gaps of time between consecutive tasks at this level of priority. The very low priority tasks such as cleaning may not be executed at all.

We make many simplifying assumptions in our model, such as ignoring user cyclical activity such as sleeping or working. We also simplify all tasks to have the same average duration whereas in reality, some tasks take longer than others. Also, it is not expected that a real person would be able to prioritise tasks as efficiently as in our model.

Our human behavioural model is consistent with Barabasi's analysis on the origin of bursts [12]. The key addition is considering social media usage as a daily task, just like responding to emails, completing assignments, or any other task. To our knowledge, human interaction with social media has not previously been considered in this way.

5.3.2 Generative model for retweet times

Our generative model for human retweet behaviour assumes that tweets arrive to a user's account at random intervals according to a Poisson process. Other tasks arrive at random intervals also, and are queued. We make many simplifying assumptions in our model, such as ignoring user diurnal activity such as sleeping or working. We also assume all tasks have the same average duration whereas in reality, some tasks take longer than others. At any time, a user executes the highest priority task in the queue.

We simulate the following model. Parameters are chosen to be as realistic as possible, but while avoiding any unnecessary complexity. We selected a task completion time slower than the task interarrival time, to ensure that all tasks couldn't be completed. The simulation time is chosen as 365 days to smooth out any noise in the output.

- Distribution of times between arriving tweets: $\text{Exp}(1/300)$ (exponential with mean 5 minutes)

- Distribution of task interarrival times: $\mathrm{Exp}(1/255)$ (exponential with mean 4 minutes 15 seconds)
- Time to read one tweet and decide whether to retweet: 10 seconds
- Task completion time: 5 minutes
- Exponential cutoff parameter: 6 hours
- Number of different task priorities (tasks uniformly assigned to priorities): 10
- Retweet task priority: 3 (1 - lowest, 10 - highest)
- Probability of retweet: 0.1
- Simulation time: 365 days.

We call a task low-medium priority if it is one of the lowest priority tasks that will still be executed. We consider the gap between tasks of priority level 3, which is low-medium priority. The gap is the number of higher priority tasks executed between tasks of a similar type. For example, if a user checks social media as their 14th task of the day and next as their 37th task, the gap is $37 - 14 = 23$. Naturally, given tasks of equal duration, the task length gap is proportional to the expected delay time between tasks. Plotting this data on a log-log plot with linear bins gives Figure 5.2. The data falls neatly on a straight line, indicating power law decay. This plot corresponds to the distribution of task gaps between checking social media, a task which will be out-preferenced by other important tasks in an individual's life. This plot is consistent with the findings of Barabasi [12] and others, about how power laws can be generated by a decision-based process.

Figure 5.3 shows a simulation of retweet times (the time between tweet arrival and retweeting) for an individual. We plot the data on a log-log axis and fit a power law with exponential cutoff curve using the *curve_fit* function from scipy.optimize in Python 3.6. The fitted curve provides a close visual fit to the data, with equation $R(t) = 1108t^{-0.6004}e^{-1.082\times10^{-4}t}$, a power law parameter of -0.6004 and an exponential cutoff time of 2.56 hours. These are realistic parameters which could be reasonably expected from one of our empirical datasets analysed in Chapter 4.

The model and simulation confirms, for an individual, that a power law with exponential cutoff distribution can be produced by a priority-based queuing model. If we assume that a similar distribution holds for all users, then this confirms our proposed explanation of the distribution of retweet times in Section 5.2.

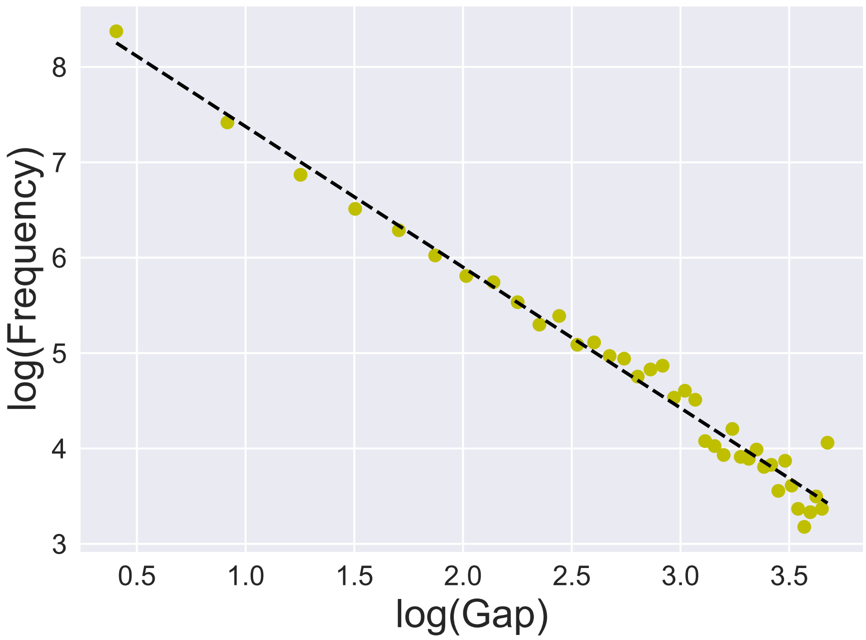


Figure 5.2: Simulated task delay log-log plot for low-medium priority task. The data points falling on a straight line suggests a power law distribution for the inter-event time.

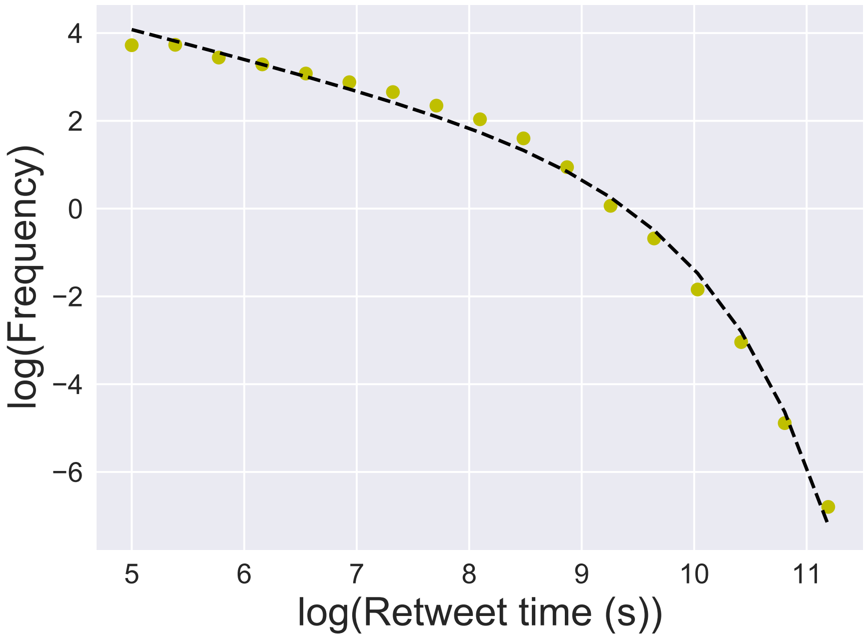


Figure 5.3: Simulated retweet times for low-medium priority task. A power law with exponential cutoff (dotted black line) provides a close fit to the dataset.

5.4 Estimation of retweet cascade size

We create a model to estimate the size of retweet cascades from high profile accounts such as news sources based upon a power law with exponential cutoff. The motivation for our method is that the number of followers of the intial seed tweet is usually several orders of magnitude greater than the number of followers of retweeters. Consequently, the rate of retweets for the entire cascade usually has approximately the same shape as the rate of retweets from the initial seed tweet. We can thus fit a power law with exponential cutoff model to the number of total retweets and use this to estimate the overall cascade size.

In this section we discuss the distribution of attention on Twitter and make observations on SEISMIC [180], a publicly-available retweet cascade dataset. We discuss boundedness of integrals of power laws, with and without exponential cutoffs, to provide further justification about why our method works. We then develop and test variants of our retweet cascade size estimation method.

5.4.1 Nature of retweet activity for news stories

As discussed in Section 2.3.3, the distribution of followers on Twitter is power law distributed with the vast majority of the Twitter population having a relatively low follower count. Consequently, the level of exposure for tweets will be most heavily dependent on the initial author and potentially, retweets by popular users.

Of the 3.2 billion tweets processed for the model SEISMIC [180], only 166,076 satisfied the conditions of having more than 50 retweets, no hashtags and being written in English. This is 0.0052%, or 1 in 19,000 tweets. The vast majority of tweets do not see more than 50 retweets as they are never exposed to a large enough audience, and are not infectious enough to explosively spread through the network. Also, we calculate that 9.8% of seed tweets with more than 50 retweets are authored by someone with fewer than 5,000 followers. This attention inequality of the SEISMIC dataset is consistent with analysis of Twitter by Zhu *et al.* [184], who found that 20% of users on Twitter get 93% of the retweets and the median number of followers for Twitter users is under one hundred.

Twitter accounts of leading news agencies have high numbers of followers, shown in Table 5.1. Even if a reasonably popular user with 1,000 followers retweets a story

News agency	Number of followers (millions)
CNN	40.5
NY Times	42.1
Reuters	19.8
Fox News	17.9
Washington Post	12.6

Table 5.1: Number of followers of selected news accounts on Twitter as of 5 June 2018. Leading news agencies are heavily followed on Twitter with over 10 million followers.

from a leading news agency, they are still only increasing the number of Twitter users exposed to the story by less than an extra 0.01%. Even when taking into account that individuals might perceive news stories differently based on whether it is being shared by a news handle or a private citizen, such a retweet will likely have minimal effect on the overall propagation of the news story. This observation becomes important as we consider population-level retweet cascades.

5.4.2 Boundedness

Here we justify why using a power law with exponential cutoff for cascade size estimation will produce a finite cascade size estimate, whereas a power law will not necessarily lead to a finite estimate. First, consider estimating the lifetime size of a retweet cascade $\hat{R}_\infty$ as the integral of a power law, estimating the total number of retweets $R_\infty(T_f)$ at some observed time T_f as

$$\hat{R}_\infty(T_f) = R_{T_f} + \int_{T_f}^{\infty} At^{-b} dt, \tag{5.1}$$

for some constant A and power law parameter b. The notation we use was outlined in Section 2.3.4, with $\hat{R}_\infty(T_f)$ denoting the estimate of the total number of retweets using information up to time T_s and R_{T_f} is the number of retweets up to time T_f. For $A > 0$ and $0 < b < 1$, this integral diverges. The proof of this is straightforward and is given in Appendix C.

Other authors have considered modelling the retweet share rate as a power law [1]. However, given that most tweets have power law parameter less than one, as we observed in Chapter 4, it is not possible to generate a finite estimate of the total cascade size by simply integrating a power law. However, if we add an exponential parameter and take the integral from some training time T_f, we have

$$\hat{R}_\infty(T_f) = R_{T_f} + \int_{T_f}^{\infty} At^{-b}e^{-ct}dt, \tag{5.2}$$

which, as we prove in Appendix C, is bounded for $A > 0$, $b > 0$ and $c > 0$. Consequently, by modelling the rate of retweets as a power law with exponential cutoff we can generate a finite estimate for the number of retweets.

5.4.3 Cascade size estimation method

We conduct our retweet cascade size estimation from the initial seed tweet and the measured response rate, with power law parameter estimated from the interval $[T_0, T_f]$. We test on real datasets where we have the lifetime retweet information, but only train the model with data up to some cutoff time T_f. The longer the data is observed, the more accurately the model can predict the total number of retweets. We estimate the total retweet count using Equation (5.2) and need to determine appropriate parameters for the model.

For very short intervals of time after the seed tweet, the power law behaviour does not hold as the probability density function tends to infinity, so we cannot select $T_0 = 0$. However, the higher the value of T_0, the less data we have for fitting the power law, so it is necessary to find a compromise. Experimentation on our datasets has shown that values between $T_0 = 30$ seconds and $T_0 = 60$ seconds work effectively. Choosing higher values of T_f gives a closer but less useful approximation, as our goal is to estimate the size of the cascade as early as possible. The number of retweets in the interval $[T_0, T_f]$ can be small, often fewer than 50 retweets. Consequently the estimation of the power law parameter is inexact so we bound it in a window $[b_l, b_u]$ = $[0.6, 1.1]$, centered around the empirical mean of retweet rate parameters $b = 0.85$, from Chapter 4. Doing this avoids excessively high or low estimated retweet rate parameters affecting the cascade size prediction.

In order to simulate and predict the number of retweets, we need to determine (or select) the power law decay and exponential cutoff parameters. In each method, the exponential cutoff parameter c is fixed, and the leading constant A is determined by the number of tweets in the training time window. The power law parameter is either taken as a constant or measured from the initial retweet rate decay. When determining power law parameters, we optimise a doubly-bounded power law over a selected region where we have data. The exponential decay parameter is taken as a constant $c = 0.00008$, the average rate of exponential decay measured from our full dataset in Chapter 4. The leading constant A is determined by dividing the number of retweets in the selected time period, by the integral of the power law with exponential cutoff function. We conduct our analysis using a range of methods to determine the power law parameter b, as follows:

- b_A: Fixed (Method-A)
- b_B: Determined by category of tweet (Method-B)
- b_C: Estimated by fitting to data in time window $[T_0, T_f]$, limited to range $[b_l, b_u]$ = $[0.6, 1.1]$ (Method-C)
- b_D: $(b_A + b_C)/2$, the average of the fixed and category parameters (Method-D)
- b_E: $(b_B + b_C)/2$, the average of the category and fitted parameters (Method-E)

The tweets are split into the categories: News, Sports, Television, People or Institutions, as in Table 4.7 in Section 4.3.5. We measure the performance of the methods by mean absolute error (MAE), mean absolute percentage error (MAPE) and median absolute percentage error (MdAPE) (defined in Section 2.6.2).

We initialise the model from data over an initial time period $[T_0, T_f]$. The higher the value of T_f, the more data used in this initialisation process, and the more accurately we can predict the retweet rate power law parameter. We select the lower bound as 30 seconds and the upper bound as $T_f = 15$ mins or $T_f = 60$ minutes, consistent with other authors [83]. For our chosen T_f values, it is not possible to accurately estimate the exponential cutoff parameter, as the cutoff has minimal effect in this time period.

5.4.4 Simulating retweet rates from a single example tweet

We show plots from a single example tweet to illustrate how the expected number of retweets varies over time, and demonstrate convergence of the estimation. We use a sample tweet from the *New York Times*, with details given in Table 5.2. This is a typical news tweet about a musical nominated for a record number of awards. The tweet was posted at 1:43 UTC, corresponding to the evening in the United States, the heaviest period of retweet activity during the day.

Author	CNN
Text	"Hamilton" has been nominated for a record 16 Tony Award nominations http://cnn.it/1Z7thoZ
Date	3 May 2016 - 1:43 PM (UTC)
Tweet ID	727493777707941889
Number of retweets	350

Table 5.2: Details of sample tweet from CNN in May 2016, with 350 retweets.

We show the expected number of retweets over time if we fit a power law to data in the first hour, using Method-C. We set our training window $[T_0, T_f] = [30, 3600]$. The estimated power law parameter in this range is $\hat{b} = 0.9510$, which lies within our range bounds $[0.6, 1.1]$ of reasonable power law values, so we keep this value for our cascade size prediction.

In Figure 5.4 we show the total number of predicted retweets both with and without an exponential cutoff. Here, we can clearly see that the curve with an exponential cutoff is converging while the curve without an exponential cutoff does not appear to be converging. Through our simulation, we can determine that for this example dataset, the expected total number of retweets is 364.15. This is 4.0% more than the actual retweet count of 350.

5.4.5 Experimental results on larger dataset

We test on the retweet datasets $\{D_i\}$, outlined in Section 4.3.1, which is one month of collected retweet data from each seed tweet. This timeframe captures the vast majority of lifetime retweets. In particular, we focus on tweets from news sources, keeping to the theme of this thesis about the temporal relationship between events, news and the associated twitter response. We also test on data from SEISMIC [180], which is publicly available.

Results from testing our method on retweet datasets $\{D_i\}$ are shown in Table 5.3. Method-A, Method-B, Method-D and Method-E all performed similarly well with MdAPE between 18.85 and 23.50. However, using only the data in the fixed window to estimate the power law parameter (Method-C) performed significantly worse with MdAPE of 36.39. This would suggest that the relatively short training interval of 15 minutes is insufficient to accurately estimate the power law parameter.

We repeat the analysis using a 60-minute training time window, with results shown

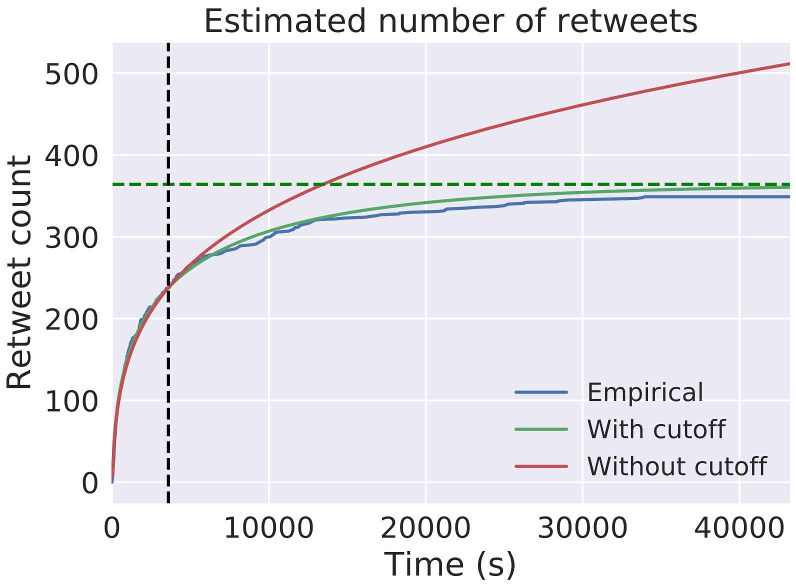


Figure 5.4: Example of the expected number of retweets for Method-C both with and without an exponential cutoff for the sample dataset. The vertical dotted black line shows the time cutoff where the prediction is made, while the horizontal dotted green line shows the asymptote of the tweet count estimation 364.15.

Method	MAE	MAPE (%)	MdAPE (%)
A	57.52	24.29	18.85
B	59.03	23.99	19.06
C	72.85	46.77	36.39
D	49.94	26.16	23.50
E	46.32	23.07	19.81

Table 5.3: Accuracy of various cascade size prediction methods for a 15-minute training window on tweets about all topics. Method-C is the least accurate, indicating that this relatively short training window is insufficient to determine the power law parameter.

Method	MAE	MAPE (%)	MdAPE (%)
A	38.70	13.95	10.83
B	38.71	13.36	9.79
C	30.51	12.91	9.66
D	33.35	12.44	10.04
E	32.79	11.82	9.24

Table 5.4: Accuracy of our cascade size prediction methods for a 60-minute training window. All methods perform similarly well with MdAPE between 9.24 and 10.83. Method-C recorded the best MAE score while Method-E recorded the best MAPE and MdAPE scores.

Method	MAE	MAPE (%)	MdAPE (%)
A	30.25	17.36	13.46
B	34.60	19.10	15.35
C	42.61	37.39	34.78
D	27.29	20.49	17.28
E	26.64	18.68	15.04

Table 5.5: Accuracy of our cascade size prediction methods for a 15-minute training window for tweets from news sources. Method-C performs the worst, demonstrating that a power law cannot be accurately fitted in this short time window.

in Table 5.4. A significant improvement in accuracy has been achieved, with MdAPE scores ranging between 9.24 and 10.83. In this training window length, estimating the power law parameter from the data (Method-C), performs much better than it did for the 15-minute training window.

We repeat the analysis only with tweets from news sources, providing a link between this part of the thesis with Chapter 3, which concerned the temporal relationship between news and tweets. The results of this experiment are shown in Tables 5.5 and 5.6. For the 15-minute window, Method-C, determining the power law parameter from the data, performs significantly worse than the other methods. This suggests that the data is too noisy in this short period to meaningfully estimate the power law parameter.

For the 60-minute window results shown in Table 5.6, the best MdAPE result comes through method-B, with the retweet power law parameter determined form the category of the tweet. The low score of 8.07 indicates that Method-B performs well and that the response to news service tweets is very predictable. Interestingly, the best MAE result comes from Method-C while the best MAPE result comes from Method-E. This variation on performance depending on the choice of metric is not uncommon [73], and highlights the difficulty of determining which prediction or

Method	MAE	MAPE (%)	MdAPE (%)
A	19.72	10.78	8.10
B	21.46	11.26	8.07
C	17.55	11.67	10.00
D	17.77	10.44	8.81
E	18.18	10.28	8.21

Table 5.6: Accuracy of our cascade size prediction methods for a 60-minute training window for tweets from news sources. The mean average error scores for all methods are better than for performing the method on all sources.

forecasting algorithms are the most effective. However, all differences are reasonably minor and so we would suggest using the fixed power law parameter as in method-A, as this performs well over any training period.

We can predict the cascade size for tweets from news sources significantly more accurately than we can for general tweets, this is likely to be because Twitter accounts for news sources have a high number of followers and a generally predictable public response. Figure 5.5 shows how one of our error metrics, median absolute percentage error, improves as the training time increases, for news tweets. As the training time becomes very large, all methods tend to achieve a similar estimation error.

5.4.6 Testing on public cascade datasets

In order to further test our method, we conduct retweet cascade size prediction on the dataset from SEISMIC [180] for tweets from accounts with more than 1,000,000 followers. The SEISMIC dataset consists of 166,076 retweet sets, so it is infeasible to manually classify each seed tweet into a category. Consequently we omit Method-B and Method-E. The results of our cascade size estimation, with a one-hour training window, are shown in Table 5.7. Note that the prediction is not as accurate as on our dataset $\{D\}$ composed of tweets from authors in the top 100 of popularity. This is expected, as the higher the number of followers, the more predictable the retweet response.

Direct comparison between methods such as SEISMIC, Dynamic Poisson Model (DPM) and Reinforced Poisson Model (RPM) is less meaningful as they do not function correctly on all datasets, so each method chooses on which datasets to make a prediction. For example, DPM fails to make a prediction for 5.79% of tweets after 60 minutes, RPM fails to make a prediction for 5.69% and SEISMIC fails to make a

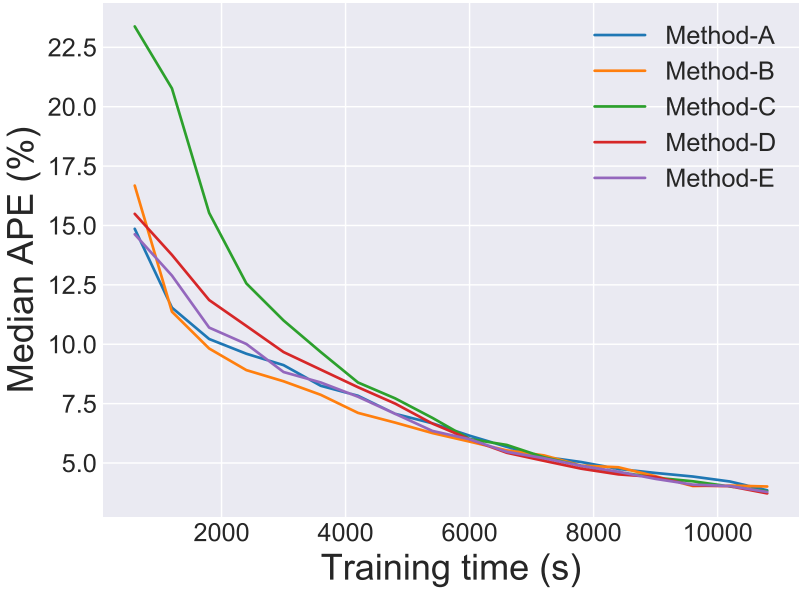


Figure 5.5: Change in MdAPE as the training time is varied. For small training windows, Method-C, fitting a power law to the training points, performs worse than the other methods. As the training time increases, all methods tend to have similar MdAPE values.

Method	MAE	MAPE (%)	MdAPE (%)
A	193.00	19.48	15.63
C	176.50	18.40	13.39
D	178.80	17.44	12.65

Table 5.7: Accuracy of our cascade size prediction on SEISMIC dataset, for tweets authored by users with at least one million followers, after a one hour training window. The retweet cascade size prediction is not as accurate as for our own dataset.

prediction for 1.29%. Additionally each method has varying objectives, often aiming to detect "break-out" cascades, an important problem for trend detection. For reference, however, state-of-the-art cascade-size prediction method SEISMIC achieves a MdAPE of 15% on their own dataset after a one hour training window.

5.5 Discussion and conclusions

To add a theoretical foundation to the numerical analysis in Chapter 4, here we provided an explanation of the power law with exponential cutoff behaviour. The power law component is explained by the time until the user checks their social media, which is governed by a priority-based queuing process. The exponential cutoff is explained by the loss of interest in topics over time. The model that we have produced gives an explanation of the phenomena that governs the spread rate of information online through Twitter. It builds upon previous work on the burstiness of human behaviour to give a better understanding of information flow in a social media system.

We simulated the process of retweets, both from an individual behavioural level up to a population level. Through simulation, we demonstrated that a priority-based queuing process can produce a power law with exponential cutoff distribution of retweet rates, similar to that observed in Chapter 4. It would be possible to extend this model by including diurnal cycles and reducing the number of simplifying assumptions. For example, when someone is sleeping they do not check their social media for around eight hours. Also, commuters taking public transport to work in the morning may be more likely to check social media. The model could be further extended by more precisely modelling the durations of tasks to be executed. Our model assumes that all tasks have the same duration, but in reality users tend to spend more time on some tasks than others.

Another possible extension to this chapter would be to implement the simulation with a more sophisticated tasking method, such as is used for high performance computing [127]. Social media could be treated as a short minor task which is squeezed into blocks of time which are too short for larger tasks. Despite being more complex and having more parameters, this could potentially be a more accurate model of how people consider checking their social media.

Since our retweet simulation was implemented, the user experience on Twitter has changed slightly, with increased levels of recommended tweets for users, including tweets which followed accounts have liked. Previously tweets would occur in a user's timeline only from accounts which the user is following and in strict chronological order. Such changes in social media interfaces are very common and could potentially change results. It is not expected that this would have a substantive impact on our findings, but this would have to be confirmed with additional analysis.

Our cascade-size estimation method is an effective way to estimate the number of retweets from an initial tweet after observing the retweet behavior for a short time period. On our dataset D which consists of retweets from users in the top 100 of popularity, Method-E achieves very accurate results with median absolute percentage error of 9.24%. In particular, after observing the response to news tweets for the first hour, we can estimate the size of the retweet cascade with median absolute percentage error of 8.07%. These results demonstrate that the Twitter response to news tweets is more consistent than for other types of tweets.

One of the challenges which we have not addressed in this chapter is retweets by bots. Some Twitter users will attempt to boost their apparent popularity by having bots retweet their tweets, often for a fee. Future work could use bot detection methods, e.g. BotOrNot [38], to predict whether a tweet is by a bot or by a human, then repeat the analysis with the bot tweets removed.

It may be possible to improve our method by using additional features. However, we refrain from doing so, as our goal is to demonstrate how simply using a power law with exponential cutoff as the decay function from the initial tweet performs adequately, rather than creating a sophisticated retweet cascade model. Additional possible work would be to incorporate our model's decay function into other retweet cascade size prediction methods to test whether performance is improved.

So far in this thesis, we have focused solely on the temporal aspects of Twitter, without considering the textual content. We turn to this important aspect of Twitter in the next chapter, when we consider the relationship between events and tweets.

Chapter 6

Event Detection and Time Estimation from Twitter

6.1 Introduction

Postings to social media platforms are increasingly used to extract useful information about real-world events. Journalists use platforms such as Twitter to summarise breaking news stories [94], while governments are interested in mining social media to provide early warning of events such as disease outbreaks [140], civil unrest events [156] and even natural disasters such as earthquakes [132]. Developing methods to summarise the large volumes of information generated on social media by such events is therefore of great importance for scientists and end-users alike, and an extensive amount of literature has appeared on real-time microblog summarisation and event detection. We discuss such previous work in Section 2.4.

A key component of automated microblog summarisation is event time estimation. For applications such as automated news production, simply knowing the keywords or topics associated with events is insufficient; it is also necessary to estimate the time of events as accurately as possible. Unlike retweets, where a user only has to hit a single button, it takes time for users to decide what to tweet, then physically input the contents of the tweet into some Twitter user interface, such as a mobile phone application, then send the tweet. Consequently, unlike retweets, the highest rate of response tweets will not occur in the initial seconds following an event, since under regular conditions (excluding bots etc.) users cannot generate tweets that quickly.

In this chapter, we develop a new algorithm for clustering microblog posts authored in response to real-world events. Our algorithm, *Social Media Event Response Clus-*

tering (SMERC) [102], detects events in near real time and estimates the time of the event. It is unsupervised, without requiring prior specification of the types or number of events. Unlike many previous methods, our algorithm uses both the content and timing of messages to estimate the likelihood that pairs of messages are related, giving more meaningful clusters of tweets associated with events. The rationale for our method to convert the time difference between tweets to an associated probability is described in Section 6.3.1. Our approach also provides insight into the mechanisms that lead to a social media response to a particular event, and allows us to model the distribution of response times. We describe our clustering algorithm in Section 6.3.2, and explain why we choose key components such as cosine similarity and affinity propagation.

To demonstrate our method, we conduct event detection for cricket and Australian Rules Football games, with the goal of automatically creating a list of events and associated event times. In Section 6.3.3 we test our method on collected datasets and demonstrate the results. While event-detection techniques can be applied to any type of events, sporting events are ideal to collect and study as the time, location and hashtags are generally known beforehand and the collective attention of often large audiences is focused on the in-game action [59,147,181]. For these reasons, automatic detection of key events in sporting contests has been a longstanding goal in academic research [114, 130]. With the widespread adoption of social media, researchers can now analyse fans' perspective on which events in the game are considered important. Furthermore, online reaction to sporting events can be useful case studies on human behaviour, with heavy levels of engagement and emotion, generally organized around the success of opposing teams. For evaluation, we apply SMERC to three Twitter datasets, demonstrating that clusters around the detected events are meaningful and that event detection is improved by temporal adjustment.

In Section 6.4 we develop and demonstrate a method to estimate the time of an event, given an associated set of tweet times. Combining the clustering algorithm and the event time estimation allows more complete microblog summarisation: tweets in response to the event are clustered together, then the time of the causal event is estimated. As indicated in Figure 6.1, this chapter fits into the thesis structure as it involves the temporal relationship between events and tweets.

This chapter makes the following new contributions:

- Showing that the time intervals between pairs of related messages are exponen-

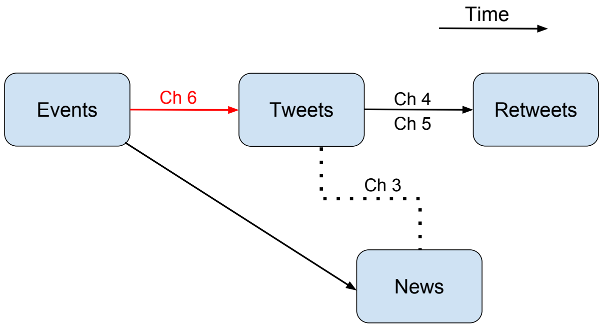


Figure 6.1: The analysis in this chapter relates to the temporal relationship between events and tweets (red link).

tially distributed.

- Presenting a novel way to conduct tweet clustering by incorporating both textual and temporal information.
- Developing a method to estimate the time of events given an observed Twitter response.

6.2 Data collection methodology

We collect tweet data from both Australian Football League (AFL) games and cricket games, the latter including the Big Bash League (BBL) and the Women's Big Bash League (WBBL), using the Twitter streaming API. The hashtags used and number of tweets collected are outlined below. As with many sporting events, cricket matches and AFL games have specific hashtags, usually the names of the teams involved, publicised by the clubs and supporters prior to the event. While our method is for tweet clustering generally, sporting events are ideal for the present study as data collection can be planned in advance. Our collected data was stored in a MongoDB database and later processed using custom Python 3.6 scripts.

AFL has different characteristics to most other sports as it is fast-paced and events are not always clearly defined. In many other sports such as cricket or the NFL (American Football), many major events (touchdowns, penalties, etc.) occur at a clearly defined time. In the AFL however, an inexperienced observer may watch a period of play without realising that an influential event has occured. For example, the umpire may have missed a *holding-the-ball* decision, potentially infuriating the audience. In this manner, AFL is more closely aligned with real life, where events happen at arbitrary times and the significance of events is subjective. Due to this difference, AFL games provide a challenging dataset for attempting to automatically determine key events from the crowd's perspective.

We collect tweets from AFL games in 2016 and 2017 using the hashtag #AFL{*Team1Nickname*}{*Team2Nickname*} where the nicknames are drawn from the list {Crows, Lions, Blues, Pies, Dons, Freo, Cats, Suns, Giants, Hawks, Demons, North, Power, Tigers, Saints, Swans, Eagles, Dogs} representing each AFL club. Not every tweet about a game contains these hashtags, but using this filter greatly simplifies the collection and provides a mapping to games of interest. There exist some *content-polluting* or *hijacking* [54] tweets about other topics using these hashtags, for example tweets used for advertising purposes, but these are a minority of the total set of tweets. In cases of bots repeatedly tweeting the same text, we manually remove these content polluting tweets. This collected data is used for both testing our clustering method SMERC and estimating the times of events from the Twitter response.

We also collect cricket tweets over the Australian 2017/2018 summer with the hashtags #BBL07 and #WBBL03. These hashtags were regularly publicised by the television broadcast of the event, encouraging fans to use these instead of inventing their own hashtags. Naturally, such uniformity in social media activity is useful for data collection.

6.3 Clustering using textual and temporal information

6.3.1 Relative probability of tweets being related over time interval

To illustrate our temporal adjustment method, we closely analyse the data from three Twitter datasets, chosen for diversity in sports, gender, and time collection period:

- Dataset Z1: Tweets from the Australian Football League preliminary final match between the Adelaide Crows and the Geelong Cats on 22 September 2017.
  Hashtag: #AFLCrowsCats
  Number of tweets collected: 5,018
  Collection timeframe: 3 hours

- Dataset Z2: Tweets from the 2017/18 Women's Big Bash (the Australian women's domestic Twenty20 cricket tournament) opening weekend on 9/10 December

2017.
Hashtag: #WBBL03
Number of tweets collected: 5,393
Collection timeframe: 48 hours

- Dataset $Z3$: Tweets from the 2017/18 Big Bash (the Australian men's domestic Twenty20 cricket tournament) between Brisbane Heat and Melbourne Stars on 20 December 2017.
  Hashtag: #BBL07
  Number of tweets collected: 3,153
  Collection timeframe: 4 hours.

We manually labelled tweets from our datasets to examine how the probability of tweets being related depends on the time interval between them. As we show, this decays exponentially; an insight that forms a critical part of the clustering algorithm SMERC which we develop in Section 6.3.2.

Our methodology to analyse the temporal relationship between related tweets is as follows:

1. Collect a series of tweets with a selected hashtag over a period of time.
2. Manually identify a set of on-field events (scoring events, penalties, controversial umpiring decisions etc).
3. Manually label tweets that are in response to these events.
4. Record the time intervals Δt_{ij} between all pairs (i, j) of tweets where both tweets are in response to the same manually labelled event.
5. Record the time intervals Δt_{ij} between all pairs (i, j) of tweets where only one of the tweets is in response to a manually labelled event, while the other is unrelated.
6. Bucketise the time interval data and count the number of related pairs of tweets and unrelated pairs of tweets within each bucket.
7. Calculate the probability that a pair of tweets within a bucket is related, and examine how this varies with Δt.

We first analyse dataset $Z1$, the response to events in an AFL game. Manually classifying the 5,018 tweets, we detected 11 key events consisting of 468 tweets. Pairing the tweets gave 18,175 pairs of related tweets and 747,779 pairs of unrelated tweets. We fit a density curve to the data using the *KernelDensity* package from the sklearn.neighbors module in Python 3.6 [122], selecting the Gaussian kernel[1]. The resultant density curves of time intervals between pairs of related and unrelated tweets are shown in Figure 6.2. The density of time intervals between unrelated tweets remains approximately uniform, while the density of intervals between two related tweets about a given event decays over time. We define the intersection point between these two curves as the *turnover time*. Before this time, any other tweet in the dataset is more likely to be related to a given tweet whereas after this time, any other tweet is more likely to be unrelated. For this dataset, the turnover time $t = 158$s.

To determine the probability of tweets being related to the same event, given the time separation, we bucketise the data and compute the ratio of related pairs of tweets to the overall number of pairs within each bucket. We plot this on a log-linear plot, in Figure 6.3, showing a roughly straight line, corresponding to exponential decay. The curve has a slope of -0.0106, meaning that a suitable model for the probability that tweet pairs with time interval Δt are related is $Ce^{-0.0106\Delta t}$, where C is a constant.

In order to demonstrate that this property is not unique to AFL and instead holds across different sports, we repeat here the same analysis for the WBBL dataset $Z2$. Manually clustering the 5,393 tweets in $Z2$, we detected 23 key events consisting of 348 tweets. Pairing the tweets gave 7,430 pairs of related tweets and 50,431 pairs of unrelated tweets. As shown in Figure 6.4, we again see that the density of time intervals between unrelated tweets remains approximately constant with Δt, while the density of time intervals between two related tweets about a given event decays over time. For this dataset we have turnover time $t = 335$s, longer than for $Z1$.

Bucketising the data and plotting the probability curve in Figure 6.5 shows a similar curve as for dataset $Z1$. The fitted curve has slope -0.0069, which is slightly less steep than for the AFL dataset $Z1$. The slight difference in slope could potentially be explained by the slower nature of cricket, with key milestone events being discussed for relatively periods of time. The linear relationship for both datasets indicates that the change in probability of tweets being related has an exponential relationship with the time interval between them. This is noteworthy, due to the highly different nature of action for the two sports. Similar to baseball, cricket is a fundamentally *discrete*

[1] From tests with the *tophat* and *Epanechnikov* kernels [47], we note that the curve shape is roughly independent of the choice of kernel.

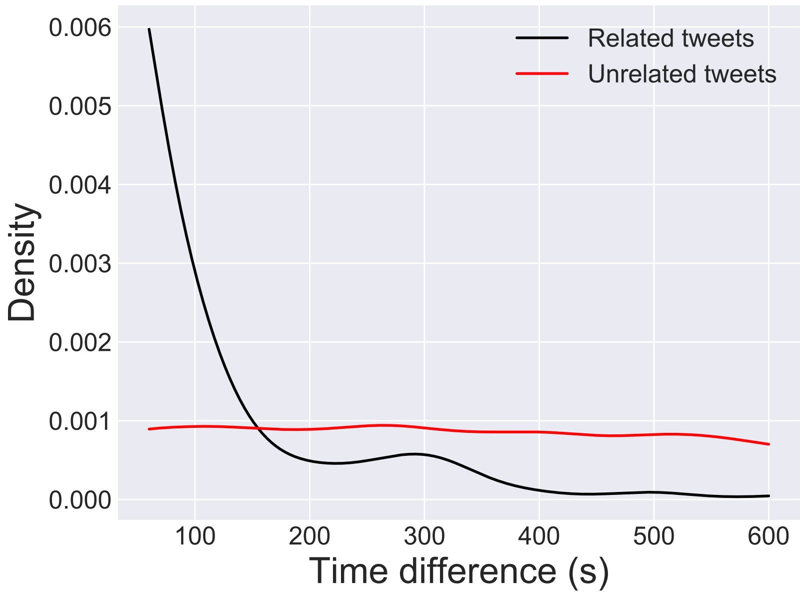


Figure 6.2: AFL dataset $Z1$: Density of time differences between pairs of related and unrelated tweets. The density of time differences between pairs of unrelated tweets stays roughly constant, while the density of time differences between related tweets decays. The slight bump at around 300 seconds is likely to be due to noise.

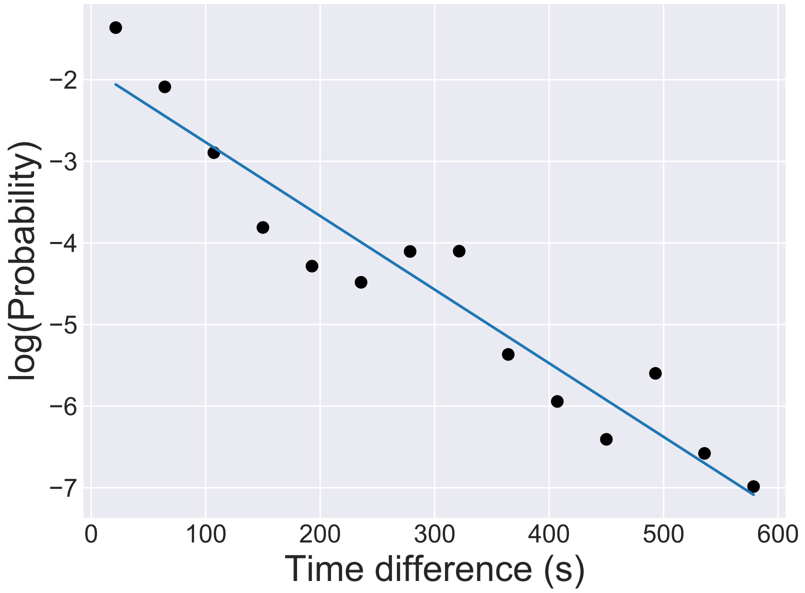


Figure 6.3: AFL dataset $Z1$: Log-linear plot of the probability of tweets being related, given the time separation. The straight line indicates exponential decay.

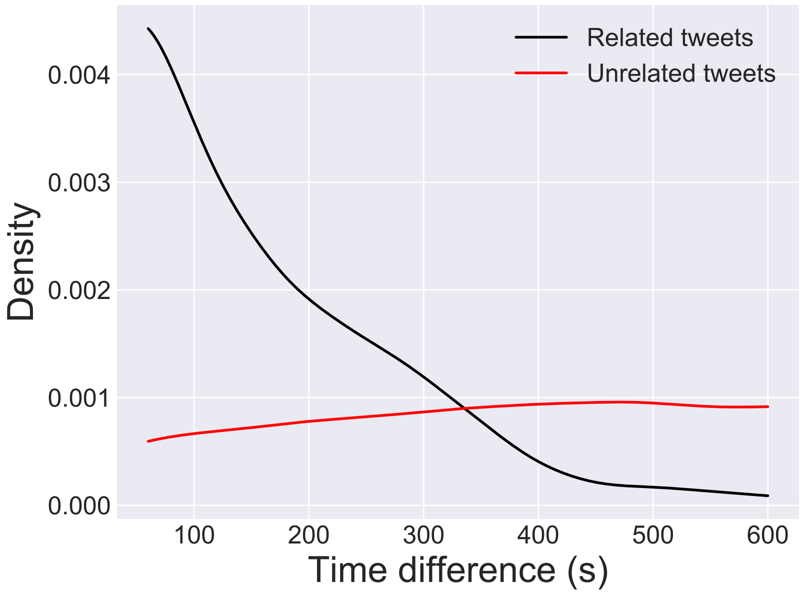


Figure 6.4: WBBL dataset *Z*2: Density of time differences between pairs of related and unrelated tweets. The density of time differences between pairs of unrelated tweets stays roughly constant with time difference, while the density of time differences between related tweets decays.

game with events only occurring at certain times, whereas AFL is fundamentally *continuous* with events happening at any time.

We have demonstrated that the probability of tweets being related decays exponentially with the time interval between them. This exponential decay is consistent with previous models which assumed that human interest in topics decays exponentially over time [90]. This motivates a key new step in our clustering algorithm, outlined in Section 6.3.2. To incorporate temporal information, we multiply a textual affinity score between two tweets by a function exponentially decaying with the time interval.

6.3.2 Social Media Event Response Clustering (SMERC)

Our algorithm SMERC creates clusters of tweets in response to an event. The algorithm is an 8-stage process, where most stages are standard techniques for text processing and clustering. Our algorithm incorporates the novel use of temporal information in the form of an exponential decay function multiplying the textual affinity between tweets. This prevents tweets that are a long time apart from being assigned to the same cluster. Many previous clustering methods use time as a linear

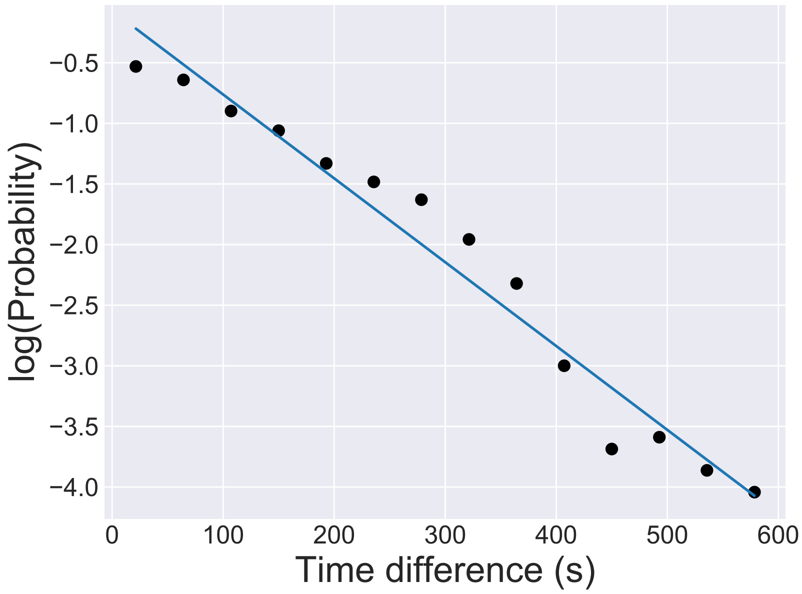


Figure 6.5: WBBL dataset Z2: Log-linear plot of the probability of tweets being related, given the time separation. The straight line indicates exponential decay.

variable which is incorporated into a feature vector [59], but this is a less meaningful incorporation of time than our multiplication of the textual affinity measure by an exponential function decaying with the time difference. It is entirely unsupervised and generates clusters of tweets related to events without requiring human assistance. The clustering algorithm, including the text pre-processing steps, is summarised as follows and explained in detail below:

1. Remove stop words from tweets.

2. Convert words to their associated stems.

3. Create a bag-of-words vector representing each tweet.

4. Use TF-IDF to more heavily weight words that occur less frequently.

5. Use cosine similarity to determine the textual similarity between TF-IDF vectors representing tweets.

6. Using the temporal information in each tweet, multiply each textual similarity by $e^{-\Delta t_{ij}/T_p}$, an exponential decay function dependent on the time interval Δt_{ij} between the ith and jth tweets, where T_p is a constant.

7. Use affinity propagation to determine clusters.

8. Filter clusters to ensure sufficiently high average affinity between elements.

We begin with m tweets, $\mathbf{w} = \{w_i\}$ for $i = 1, \ldots, m$. Steps 1 and 2, the removal of stop words and taking the stem of each word, are standard techniques used in natural language processing. The pre-defined list of stop words in the Python Natural Language Toolkit (NLTK) [96] is used to remove words that are purely for language structure, such as *the* or *at*. We perform stemming using the *PorterStemmer* package from the Python NLTK, to remove word suffixes such as *ed* or *ing*. It reduces the computational complexity by reducing the size of eventual word vectors, and allows easier identification of repeated word meanings. After the removal of stop words and stemming, we are left with a "cleaned" set of tweets, $\mathbf{w}' = \{w'_i\}$, $i = 1, \ldots, m$.

In Step 3, we use bag-of-words [133] to vectorise the tweets $\mathbf{w}'$ into a set of numerical vectors $\mathbf{x} = \{x_i\}$ for $i = 1, \ldots, m$, where the elements of $x_i \in \mathbb{R}^n$ represent the count f_{li} for $l = 1, \ldots, n$, of each word stem from a vocabulary V of size n that is used within tweet w'_i. We remark that this particular embedding of tweet w'_i is not essential; the techniques developed here can also be used with other vectorisation methods such as word2vec [109]. We use this standard method of vectorisation for its simplicity and performance; experimentation showed that bag-of-words resulted in superior performance to word2vec when analysing sporting events. This is likely to be due to the context-specific word definitions used in sport-related tweets. For example, we used the standard Python implementation of word2vec[2], trained on Wikipedia data, and words such as *tackle* likely have different meanings in sport than the Wikipedia contexts. Also, word2vec is not able to provide a meaningful vector representation of proper nouns, which are important for our context. If a unique dictionary was created for each sport, or if a sufficiently-large training corpus specific to the purpose could be created (e.g. comprising sport-related tweets), it is possible that word2vec would have superior performance. This presents an interesting avenue for future research.

In Step 4, we use Term Frequency - Inverse Document Frequency (TF-IDF), as explained in Section 2.4.5, to more heavily weight words that discriminate between documents. This increases the likelihood that tweets with rarely used but informative words will be clustered together, a desired property for our clustering algorithm.

As discussed in Section 2.4.3, cosine similarity emphasises the *closeness* between texts, which for our purposes is the number of words in common, weighted by their fre-

[2] https://github.com/danielfrg/word2vec

quency in the entire document [174]. For this reason, cosine similarity is our preferred similarity measure, and our design of the clustering algorithm is tailored to ensure that it can be used. In Step 5, we use cosine similarity, explained in Section 2.4.3, to measure the similarity between any pair of tweet vectors x_i and x_j. We then have an $m \times m$ symmetric matrix A of cosine similarities, where $A_{ij} = A_{ji}$ is the cosine similarity between the ith and jth tweets.

Section 6.3.1 showed that for two tweets separated by time interval Δt_{ij}, the probability that the tweets are related is proportional to $e^{-\Delta t_{ij}/T_p}$ where T_p is a constant. Consequently, in Step 6, for each pair of tweets w_i and w_j that occurred at times t_i and t_j respectively, we multiply the cosine similarity score A_{ij} by $e^{-\Delta t_{ij}/T_p}$ where $\Delta t_{ij} = |t_i - t_j|$. The value of T_p is chosen to be the time where the probability of tweets being related has fallen by the factor $1/e$, determined from our analysis in Section 6.3.1. The value of T_p is very important to the effectiveness of the method, as it affects the impact of time separation on clustering. We subsequently have a matrix of exponential scaling factors E, where $E_{ij} = e^{-\Delta t_{ij}/T_p}$. We then take the Hadamard (or element-wise) product of the matrix of cosine similarities A with the exponentially decaying scaling factor matrix E, giving the resulting affinity matrix $C = A \circ E$.

In Step 7 we use the *scikit-learn* [122] implementation of affinity propagation for the clustering, selecting a pre-defined affinity matrix. (Previous literature about clustering algorithms is discussed in Section 2.4.4.) Affinity propagation has order of complexity $O(km^2)$ where m is the number of tweets, and k is the number of iterations. We choose affinity propagation as it is one of few clustering algorithms that meets the following requirements. First, we do not know a priori the number of relevant clusters. This rules out clustering algorithms such as k-means or spectral clustering where the number of clusters must be specified beforehand. Second, due to our use of the exponential decay function, we require a clustering algorithm that accepts a similarity matrix. Third, our data tends to have a large number of clusters, representing the number of events and diverse topics of tweets, so we need a clustering algorithm capable of breaking the tweet set into a large number of clusters. Another clustering algorithm that met our requirements is DBSCAN [48], but we found that affinity propagation was more effective in practice.

The processes of calculating the cosine similarity between each pair of tweets, multiplying by an exponential decay function, and undertaking the clustering by affinity propagation are all operations with time complexity $O(m^2)$, where m is the number of input tweets. These potentially slow down the processing for large tweet collections. However, as the calculations for each pair of tweets are independent, we can speed up this process by parallelising the algorithm and evaluating the exponential function on GPUs instead of CPUs. If the number of tweets in the dataset is still too large for efficient processing, we can split the input dataset into smaller windows of time. As our tweet clustering algorithm avoids putting tweets with high temporal distance in the same cluster, such an action will have an effect only on the output around the time-window's boundaries. Slightly overlapping the time windows is a practical solution that could be used for an industrial implementation of this technique over an extended time period.

After clustering, we measure the internal average affinity between elements. If this is above a threshold value δ, we keep the cluster (Step 8 of our method). Testing for this threshold is necessary for the purpose of event prediction, as the affinity propagation algorithm assigns all tweets to a cluster. Consequently, there will be a number of clusters containing tweets that are unrelated to specific events. While these clusters are informative about the background topics of conversation taking place during a sporting contest, for the purpose of event prediction they are discarded. Through experimentation, we find that an average cluster affinity threshold of $\delta = 0.1$ is a good compromise between removing clusters of insufficiently related tweets, while retaining clusters about events of interest. We also automatically filter out clusters with less than five elements, due to the low overall attention to such events.

6.3.3 Experiments and results

We experiment with our algorithm on collected datasets $Z1$, $Z2$ and $Z3$. The performance of the exponential cutoff component of the algorithm is measured by determining the percentage of tweets that are correctly or incorrectly removed from clusters. We also give some examples of output clusters to illustrate performance. We note there is not necessarily one-to-one correspondence between clusters and events: it is possible that multiple clusters are associated with the same event. While we focus on these three datasets for illustrating SMERC, we remark that performance on other datasets was similar.

6.3.3.1 Performance metrics

As seen in Table 6.1, our temporal adjustment does not necessarily increase or decrease the number of clusters generated by affinity propagation.

Dataset	# Clusters without temporal adjustment	# Clusters with temporal adjustment
Z1: AFL first prelim final	203	229
Z2: WBBL first weekend	140	133
Z3: Heat vs Stars BBL game	241	281

Table 6.1: Dataset clustering summary. Temporal adjustment does not systematically increase or decrease the number of clusters.

For each event cluster detected by both methods, with and without the temporal adjustment, we calculate intra-cluster precision: the percentage of tweets within a cluster that are correctly classified. Table 6.2 shows an increase in precision after our temporal adjustment. We move from a precision around 50% to a precision of around 90%, a very large improvement. This occurs because tweets that are temporally distant to an event tend to be removed from the associated cluster.

Dataset	Average precision before temporal adjustment (%)	Average precision after temporal adjustment (%)
Z1: AFL first prelim final	45.5	88.7
Z2: WBBL first weekend	56.8	97.1
Z3: Heat vs Stars BBL game	56.7	92.1

Table 6.2: Improvement of intra-cluster precision from around 50% to around 90% with temporal adjustment.

6.3.3.2 Example clusters

We give two examples demonstrating the effectiveness of SMERC on our collected Twitter datasets. We first examine the output of our method on dataset Z1, tweets about the AFL match between Adelaide Crows and Geelong Cats on 22 September 2017. We ran SMERC both with and without temporal adjustment. Table 6.3 shows the tweets in a particular cluster that were included with the temporal adjustment, while Table 6.4 shows additional tweets included in the cluster when operated without temporal adjustment. The clustering method correctly clustered a series of tweets following a goal at around 09:54 UTC by Eddie Betts. However, without the temporal adjustment, tweets about an additional goal at around 10:14 UTC were also clustered together. It is clear by watching the games and looking at the tweet times that these were distinct events and hence that our method correctly separates the tweets into appropriate groupings.

Tweet body	**Time (Sep 22, 2017) UTC**
Eddie!!!! #AFLCrowsCats	09:54:11
Eddie, you beauty!!! #AFLCrowsCats	09:54:17
#AFLCrowsCats sorry cats fans... I LOVE EDDIE!	09:54:20
Fair shark by Eddie. #AFLCrowsCats	09:54:32
#AFLCrowsCats Eddie's Best	09:54:51
Eddie's goal from a stoppage was a coach killer! Can't let him move like that in F50! #AFLCrowsCats	09:56:02

Table 6.3: AFL dataset: Example of a particular cluster of tweets using our method, with the temporal adjustment. All tweets refer to the same event, a goal by Eddie Betts.

Tweet body	**Time (Sep 22, 2017) UTC**
Eddie! What a goal! 37-8 #AFLCrowsCats	10:14:34
EDDIE.************.BETTS.#AFLCrowsCats	10:14:36
It's Eddie's world and we're just living in it #AFLCrowsCats	10:14:37
Eddie! You are the king of Adelaide! #AFLCrowsCats	10:14:38
Uncle Eddie, ******* hell. #AFLCrowsCats	10:14:39
Eddie. What more can you say? #AFLCrowsCats	10:14:45
That was delicious, Eddie! #AFLCrowsCats	10:14:45
Eddie. Betts. He is that good! #AFLCrowsCats	10:14:47
Eddie. #AFLCrowsCats #WeFlyAsOne	10:14:48
#AFLCrowsCats Eddie's on fire	10:14:54
Beautiful Eddie. Beautiful. @Adelaide_FC #AFLCrowsCats #AFLFinals	10:15:07

Table 6.4: AFL dataset: Example of additional tweets included in the cluster when operated without temporal adjustment. These tweets all refer to a later goal in the match, and were correctly split into a different cluster when temporal adjustment was included.

A second set of example clusters is given in Tables 6.5 and 6.6, for dataset Z2, a WBBL match. Table 6.5 shows the tweets that were included with the temporal adjustment, while Table 6.6 shows additional tweets included in the cluster when operated without temporal adjustment. The event of interest that our algorithm is clustering is the milestone score of *fifty* by cricketer Rachael Haynes.

Tweet body	**Time (Dec 9, 2017) UTC**
50 for Haynes and it comes from just 37 balls 4/104 #ThunderNation #WBBL03	03:10:50
What a knock from Haynes, she reaches 50 off just 37 balls for @ThunderWBBL #WBBL03	03:11:34
50 for @RachaelHaynes in just 37 balls. The first fifty of #WBBL03	03:12:02
#WBBL03 First FIFTY of the season. The Aussie skipper @RachaelHaynes brings it up in 37 balls.	03:13:55
#WBBL03 First FIFTY of the season. @RachaelHaynes brings it up in 37 balls.	03:15:32

Table 6.5: WBBL dataset: Example of a particular cluster of tweets using our method, with the temporal adjustment. All tweets refer to a milestone score by cricketer Rachael Haynes.

Tweet body	**Time (Dec 9, 2017) UTC**
That's what happens when my favourite leftie @RachaelHaynes Middles the ball. First six in #WBBL03	02:57:01
A maiden @WBBL fifty for @Jess_cameron27. This is also the second fifty of this season! #WBBL03	04:50:33
50 in just 22 balls. That's @ashleighgardne2 for you. #WBBL03	07:12:27

Table 6.6: WBBL dataset: Example of additional tweets included in the cluster when operated without temporal adjustment. None of these tweets refer to the milestone score by Rachael Haynes referenced in Table 6.5, so are correctly removed from the cluster.

6.4 Event time estimation

We now develop a method to estimate the times of events from the Twitter response. As with SMERC earlier in this chapter, we develop and evaluate our event time estimation method using data from sporting event datasets.

6.4.1 Data collection and manual labelling

Television coverage of sporting events regularly has a several-second delay, even if advertised as *live*. Also, there can be multiple times for broadcasts of the same event. For example, AFL events in Australia are often broadcast near live in some states, but at a 30-minute delay in other states. These variations in broadcast timings cause a corresponding delay in the social media response. Many users believe they are watching a live event and tweet accordingly. As the timing of events is critical to this chapter, wherever possible we use data from events that are broadcast as close to live as possible. For training data, we select events followed by at least ten tweets that by our judgement, from watching the games and reading the Twitter feed, were directly in response to the event. For example, after an AFL goal many people immediately tweet about both the goal and the player who kicked the goal. We omit any event with a large discrepancy in response times (e.g. a ten-minute interval between response tweets), usually caused by a television broadcast delay.

We create labelled data by recording the times of events of interest directly from the broadcast itself. Many Twitter users respond to television broadcasts of the event, so we take the broadcast time of the event as the reference point. We denote our collected datasets $\mathbf{Q} = \{Q_i\}$, consisting of $n = 22$ hand-labelled event datasets from the 2016/2017 AFL seasons. Let t_i be the time of the ith event, $i = 1, \ldots, n$, determined by watching the broadcasts and manually recording the times of key events. Let t_{ij} be the time of the jth response tweet to the ith event, determined by manually labeling tweets in response to key events. From the response times $\{t_{ij}\}$ for each event i, our goal is to estimate the time t_i of the event as accurately as possible.

6.4.2 Example tweets

In Table 6.7 we give details of the events used for our example plots. All events are from the AFL match on 24 March 2016, between the Richmond Tigers and the Carlton Blues. This was the first match of the 2016 AFL season, occurring on a Thursday evening at 6:50 pm ACDT (8:20am 24 March 2016 UTC).

To visualise the response distribution, we show in Figure 6.6 histograms of the times between event and response tweets for each event. The rate of tweets about each event generally starts at a low level, increases to a peak, and then decays over time. This shape is the characteristic of both the log-normal and Weibull distributions with suitable parameters. As our datasets are relatively small (less than sixty tweets in each case), there is a significant amount of noise in the dataset.

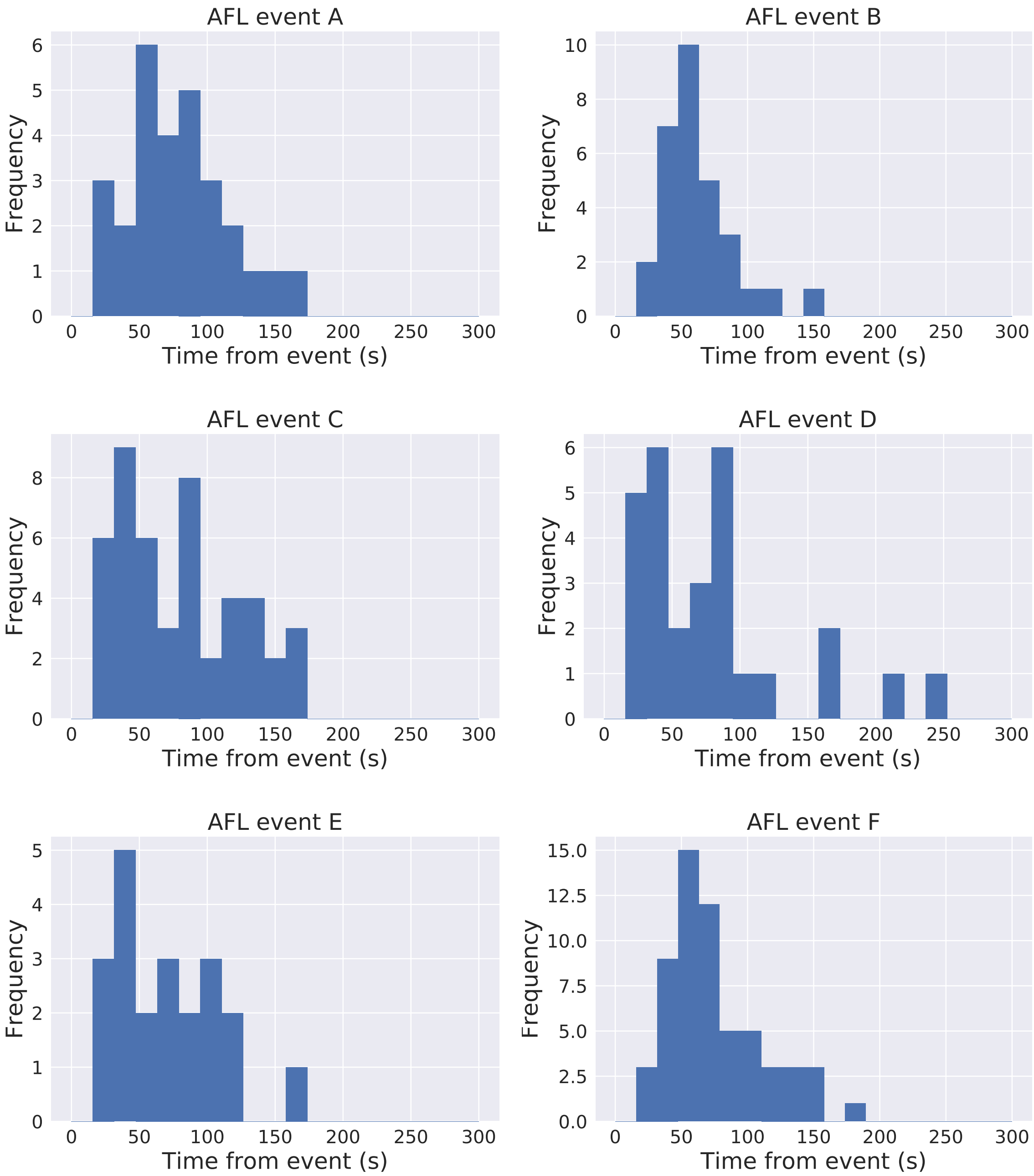


Figure 6.6: Times between event and response tweets about the sample AFL events. The peak is generally a short time (approximately 60 seconds) after the relevant event. This data has a high proportion of noise due to the relatively small number of tweets about each event. Note that the y-axis scales vary for each plot.

Label	Event description	Event time (UTC)	Number of tweets
AFL event A	First goal of the season to Jason Castagna (Richmond Tigers)	2016/3/24 08:22:50	28
AFL event B	Goal to Toby Nankervis (Richmond Tigers)	2016/3/24 08:25:43	30
AFL event C	Dustin Martin and Dan Butler combine for a goal (Richmond Tigers)	2016/3/24 08:40:38	47
AFL event D	Goal to Dennis Armfield, (Carlton Blues)	2016/3/24 08:46:27	28
AFL event E	Deliberate out of bounds against Bryce Gibbs (Carlton Blues)	2016/3/24 09:04:30	21
AFL event F	Goal to Dustin Martin with a torpedo kick (Richmond Tigers)	2016/3/24 09:32:56	59

Table 6.7: AFL events from Carlton vs Richmond AFL game for which associated tweet data was collected.

6.4.3 Event time response distribution

We fit a Weibull and log-normal distributions to our data to determine whether they are suitable for event time estimation. To visually demonstrate the curve fitting, we define the event time at $t = 0$.

We first fit an *offset* Weibull distribution to each set of data points. The shape of the Weibull distribution is similar to the observed shape of the Twitter response to events, and the mechanics of the process can be associated if we consider the time until sending a tweet analogous to the time until failure in a system. The calculation of the offset allows estimation of the event time. Compared to the regular Weibull distribution in Equation (2.7), the distribution we fit is offset by time t_0 from the origin. An offset Weibull distribution with shape parameter k and scale parameter λ has probability density function

$$f(t;\lambda,k) = \begin{cases} \frac{k}{\lambda}\left(\frac{t-t_0}{\lambda}\right)^{k-1} e^{-((t-t_0)/\lambda)^k} & t \geq t_0, \\ 0 & t < t_0. \end{cases} \tag{6.1}$$

We use maximum likelihood estimation to determine parameters t_0, k and λ, with t_0 used to estimate the event time. We measure how well the offset Weibull distribution fits the data using the KS statistic.

We also fit an *offset* log-normal distribution to our data. The theoretical foundation of the causes of log-normal distributions are discussed in Section 2.6.1. A log-normal process can be created by the multiplicative product of many independent variables, or the accumulation of many small percentage changes [110]. When creating a tweet about a given event, a user has many choices including content, length and whether or not to subsequently edit their tweet text. If these choices lead to an accumulation of many small percentage changes in the time taken to create the tweet, this would lead to a log-normal distribution. For a log-normally distributed random variable X, we have $X = e^{\mu+\sigma Z}$ where Z is a standard normal variable, μ and σ are location and scale parameters for the variable's natural logarithm respectively. A log-normal distribution with offset t_0 from the origin has probability density function

$$f(t;\mu,\sigma) = \frac{1}{(t-t_0)\sigma\sqrt{2\pi}} e^{-\frac{(\log(t-t_0)-\mu)^2}{2\sigma^2}}, \quad t \geq t_0. \tag{6.2}$$

We use maximum likelihood estimation to fit the curve and determine parameters t_0, σ and μ. Again, offset parameter t_0 will be used to estimate the event time. The scale parameter σ is higher for distributions with slower decay rate.

Figure 6.7 shows the cumulative distribution function for fitting both an offset log-normal distribution and an offset Weibull distribution to our sample dataset. The log-normal distribution also fits the data well with KS distances ranging from 0.0658 to 0.0956. These are slightly lower than for the Weibull distribution. However, for log-normal distributions, particularly those with low σ values, the CDF exhibits a long region of low event probability. Of our sample datasets, AFL event A is the most extreme example of this phenomenon, with a long tail ranging from -67.59 to 10. Clearly it is not possible for reactionary tweets to occur before the event, so this distribution is not realistic. Also, the flat end to the distribution causes the end point to heavily fluctuate based on small changes to the data, and this sensitivity limits this model's usefulness for event time estimation. There exist alterations that could potentially reduce the severity of this problem such as fixing the σ value, or taking the end point of the distribution when the CDF goes below a specified low value. However, we prefer to use a different distribution that does not suffer from this problem. Due to the x-axis intercept of the Weibull distribution CDF being sharper than the log-normal distribution, a Weibull distribution is more effective at estimating event times. We summarise parameters for the cumulative distribution fits to the datasets in Table 6.8.

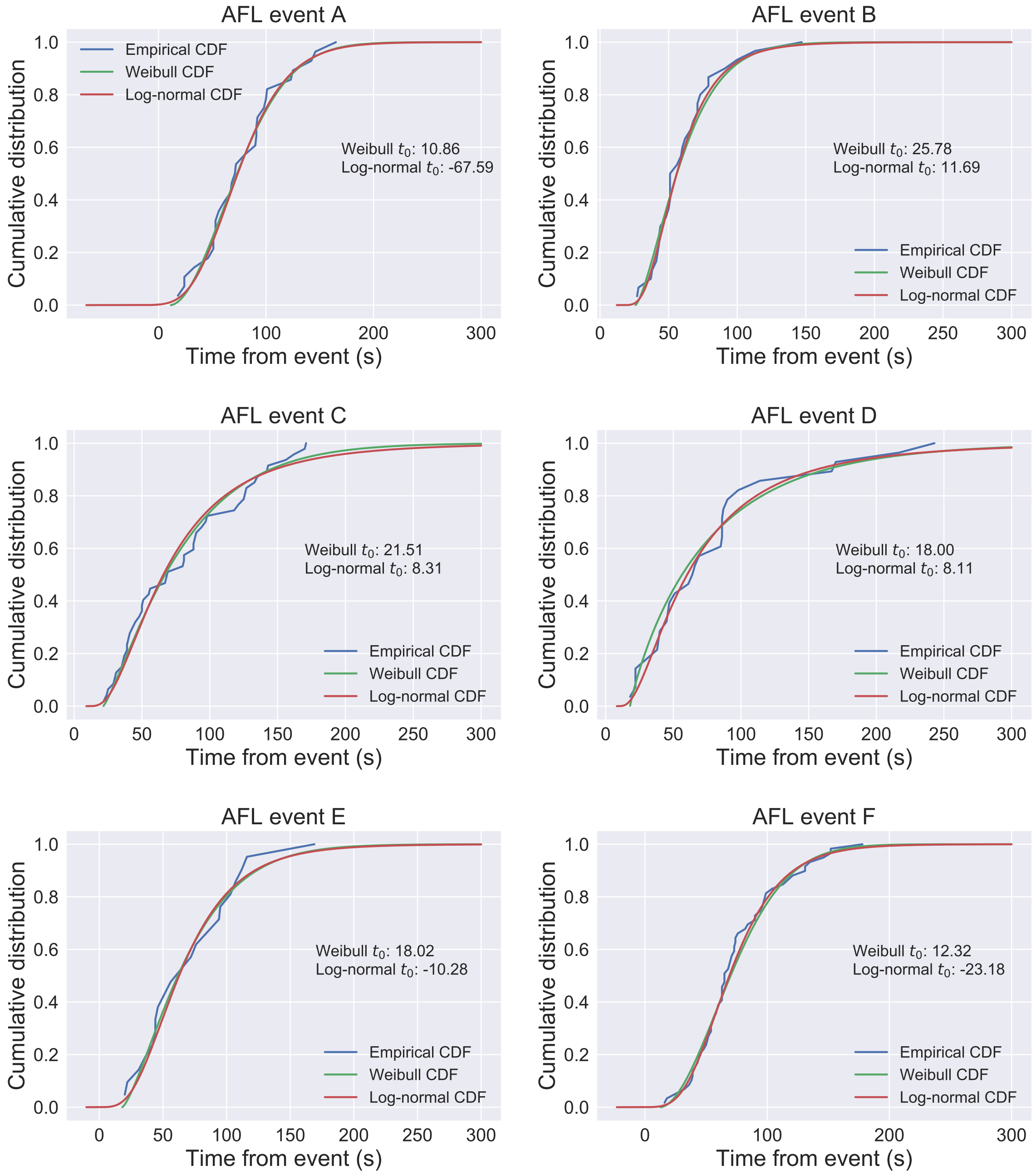


Figure 6.7: Weibull and log-normal fits for the cumulative distribution function of tweet times about sample AFL Events. Both distributions have a similar shape which provides a good fit to the data, with the primary difference being the longer left tail for the log-normal distribution. The x-intercept t_0 for the Weibull fit ranges between 10.86 and 25.78, a relatively small window. However, for the log-normal fit, the t_0 values range between -67.59 and 11.69, a very wide range indicating that this distribution is less well suited to our event time estimation approach. Parameters for each curve are given in Table 6.8.

	Weibull					**Log-normal**			
Label	t_0	λ	k	**KS**		t_0	μ	σ	**KS**
AFL event A	10.86	75.78	1.86	0.0730		-67.59	4.95	0.26	0.0658
AFL event B	25.78	39.07	1.43	0.0856		11.69	3.79	0.48	0.0956
AFL event C	21.51	61.66	1.21	0.0768		8.31	4.05	0.69	0.0936
AFL event D	18.00	57.62	0.90	0.1045		8.11	3.95	0.81	0.0909
AFL event E	18.02	57.24	1.36	0.0775		-10.28	4.29	0.46	0.0928
AFL event F	12.32	70.41	1.85	0.1010		-23.18	4.52	0.35	0.0846

Table 6.8: Parameters for Weibull and log-normal fits to the sample AFL dataset. Both distributions have similar KS distance values, ranging between 0.0658 and 0.1045 for all datasets. For all datasets except AFL event D, the shape parameter k for the Weibull distribution is in the expected range of between 1 and 2. The Weibull distribution fit has a much narrower range of t_0 values, all of which are positive, making it more suited for our event time estimation method.

Our Weibull distribution fits in Figure 6.7 had sharper k parameter ranging from 0.90 to 1.86. Figure 6.8 shows Weibull fits to our data with a fixed $k = 1.40$, chosen to reflect the approximate average observed shape of the empirical distributions. The performance of the event detection method is not sensitive to the choice of this parameter: experimentation showed that values between $k = 1.30$ and $k = 1.50$ gave a similar level of fit accuracy. As seen in the plots, for event D, this method does not fit the data as closely as the Weibull distribution with variable k. However, as we will show, fixing the value of k does a better job of estimating event times as we have less parameters in our equation, and therefore less tendency to overfit.

6.4.4 Estimating event times on a larger dataset

To estimate event times, we fit a distribution to the observed tweet time responses, then take a fixed time offset from the start point of this distribution. Our method consists of the following steps:

1. Fit selected distribution D to dataset $\{t_{ij}\}$ with maximum likelihood estimation.
2. Record initial time parameter t_{i0} from the fitted distribution.
3. Calculate the estimated event time as $\hat{t}_i = t_{i0} - t_p$.

In Step 1, we select distribution D as either offset Weibull, defined in Equation (6.1), or offset Weibull distribution with fixed shape parameter $k = 1.40$. After fitting the distribution, in Step 2 we extract the parameter of interest t_0. The shape and scale

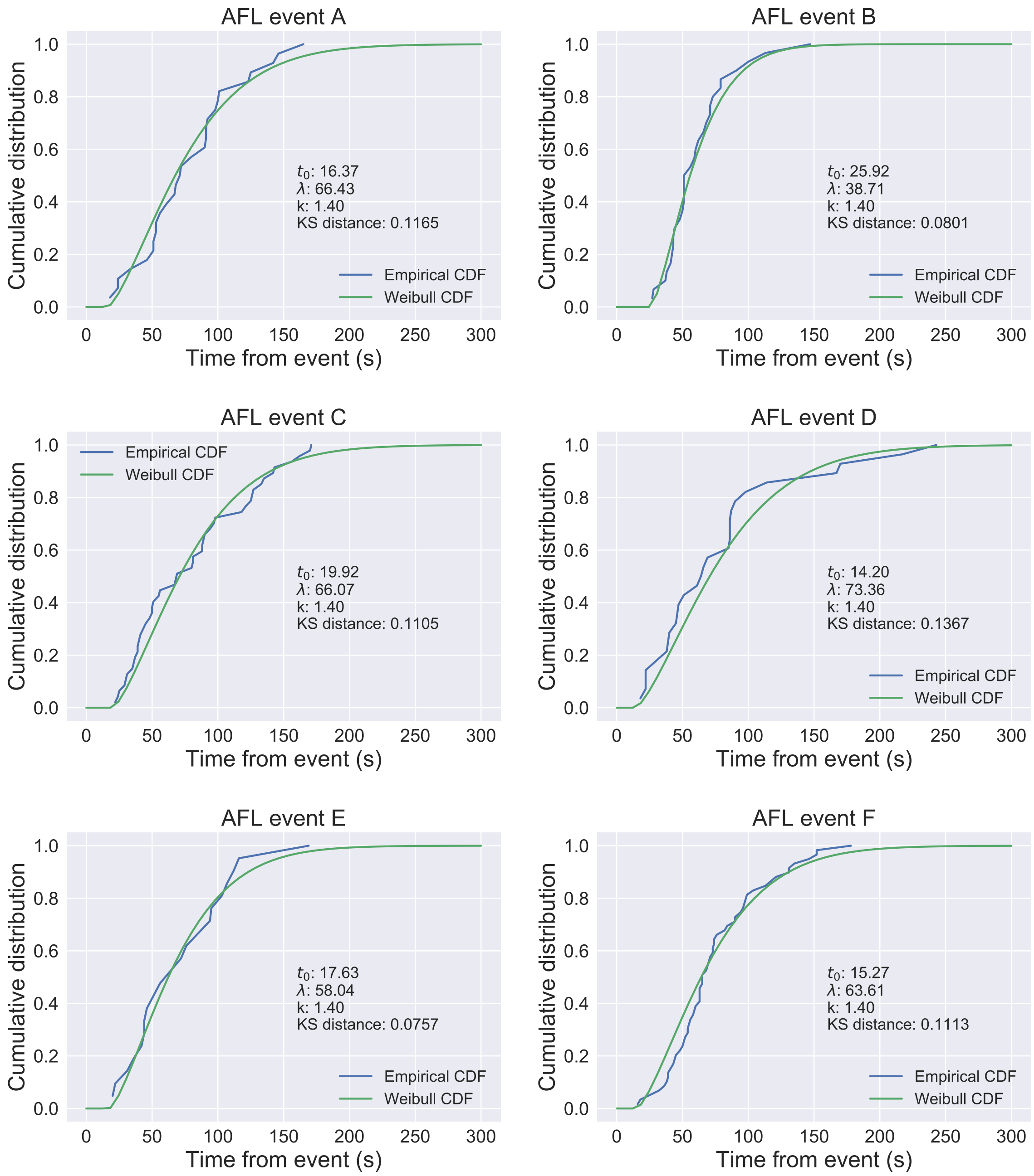


Figure 6.8: Weibull fit with fixed shape parameter $k = 1.40$ for times of tweets about the sample AFL Events. The Weibull fit with fixed k provides a slightly worse fit than with variable k, but is still reasonable, as can be seen by the KS distance. The t_0 values range between 14.20 and 25.92, the smallest window of all the methods that we have examined.

Method	**Event time prediction MAE (s)**	**KS-statistic**
Fixed time from first tweet	6.39	N/A
Weibull fit	4.82	0.0827
Weibull fit with fixed k	3.06	0.1051

Table 6.9: Results of performance evaluation between event time estimation methods. (MAE = Mean Absolute Error.)

parameters are not used for the subsequent event time estimation. Finally, in Step 3 we subtract a fixed time $t_p = 18$ seconds from the fitted start of the distribution t_{i0}. Physically, t_p represents the short period of time after an event where it is physically infeasible for someone to write and send a tweet, a parameter selected by analysing tweet datasets. Visual representations of our event time estimations are given in Figure 6.9.

We test our event-time estimation method both with and without fixing the value of k on dataset $\mathbf{Q} = \{Q_i\}$; the results are given in Table 6.9. We also test against a baseline method where we subtract a fixed time of $t_p = 22$ seconds from the first observed tweet, selected to minimise the error of the prediction. The best mean absolute error occurs for the Weibull distribution with fixed $k = 1.40$ method.

6.4.5 Bootstrapping to estimate error in measurement

We bootstrap our input time data in order to estimate the uncertainty in our event time estimation. (An overview of bootstrapping and related literature is given in Section 2.6.11.) We use bootstrapping as our datasets are often too small to use other statistical methods. For each dataset i, we create $m = 1000$ resampled datasets $\{t_{ij}\}$ from the original sample with replacement. Each resampled dataset is the same size as the original, and is used to conduct the event time estimation. We then obtain a set of time estimations $\tilde{t}_i^m$ for the mth resample of each dataset i. We calculate the sample standard error of these times to provide a confidence interval for all data:

$$\mathrm{CI} = \bar{x} \pm t_{\alpha/2}\, \sigma_{\hat{\theta}}, \tag{6.3}$$

where $\bar{x}$ is the estimated mean value, $\sigma_{\hat{\theta}}$ is the sample standard error calculated through bootstrapping and $t_{\alpha/2}$ is the critical value of the test statistic. The standard error values produced from bootstrapping are shown in Table 6.10. The event with the highest number of response tweets, event C, has the lowest standard error.

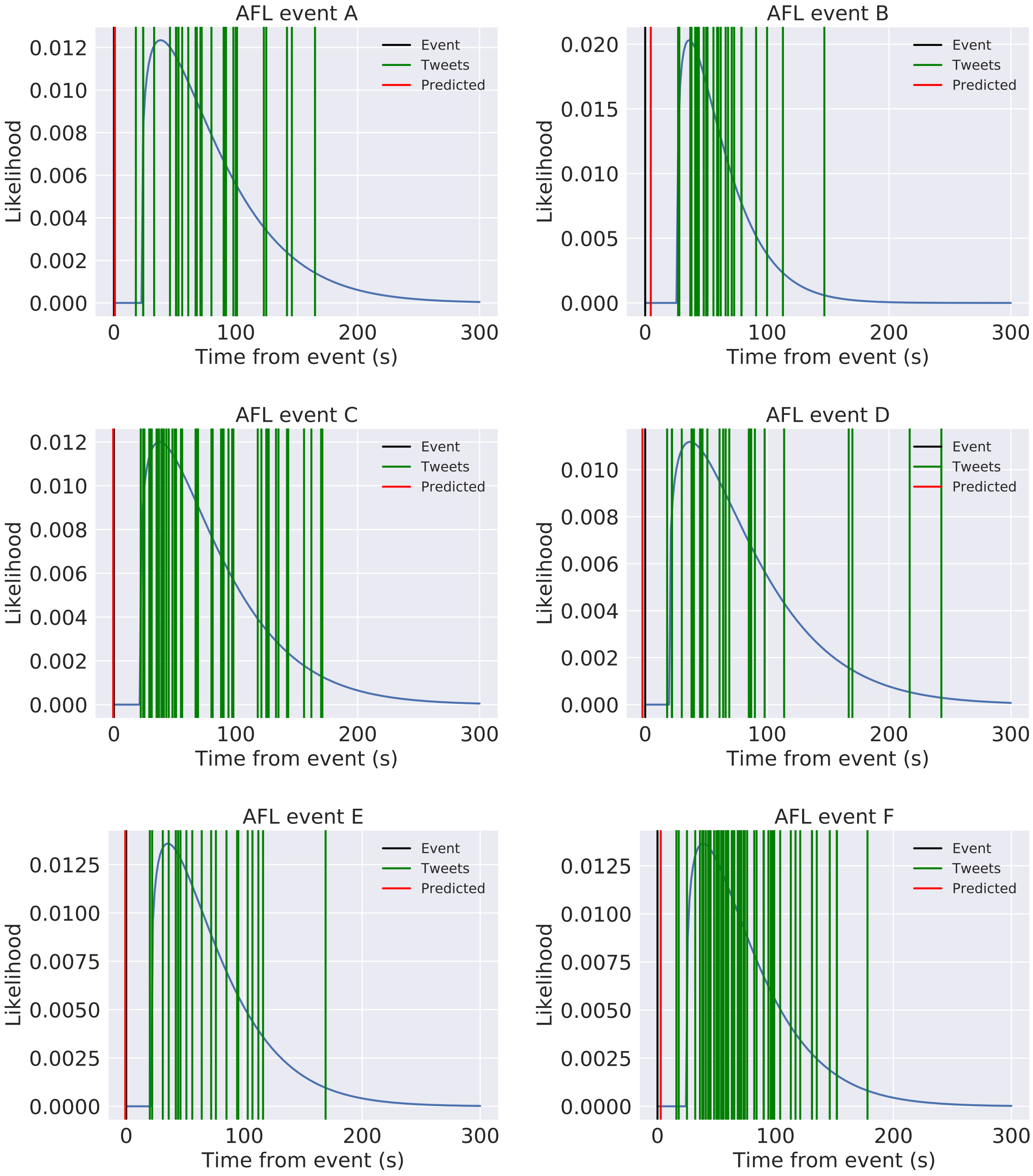


Figure 6.9: Event time estimation from tweets about selected AFL events using Weibull distribution with fixed k value. Vertical green lines represent the times of tweets after the causing event. The blue line is the probability density function of the fitted offset Weibull distribution. The black vertical line represents the time of the actual event, and the red vertical line is the estimated event time. All events were on March 24, 2016.

	Actual (UTC)	Predicted (UTC)	Error (s)	SE (s)
AFL event A	08:22:50	08:22:48.37	-1.63	5.78
AFL event B	08:25:43	08:25:50.92	+7.92	3.52
AFL event C	08:40:38	08:40:39.92	+1.92	1.74
AFL event D	08:46:27	08:46:23.20	-3.80	2.72
AFL event E	10:04:30	10:04:29.63	-0.37	5.92
AFL event F	10:32:56	10:32:53.27	-2.73	4.36

Table 6.10: Bootstrapping output from sample tweets. All events were on March 24, 2016. (SE = Standard Error.)

This process allows us to provide a confidence interval for the estimated event time. For example, we can give a 90% confidence interval for the time of AFL event C as

$$\begin{aligned}\text{CI} &= \text{08:40:39.92} \pm 1.645 \times 1.74\\ &= \text{08:40:39.92} \pm 2.86 \text{ seconds,}\end{aligned}$$

which contains the actual event time. Five of our six example events have a 90% confidence interval which contains the actual event time, all except *AFL event B*.

6.5 Discussion and conclusions

In this chapter we developed a new algorithm, *Social Media Event Response Clustering*, for clustering tweets using both textual and temporal information. This exponentially-decaying model was informed by a detailed analysis of a number of Twitter datasets collected around sporting events. This is critical for improving the quality of clusters generated by the method. In addition, we developed a technique to estimate the times of events from the social media response.

Using data from sporting events has the advantage of known event times and hashtags. However, it is a specific type of data to study and has potential limitations. For example, some incidents in sporting events will only be of interest for an extremely short period of time. A questionable umpiring decision may cause immediate outrage from supporters, but may then be forgotten minutes afterwards. Additional work would be needed to determine the effectiveness of our clustering method for other fields of study.

Our clustering and time estimation approaches could potentially by improved by automated removal of noise on the input Twitter data, as done for other purposes [94]. Spam and advertisements tend to repeat identical or very similar tweets, which have high pairwise text similarity. Automated removal of these tweets before clustering will improve the amount of information content in the output clusters.

In addition to sport, SMERC could be applied to other fields such as social unrest or natural disasters, where people respond to real world events, with a much slower exponential decay. Provided that the tweets contained sufficient regional information, it could also be used to detect contagion events, where many users in a region tweet about outbreaks of illnesses within a short period of time. Also, in addition to Twitter data, this work could be applied to data from other social media platforms such as Facebook or Youtube. The difficulty of event detection varies depending on the topic of the event. Detecting rare events with known keywords such as *earthquakes* or *goals* in soccer, is much easier than detecting less well defined or frequently occurring events. After an earthquake, many people will tweet the word "earthquake", and rarely otherwise. Consequently, the choice of dataset will affect the measured performance of the algorithms. Future work could involve testing SMERC on such data streams as they become available. In the interests of reproducibility of work and having common tweet sets for testing, we store our datasets on Github and make them publicly available[3].

For event time estimation, of the methods we tested, we found the Weibull distribution with fixed shape parameter k was most effective at estimating events times from times of reactionary tweets. This provides a useful addition to microblog summarisation methods. A series of tweets can be automatically separated into clusters and given an associated event time. Such a technique would be useful for many entities who would benefit from automated summaries of events, such as news agencies or governments.

Occasionally we observed a Twitter response that was faster than would be physically possible if the author of the tweet began their response after the event had occurred. We attribute these cases to when a user pre-prepares their tweet and then submits it once the event of interest occurs. For example, in AFL when a player takes a mark and subsequently has a shot on goal, there is potentially a 30-second interval between the mark and the kick. In this time a user could prepare a tweet to celebrate the goal, and then submit the tweet once the goal occurs. (For readers not familiar with AFL

[3] https://github.com/pete1729/phd-thesis

football, a corresponding example in basketball would be a free throw.)

Both the Weibull and log-normal distribution have an appropriate shape to model Twitter response to events. The Weibull was more suited to estimating times of events than the log-normal distribution, and can be linked with the underlying processes that lead to the time of Twitter response. A possible explanation of the Weibull distribution would be to consider the time until sending a tweet analogous to the time until failure in a system. If the marginal rate of sending a tweet increases over time (corresponding to the failure rate increasing over time, and shape parameter $k > 1$), then the time it takes to send tweets would have a Weibull distribution with our observed shape.

From a theoretical perspective, our work improves the understanding of the distribution of the decay in human interest, reflected in online social media data streams around events. Practically, it provides an effective method to cluster tweets for the purpose of event detection and then estimate the times of events. We evaluated this both quantitatively through the calculation of the standard evaluation metric precision and the mean absolute error of the estimated event time, and also qualitatively through inspection of the actual tweets clustered together by our method. We believe our method could be deployed by governments or other organisations to conduct social sensing using microblogs.

Chapter 7

Discussion and Conclusions

7.1 Temporal relationship between social media, events and news

A key goal in this thesis was to improve understanding of the temporal relationship between social media, events and news. Figure 7.1 provides a visual outline of the temporal distribution and causality findings we have discovered. In Chapter 3 we found that news and tweet volumes are strongly correlated, and that Twitter activity Granger-causes news activity. In Chapter 4 we showed that the rate of retweets is well-modelled by a power law with exponential cutoff. We also showed that for a given tweet about a topic, the likelihood that another tweet is about the same topic decreases exponentially with the time between the tweets. In Chapter 6 we also showed that the time taken between tweets and events has a shape resembling a Weibull distribution.

Due to scope limitations, we analysed only selected datasets in a small range of topics, predominantly politics and sport. Consequently, we do not claim that our temporal relationship findings necessary hold for all datasets across all topics. We also note that no theoretical distribution will ever be a perfect model of the real world.

7.2 Overcoming challenges in social media analysis

Section 1.4.1 outlined many of the challenges facing researchers while conducting social media analysis. Our extensive analysis on temporal relationships conducted

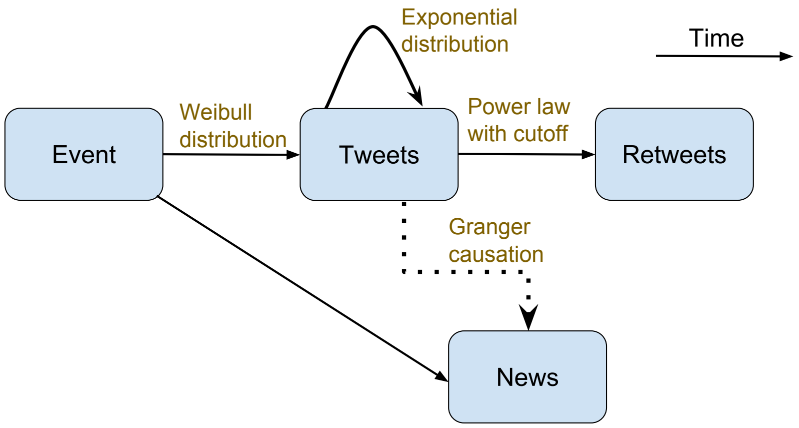


Figure 7.1: The temporal relationship between events, news, tweets and retweets was analysed in this thesis and the associated distribution or causality findings are indicated.

for this thesis gives us a perspective on the effectiveness of techniques to overcome such challenges.

In temporal social media analysis, we are most interested in understanding human behaviour, so bot behaviour can adversely affect our analysis results. As would occur from any social media platform, the data we collected contained tweets from bots. By observing retweets only after a set initial time in Chapter 4, we automatically filtered out instantaneous bot retweets. Also, we filtered out bot tweets in Chapter 6 manually through tweet text inspection, which is an effective but time-consuming solution.

As most of this thesis analysed temporal response instead of tweet contents, the difficulty of text processing in short 140/280 character messages did not greatly affect results in the majority of chapters. For clustering Twitter messages in Chapter 6, we found bag-of-words combined with TF-IDF to be a more effective solution than more modern techniques such as Tweet2Vec [41]. This is particularly true when analysing sports data, which has a unique language not well captured by algorithms trained on Wikipedia data.

One of the key practical challenges in microblog summarisation or event detection is the time-consuming task of manually labelling data for testing. Having publicly available test sets and evaluation methods would potentially advance the field considerably, as it would save time for researchers, allowing them to focus on algorithm development. We make all of our datasets available on Github in the hope that they will be of benefit to other researchers.

To adjust for diurnal cycles, another key challenge facing researchers but often ignored, we found our stochastic normalising approach in Chapter 4 to be an effective solution. The key benefit compared to other methods is that our stochastic approach outputs a discrete dataset, allowing follow-up statistical analysis. Also, we found that using our neural network in Chapter 3 performed well at modelling the diurnal cycle, better than linear regression which is not able to learn the cyclical shape.

7.3 Possible future directions

Our work presents many possibilities for future research. Many of our datasets have been from politics and sport, so a natural continuation would be to see if our results hold for other popular social media discussion topics such film or music. Due to the bursty and often unpredictable nature of Twitter discussions on these topics, such work may require new techniques to collect data, particularly if only using the public Twitter APIs as we have done.

Our work on news prediction in Chapter 3 could be extended by collecting more training and testing data over a longer time frame. Neural networks have higher prediction accuracy when large quantities of training data are available. Other features could also be incorporated into collected datasets, such as the changing number of followers for candidates or number of days until the election for political analysis.

For our work on the distribution of retweets, a natural continuation would be to determine whether our findings apply to users who are not in the top 100 based on worldwide popularity. Due to the sparser retweet sets of these less popular users, this would pose distribution fitting challenges and may require combining retweet datasets from different seed tweets. This would require making an assumption of homogeneity amongst users, which is unlikely to be well-motivated. Our method to estimate the size of a retweet cascade could be extended by combining it with other methods that include the rate of retweets as a component. Generally, authors estimate this as another function such as a power law [180], but changing this to a power law with exponential cutoff could potentially improve results.

Our event clustering and time estimation method SMERC could be extended by finding ways to reduce the time complexity of the algorithm. It could also be adapted into a system to detect events in real time, rather than using post processing. Also, creating a dedicated dictionary for the topics that we are analysing would potentially allow the use of more modern tweet processing methods such as Tweet2Vec [41].

Furthermore, the models and results which we developed could be tested by examining human behaviour. For example, we could record the actual orderings of human activity, including checking social media, to test whether this process is accurately modeled by a priority based queue. This is outside the scope of this thesis, but could be targeted for combined future research with social scientists.

7.4 Contribution to knowledge and development of new techniques

From a theoretical perspective, this thesis makes significant contributions to understanding the temporal social media response to events and the mechanics of digital information propagation. We made advances towards understanding the relationship between Twitter and the news, showing that tweet and news counts are correlated, that the Twitter reaction to events tends to occur before the news reaction, and that information contained within Twitter volumes can be used to improve prediction of news volumes. We analysed retweet temporal activity more deeply than previous authors, showing that a power law with exponential cutoff provides a better fit to retweet rates than the commonly used power law. An explanation for this relationship was developed using human behaviour theory, linking social science knowledge with social media temporal dynamics. We demonstrated that the likelihood of two tweets being related decays exponentially with the time gap between them. Finally, we showed that the rate of Twitter response to an event can be well-modelled by a Weibull distribution.

This thesis has also made advances in technical analysis of temporal dynamics of social media data. Our stochastic diurnal adjustment method allows automated event detection from social media data independent of the hour of the day. Our development of understanding the distribution of retweets naturally led to a method for prediction of retweet cascade size, which works particularly well for selected seed tweets from news sources. We developed techniques to automatically cluster social media data and estimate the time of events causing the social media response. This allows automated social sensing from microblogs, a useful tool in understanding past and current human attention.

As the prevalence of social media continues to grow, being able to understand and accurately model the underlying dynamics will provide insights into the changing nature of our digitial society. Our work provides a significant contribution towards this goal.

Appendix A

Supplementary Material for Chapter 3

This appendix provides additional content for Chapter 3, *The Temporal Relationship between Tweets and News*.

We examine the tweet versus news relationship for Australian politicians Malcolm Turnbull and Bill Shorten. Figure A.1 shows the public tweet, candidate tweets and news stories about Malcolm Turnbull from 6 June to 3 July, 2016. A clear diurnal cycle can be seen with low points around 18:00 UTC, corresponding to 3am on the East Coast of Australia. In general, the news counts and public tweet counts collected for Australian politicans are slightly lower than for US politicans, leading to a noisier shape. As with US politicians, higher public tweet counts occur near news stories and candidate tweets. Repeating these plots for politician Bill Shorten, provided in Figure A.2, gives a similar result with higher public tweet counts tending to occur closer to news stories or candidate tweets.

Figures A.3 and A.4 show the diurnal cycles of news and tweet counts for Malcolm Turnbull and Bill Shorten, respectively. The diurnal cycles show some similar features, especially the lack of activity overnight, but are less similar than we observed for US politicians. During the afternoon / evening in Australia (5 am to 12 pm UTC) the public tweet / news ratio is higher than at other times of the day. This is likely to be due to Australian journalists tending to produce less news stories in the evening, compared to news sources in the United States producing stories at all hours of the day.

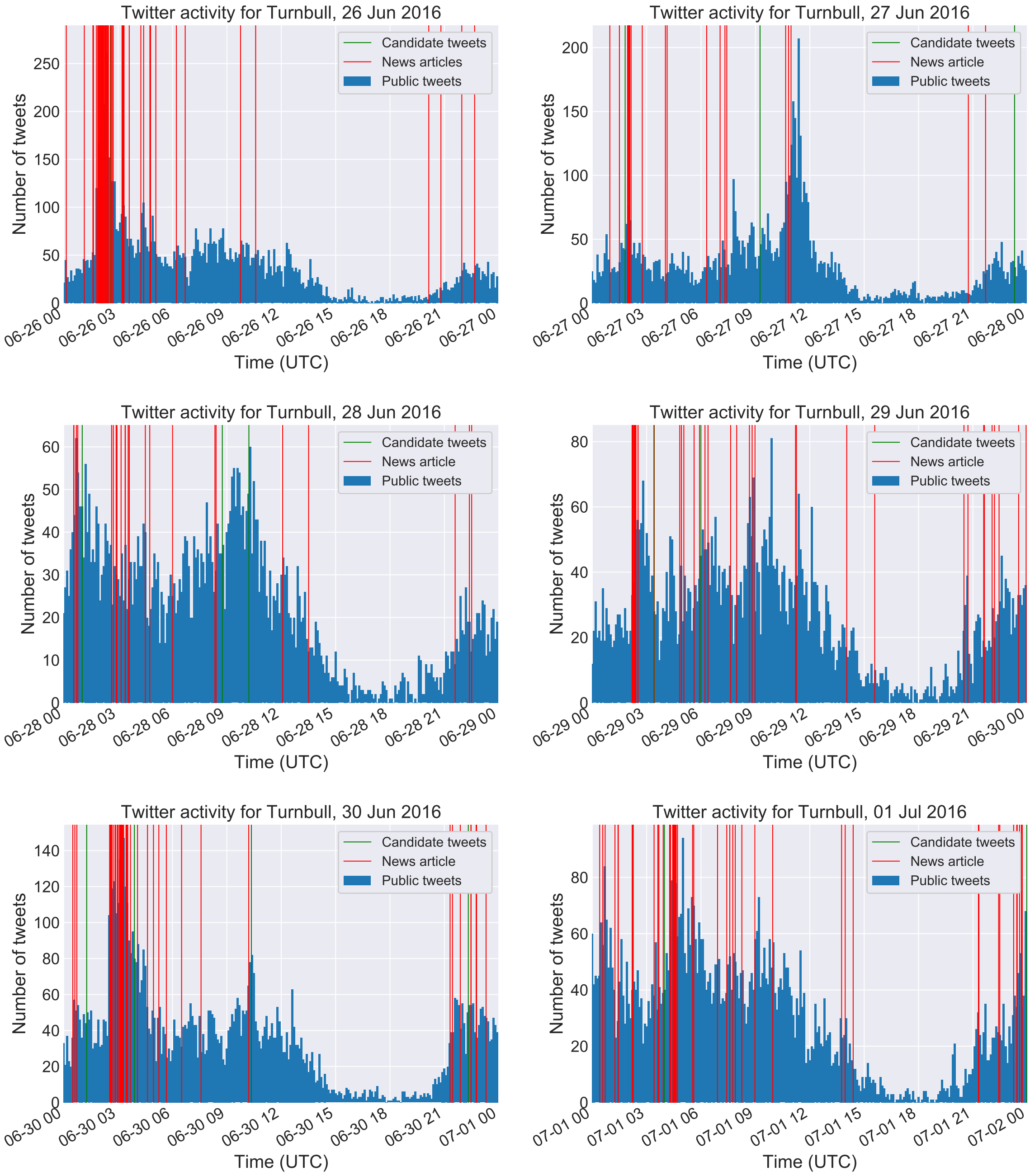


Figure A.1: Twitter activity for Malcolm Turnbull, 26 June to 1 July 2016. The public tweet data is split into six minute blocks. The diurnal cycles are clearly visible but due to the lower tweet counts than we observed for US politicians, the plots show a higher level of noise.

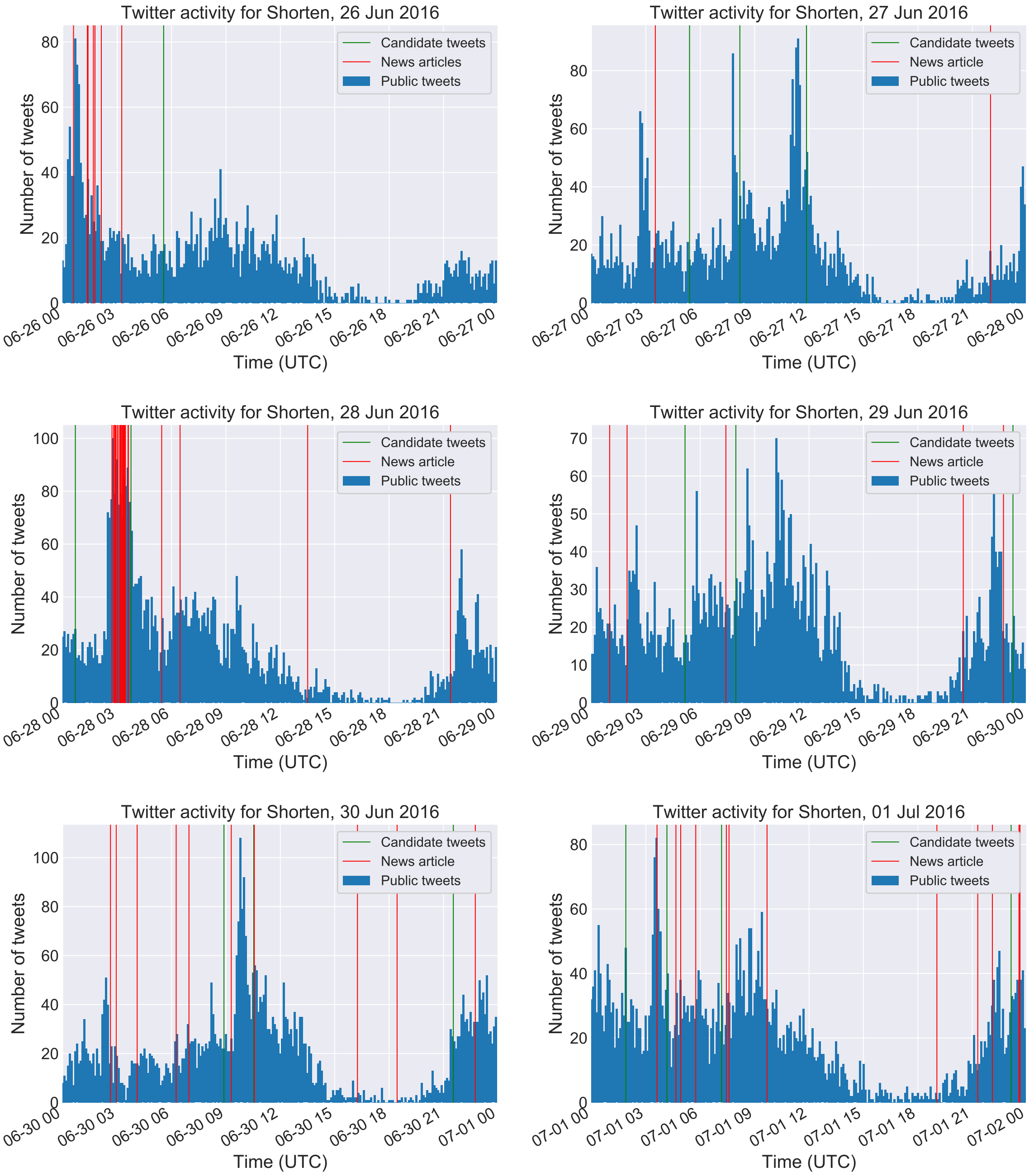


Figure A.2: Twitter activity for Bill Shorten, 26 June to 1 July 2016. The public tweet data is split into six-minute blocks. Again, the plots have a higher level of noise than we observed for US politicians in Chapter 3.

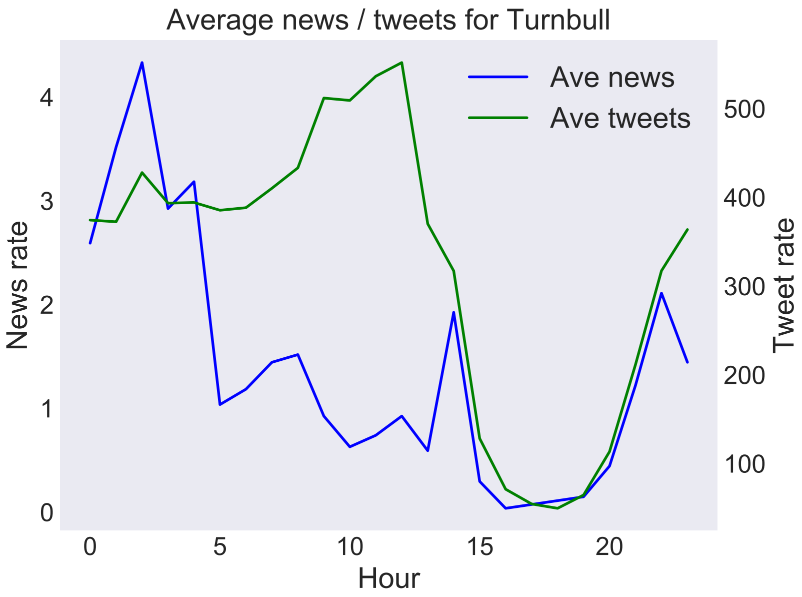


Figure A.3: Average hourly (UTC) tweet and news rates for Malcolm Turnbull in June 2016. The curves have a similar shape, except the tweet/news ratio is higher between 05:00 and 12:00 UTC, corresponding to the afternoon / evening (2 pm to 9 pm) on the East Coast of Australia.

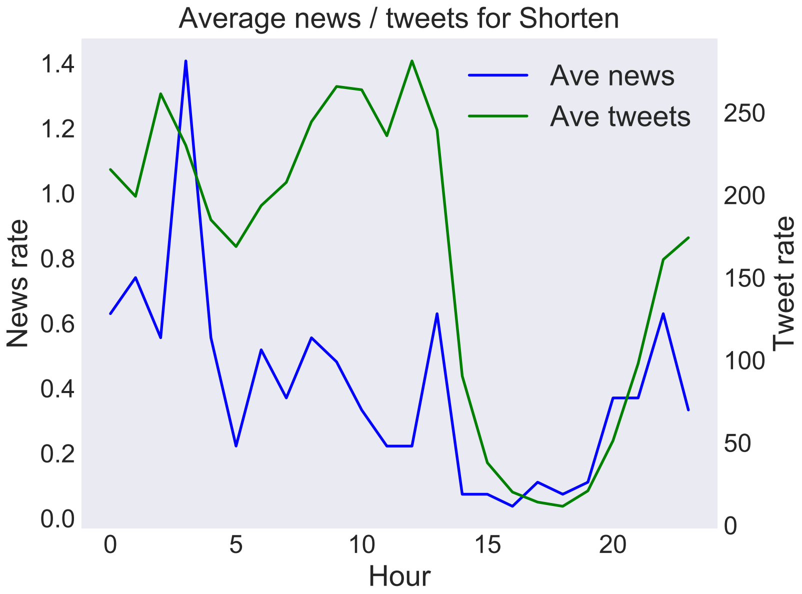


Figure A.4: Average hourly (UTC) tweet and news rates for Bill Shorten in June 2016. The shape is similar to that which was observed for Malcolm Turnbull with a higher tweet/news ratio in the afternoon / evening of Australia.

Appendix B

Supplementary material for Chapter 4

This appendix provides additional content for Chapter 4, *The Temporal Distribution of Retweets*.

B.1 Retweet distribution for Ted Cruz

In addition to the seed tweet examples for Donald Trump in Chapter 4, we present a similar analysis for Ted Cruz. We wish to check whether the overall characteristics are similar, and whether a power law with exponential cutoff does a good job of modeling the retweet rate. We collect the retweets from six Ted Cruz seed tweets between 8-11 March 2016. At the time, Ted Cruz was one of the favourites to be Republican nominee for President. Details of the tweets are in Table B.1.

We first analyse the retweets for the first three hours after an initial tweet from Ted Cruz. A histogram of the retweet frequencies is shown in Figure B.1. As expected, the retweet rate decays slowly over time. We plot the distribution on a log-log graph in Figure B.2. Similar to what we had for Donald Trump, we again observe a linear pattern indicating a power law distribution. Parameters for the lines of best fit are shown in Table B.2. We calculate power law parameter α values between 0.446 and 0.802, roughly similar to those obtained for Donald Trump's retweet distribution. As Ted Cruz's tweets have less retweets overall, the data is noisier and we have lower R^2 values than we had for Donald Trump. There is no clear diurnal effect over this short time period.

Label	Tweet text	Tweet date (UTC)
Cruz tweet A	In Hawaii, Idaho, Michigan, or Mississippi? I'm asking for your vote TODAY: #ChooseCruz	2016-03-08 13:23:16
Cruz tweet B	#ChooseCruz: https://t.co/BG5M3LCUBD https://t.co/E4x3feRNQA	2016-03-08 23:15:52
Cruz tweet C	SUNDAY: Join me, @GlennBeck, and Chuck Norris: https://t.co/UVjmlfPXsg	2016-03-09 01:25:08
Cruz tweet D	Missouri: Remember in November the Democrats who filibustered over 30 hours to fight against religious liberty. #DefendReligiousLiberty	2016-03-09 07:05:05
Cruz tweet E	WATCH LIVE: #CruzToVictory Rally in Miami, Florida at 10 am ET: https://t.co/J8h8NlAX4o #CruzCrew	2016-03-09 14:39:41
Cruz tweet F	#CruzCrew: we need your help to March to Victory. Join us –;	2016-03-11 18:55:35

Table B.1: Tweet details from Ted Cruz (Twitter: @tedcruz). These tweets were collected in the middle of the Republican nomination process, between the two key "Super Tuesday" primary dates.

Dataset	α	R^2
A	0.446	0.890
B	0.742	0.928
C	0.745	0.914
D	0.802	0.862
E	0.738	0.898
F	0.662	0.905

Table B.2: Power law parameters for three hour retweet collection - Ted Cruz. The values for α are more spread than for Donald Trump but are still centered around the interval $(0.6, 0.7)$. The R^2 values are lower, indicating that the data is slightly noisier.

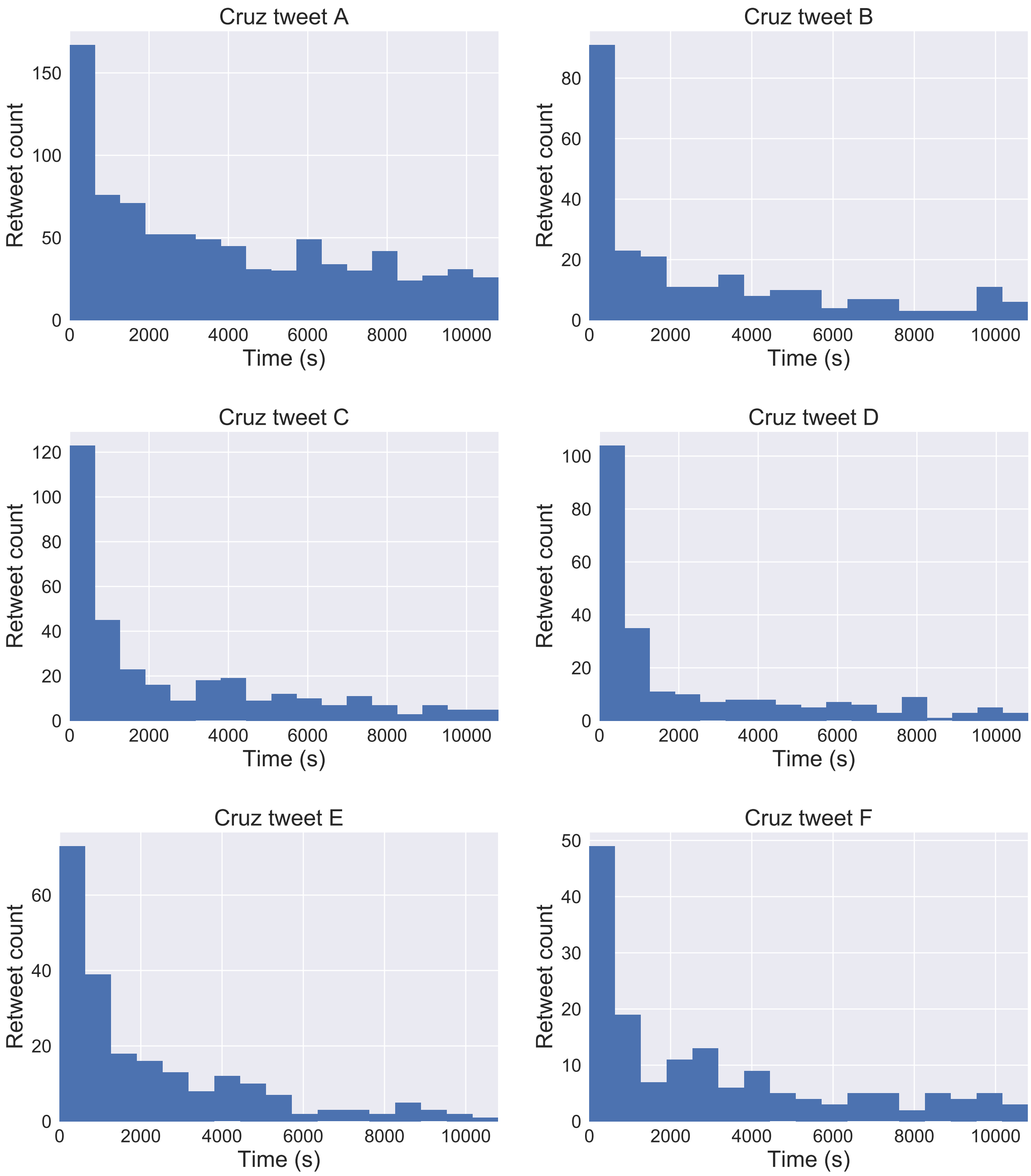


Figure B.1: Ted Cruz seed tweets: First three hours of retweet distribution. The retweet rate decays over time, with some level of noise.

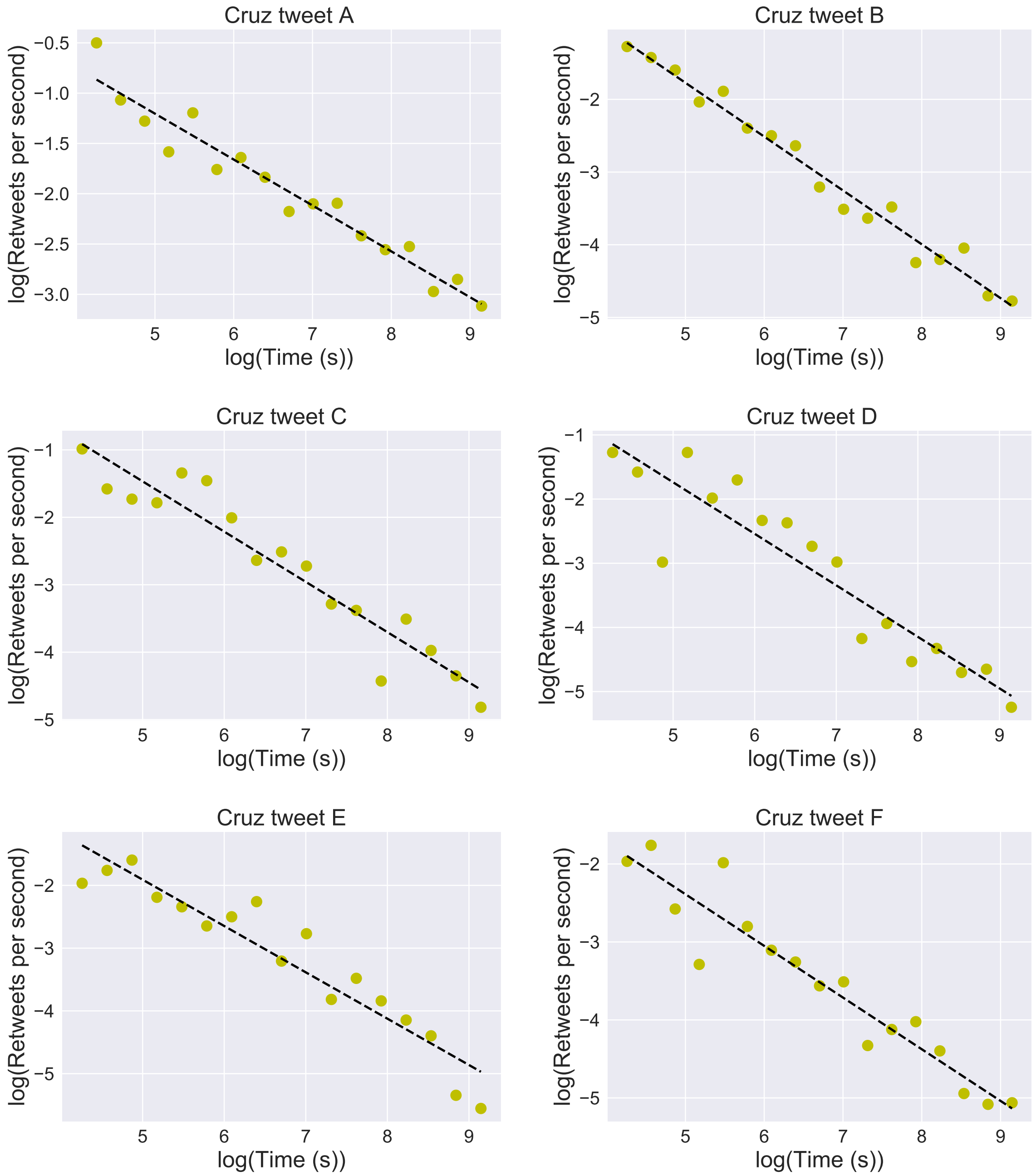


Figure B.2: Ted Cruz seed tweets: The first three hours of the retweet distribution presented on a log-log plot. The linear shape of the curve on the log-log plot indicates a power law distribution over this timeframe.

Dataset	A	b	c
A	1.099	0.2830	6.434×10^{-5}
B	14.187	0.8617	5.961×10^{-6}
C	1.210	0.7833	7.029×10^{-6}
D	3.005	0.6070	-7.313×10^{-6}
E	3.423	0.6391	5.156×10^{-5}
F	1.674	0.5891	1.797×10^{-5}

Table B.3: Ted Cruz seed tweets: Parameters for power law with exponential cutoff curves of best fit for first 24 hours

We also analyse the retweet rates for these Ted Cruz initial seed tweets over the first 24 hours after the initial tweet. Histograms of the retweet rate are shown in Figure B.3. The 24-hour retweet distribution curves for Ted Cruz, which we show in Figure B.4 are much more varied than we saw for Donald Trump. This is partially due to the much lower overall retweet rate and consequently, less data giving higher noise ratios. However, it is also due to the time of the tweets. In particular, tweet D was sent at 07:05:05 UTC corresponding to approximately 2 am US East Coast time. The retweet density histogram shows an extended spike over a period of 4 to 10 hours after the inital tweet occurred. It is reasonable to expect that the majority of Americans would be sleeping when the initial tweet occurred and only saw the retweet after they woke up. Looking at the 24 hour histograms for each of the six tweets, B and C have similar shapes. These tweets were sent at 23:15:52 UTC and 01:25:08 UTC respectively, similar times of the day.

The parameters for the power law with exponential cutoff best fit to the curve are given in Table B.3. The parameters are more varied than we observed for Donald Trump, due to the smaller retweet datasets.

Overall, we conclude that for this secondary dataset, like the seed tweets from Donald Trump, a power law does a good job of modelling the retweet distribution over the first three hours, and a power law with exponential cutoff performs better than a power law over the first 24 hours. We also observe that diurnal effects have a very strong effect on this dataset.

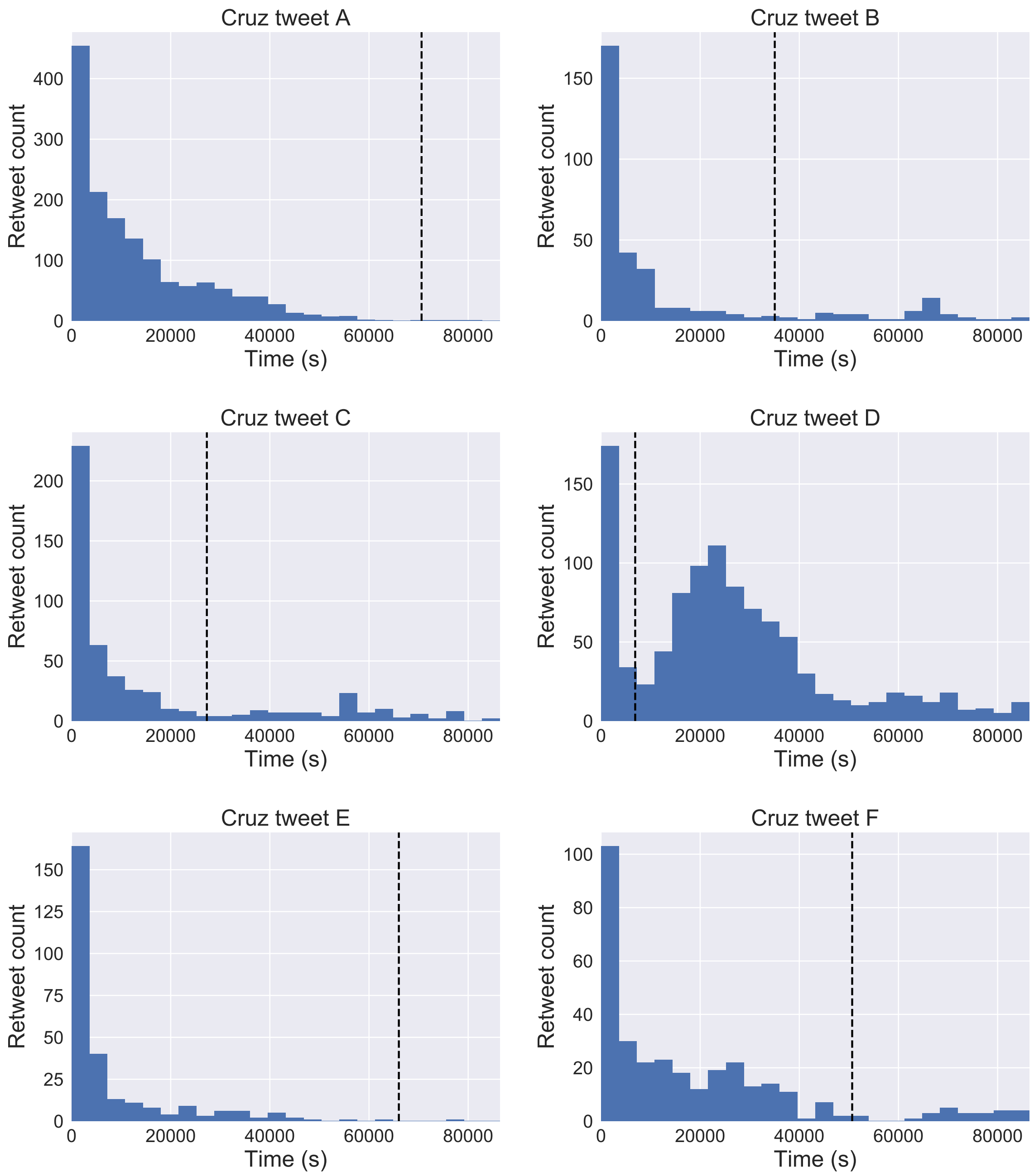


Figure B.3: Ted Cruz seed tweets: First 24 hours of retweet distribution. The curve shapes are heavily affected by the diurnal cycle. Tweet D for example was sent late at night, so it displays a large bump at around 25,000 seconds, approximately 7 hours after the initial tweet. This is the time in the United States when people wake up and check their social media accounts. The vertical dotted black line indicates 9am UTC, the time of lowest Twitter activity in the United States.

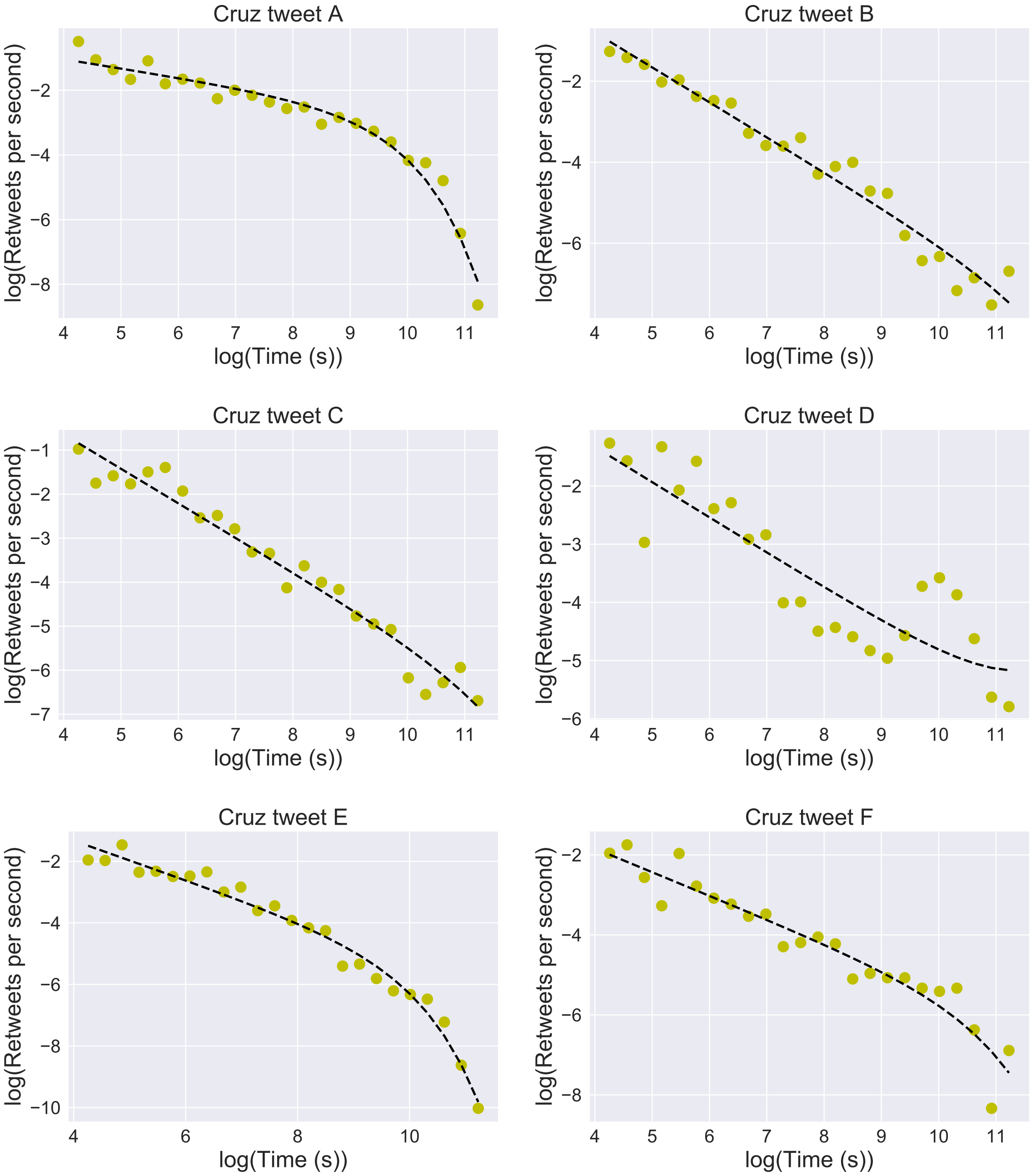


Figure B.4: Ted Cruz seed tweets: Power law with exponential cutoff for first 24 hours of retweet distribution. This generally provides a closer fit than a power law without cutoff. However, retweet set D is heavily affected by diurnal effects so the curve does not fit the dataset well.

B.2 Seed tweet ids

To enable replication of our work, the tweet ids of 1,676 seed tweets for which we collected retweets are available at https://github.com/pete1729/phd-thesis/tree/master/Chapter4.

B.3 Most followed Twitter accounts

Tweets were collected from the 100 twitter users with the most followers as of 6 April 2016. These Twitter users and their associated usernames were:

Twitter name	Twitter username
PRIYANKA	@priyankachopra
Kevin Durant	@KDTrey5
Hrithik Roshan	@iHrithik
Snoop Dogg	@SnoopDogg
Shugairi	@Shugairi
Alejandro Sanz	@AlejandroSanz
Twitter Sports	@TwitterSports
MTV	@MTV
Paris Hilton	@ParisHilton
The Economist	@TheEconomist
BBC News (World)	@BBCWorld
Facebook	@facebook
Ryan Seacrest	@RyanSeacrest
Deepika Padukone	@deepikapadukone
Beyonce Knowles	@Beyonce
Google	@google
Ivete Sangalo	@ivetesangalo
MohamadAlarefe	@MohamadAlarefe
Leonardo DiCaprio	@LeoDiCaprio
AGNEZ MO	@agnezmo
Kylie Jenner	@KylieJenner
Jim Carrey	@JimCarrey
Twitter en español	@TwitterEspanol
NASA	@NASA
Mariah Carey	@MariahCarey
Chris Brown	@chrisbrown
Christina Aguilera	@xtina
Vine	@vine
Kendall Jenner	@KendallJenner

Blake Shelton	@blakeshelton
Ed Sheeran	@edsheeran
Coldplay	@coldplay
Salman Khan	@BeingSalmanKhan
NFL	@NFL
Aamir Khan	@aamir_khan
FC Barcelona	@FCBarcelona
ashton kutcher	@aplusk
zayn	@zaynmalik
Real Madrid C. F.	@realmadrid
Shah Rukh Khan	@iamsrk
Kourtney Kardashian	@kourtneykardash
Narendra Modi	@narendramodi
Khloé	@khloekardashian
Avril Lavigne	@AvrilLavigne
David Guetta	@davidguetta
Marshall Mathers	@Eminem
Amitabh Bachchan	@SrBachchan
Conan O'Brien	@ConanOBrien
NICKI MINAJ	@NICKIMINAJ
NBA	@NBA
KANYE WEST	@kanyewest
Emma Watson	@EmWatson
Neymar Jr	@neymarjr
Pitbull	@pitbull
Louis Tomlinson	@Louis_Tomlinson
BBC Breaking News	@BBCBreaking
Liam	@Real_Liam_Payne
daniel tosh	@danieltosh
Alicia Keys	@aliciakeys
Neil Patrick Harris	@ActuallyNPH
CNN	@CNN
Kaka	@KAKA
KOE	@wizkhalifa
Niall Horan	@NiallOfficial
Bruno Mars	@BrunoMars
Adele	@Adele
SportsCenter	@SportsCenter
The New York Times	@nytimes
Lil Wayne WEEZY F	@LilTunechi
ESPN	@espn
Kevin Hart	@KevinHart4real
Harry Styles.	@Harry_Styles
P!nk	@Pink

One Direction	@onedirection
Bill Gates	@BillGates
Miley Ray Cyrus	@MileyCyrus
LeBron James	@KingJames
Drizzy	@Drake
Oprah Winfrey	@Oprah
Jennifer Lopez	@JLo
Demi Lovato	@ddlovato
CNN Breaking News	@cnnbrk
Shakira	@shakira
Ariana Grande	@ArianaGrande
jimmy fallon	@jimmyfallon
Instagram	@instagram
Britney Spears	@britneyspears
Kim Kardashian West	@KimKardashian
Cristiano Ronaldo	@Cristiano
Selena Gomez	@selenagomez
Taylor Swift	@taylorswift13
KATY PERRY	@katyperry
Justin Timberlake	@jtimberlake
Twitter	@twitter
Ellen DeGeneres	@TheEllenShow
Lady Gaga	@ladygaga
Rihanna	@rihanna
YouTube	@YouTube
Barack Obama	@BarackObama
Justin Bieber	@justinbieber

Appendix C

Supplementary Material for Chapter 5

This appendix provides additional content for Chapter 5, *Simulating Retweet Activity and Cascade Size Estimation*.

C.1 Integral of power law

We prove that under the conditions of Chapter 5, with $A > 0$, $T_0 > 0$ and $0 < b < 1$, the integral of a power law will be unbounded.

Theorem 1. *For* $A > 0$, $T_0 > 0$ *and* $0 < b < 1$,

$$\int_{T_0}^{\infty} At^{-b}dt \to \infty. \tag{C.1}$$

Proof. Let

$$\begin{aligned} I &= \lim_{T_f \to \infty} \int_{T_0}^{T_f} At^{-b}dt && \text{(C.2)} \\ &= \lim_{T_f \to \infty} \left[\frac{A}{1-b} t^{1-b} \right]_{T_0}^{T_f} && \text{(C.3)} \\ &= \lim_{T_f \to \infty} \frac{A}{1-b} T_f^{1-b} - \frac{A}{1-b} {T_0}^{1-b} && \text{(C.4)} \end{aligned}$$

Now for $0 < b < 1$, $\lim_{T_f \to \infty} T_F^{1-b} \to \infty$. Hence as $T_f \to \infty$, $I \to \infty$. □

Here we prove that the integral of an exponential cutoff will be bounded for parameters $A, b, c > 0$.

Theorem 2. *For $A > 0$, $b > 0$, $c > 0$ and $T_s > 1$,*

$$\int_{T_s}^{\infty} At^{-b}e^{-ct}dt < \infty \tag{C.5}$$

Proof. Let

$$\begin{aligned} I &= \lim_{T_f \to \infty} \int_{T_s}^{T_f} At^{-b}e^{-ct}dt && \text{(C.6)} \\ &< \lim_{T_f \to \infty} \int_{T_s}^{T_f} Ae^{-ct}dt \qquad [t^{-b} < 1 \text{ for } b > 0, t > 1] && \text{(C.7)} \\ &= \lim_{T_f \to \infty} \left[-\frac{A}{c}e^{-ct} \right]_{T_s}^{T_f} && \text{(C.8)} \\ &= \lim_{T_f \to \infty} -\frac{A}{c}e^{-cT_f} + \frac{A}{c}e^{-cT_s} && \text{(C.9)} \\ &= \frac{A}{c}e^{-cT_s} && \text{(C.10)} \end{aligned}$$

which is bounded. □

Appendix D

Supplementary Material for Chapter 6

This appendix provides additional content for Chapter 6, *Event Detection and Time Estimation from Twitter*.

D.1 Affinity measures

The following nine categories of affinity algorithms were identified by Cha *et al.* [30]. We found that the cosine similarity, within the inner product family, is the best suited for our purpose of measuring the textual similarity between tweets.

- L_p Minkowski family, involving a norm of some kind (e.g. Euclidean distance, Manhattan distance)
- L_1 family(e.g. Sorensen's quotient of similarity)
- Intersection family
- Inner product family (e.g. Inner product, cosine similarity)
- Fidelity family
- Squared L_2 family
- Shannon's entropy family
- Combinations
- Vicissitude

Category	Example algorithms
Partition	*k*-means, *k*-medoids, PAM, CLARA, CLARANS
Hierarchy	BIRCH, CURE, ROCK, Chameleon
Fuzzy theory	FCM, FCS, MM
Distribution	DBCLASD, GMM
Density	DBSCAN, OPTICS, Mean-shift
Graph theory	CLICK, MST
Grid	STING, CLIQUE
Fractal theory	FC
Model	COBWEB, GMM, SOM, ART

Table D.1: Traditional clustering algorithms, as summarised by [174].

Category	Example algorithms
Kernel	kernel *k*-means, kernel SOM, kernel FCM, ...
Ensemble	CSPA, HGPA, MCLA, VM, HCE, LAC, ...
Swarm intelligence	ACO_based(LF), PSO_based, SFLA_based, ...
Quantum theory	QC, DQCM
Spectral graph theory	SM, NJW
Affinity propagation	AP
Density and distance	DD
Spacial data	DBSCAN, STING, Wavecluster, CLARANS
Data stream	STREAM, CluStream, HPStream, DenStream
Large scale data	*k*-means, BIRCH, CLARA, DBSCAN, ...

Table D.2: Modern clustering algorithms, as summarised by [174]. Affinity propagation was selected for our tweet clustering method SMERC.

D.2 Clustering methods

Xu [174] summarised traditional and modern clustering algorithms as shown in Tables D.1 and D.2. Of these algorithms, *k*-means is the most commonly used but did not meet our requirements as it needs the number of clusters to be specified beforehand. The density based clustering algorithm DBSCAN was a candidate for our clustering method SMERC, but affinity propagation was found to have the best peformance.

References

1. Deepak Agarwal, Bee-Chung Chen, and Pradheep Elango. Spatio-temporal models for estimating click-through rate. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 21–30, New York, NY, USA, 2009. ACM. (Cited on pages 22 and 99).

2. Charu C. Aggarwal, Jiawei Han, Jianyong Wang, and Philip S. Yu. A framework for projected clustering of high dimensional data streams. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, VLDB '04, pages 852–863. VLDB Endowment, 2004. (Cited on pages 19 and 93).

3. Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–97, Jan 2002. (Cited on pages 16 and 84).

4. Saleem Alhabash and Mengyan Ma. A tale of four platforms: Motivations and uses of Facebook, Twitter, Instagram, and Snapchat among college students? *Social Media + Society*, 3(1):2056305117691544, 2017. (Cited on page 1).

5. Jeff Alstott, Ed Bullmore, and Dietmar Plenz. powerlaw: A python package for analysis of heavy-tailed distributions. *PLOS ONE*, 9(1):1–11, 01 2014. (Cited on page 16).

6. Cory L. Armstrong and Fangfang Gao. Now tweet this: How news organizations use Twitter. *Electronic News*, 4(4):218–235, 2010. (Cited on page 43).

7. Farzindar Atefeh and Wael Khreich. A survey of techniques for event detection in Twitter. *Computational Intelligence*, 31(1):132–164, February 2015. (Cited on pages 24, 25, and 26).

8. James P. Bagrow. *Information Spreading During Emergencies and Anomalous Events*, pages 269–286. Springer International Publishing, Cham, 2018. (Cited on page 1).

9. Per Bak, Chao Tang, and Kurt Wiesenfeld. Self-organized criticality: An Expla-

nation of 1/f noise. *Phys. Rev. Lett.*, 59:381–384, 1987. (Cited on page 17).

10. Stephanie Alice Baker. From the criminal crowd to the "mediated crowd": the impact of social media on the 2011 English riots. *Safer Communities*, 11(1):40–49, 2012. (Cited on page 1).

11. Eytan Bakshy, Jake M. Hofman, Winter A. Mason, and Duncan J. Watts. Everyone's an influencer: Quantifying influence on Twitter. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 65–74, New York, NY, USA, 2011. ACM. (Cited on page 21).

12. Albert-László Barabási. The origin of bursts and heavy tails in human dynamics. *Nature*, 435:207, 2005. (Cited on pages 18, 19, 93, 95, and 96).

13. Albert-László Barabási and Reka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999. (Cited on page 18).

14. Albert-László Barabási, Reka Albert, and Hawoong Jeong. Mean field theory for scale-free random networks. *Physica A Statistical Mechanics and its Applications*, 272:173–187, October 1999. (Cited on pages 18 and 93).

15. Heiko Bauke. Parameter estimation for power-law distributions by maximum likelihood methods. *The European Physical Journal B*, 58(2):167–173, 2007. (Cited on page 16).

16. Yoshua Bengio and Yves Grandvalet. No unbiased estimator of the variance of k-fold cross-validation. *Journal of Machine Learning Research*, 5:1089–1105, December 2004. (Cited on page 38).

17. Donald J. Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, AAAIWS'94, pages 359–370. AAAI Press, 1994. (Cited on page 24).

18. Luis M. A. Bettencourt, Ariel Cintron-Arias, David I. Kaiser, and Carlos Castillo-Chavez. The power of a good idea: Quantitative modeling of the spread of ideas from epidemiological models. *Physica A: Statistical Mechanics and its Applications*, 364:513–536, 5 2006. (Cited on page 13).

19. David R. Bild, Yue Liu, Robert P. Dick, Z. Morley Mao, and Dan S. Wallach. Aggregate characterization of user behavior in Twitter and analysis of the retweet

graph. *ACM Transactions on Internet Technology*, 15(1):4:1–4:24, March 2015. (Cited on page 14).

20. Johan Bollen, Huina Mao, and Alberto Pepe. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. *Association for the Advancement of Artificial Intelligence*, 2011. (Cited on page 5).

21. Alexandre Bovet and Hernán A. Makse. Influence of fake news in twitter during the 2016 US presidential election. *CoRR*, abs/1803.08491, 2018. (Cited on page 12).

22. Anna D. Broido and Aaron Clauset. Scale-free networks are rare. *ArXiv e-prints 1801.03400*, January 2018. (Cited on page 86).

23. Michal Brzezinski. Power laws in citation distributions: Evidence from scopus. *CoRR*, abs/1402.3890, 2014. (Cited on page 17).

24. Jaroslav Bukovina. Social media and capital markets. An overview. *Procedia - Social and Behavioral Sciences*, 220:70 – 78, 2016. 19th International Conference Enterprise and Competitive Environment 2016. (Cited on page 6).

25. Pete Burnap, Rachel Gibson, Luke Sloan, Rosalynd Southern, and Matthew Williams. 140 characters to victory?: Using Twitter to predict the UK 2015 general election. *Electoral Studies*, 41:230 – 233, 2016. (Cited on pages 5 and 6).

26. Jean Carlson and John Doyle. Highly optimized tolerance: A mechanism for power laws in designed systems. *Physical Review E*, 60:1412–1427, Aug 1999. (Cited on page 18).

27. Lidia Ceriani and Paolo Verme. The origins of the Gini index: extracts from Variabilita e Mutabilita (1912) by Corrado Gini. *The Journal of Economic Inequality*, 10(3):421–443, Sep 2012. (Cited on page 20).

28. Andrea Ceron, Luigi Curini, and Stefano M Iacus. First and second-level agenda setting in the Twittersphere: An application to the Italian political debate. *Journal of Information Technology & Politics*, 13(2):159–174, 2016. (Cited on page 43).

29. Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and Krishna P. Gummadi. Measuring user influence in Twitter: The million follower fallacy. In *4th International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2010. (Cited on pages 20 and 21).

30. Sung-Hyuk Cha. Comprehensive survey on distance/similarity measures between probability density functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, 1(4):300–307, 2007. (Cited on pages 26 and 159).

31. Deepayan Chakrabarti and Kunal Punera. Event summarization using tweets. In *Proc. 6th AAAI Int. Conf. on Weblogs and Social Media*, 2011. (Cited on page 24).

32. Eric M. Clark, Jake R. Williams, Richard. A. Galbraith, Chris. A. Jones, Christopher. M. Danforth, and Peter. S. Dodds. Sifting robotic from organic text: A natural language approach for detecting automation on Twitter. *Journal of Computational Science*, 16:1–7, 2016. (Cited on page 4).

33. Aaron Clauset, Cosma Rohilla Shalizi, and Mark E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, November 2009. (Cited on pages 16, 17, 66, 82, 84, 85, 86, and 90).

34. Bethany A. Conway, Kate Kenski, and Di Wang. The rise of Twitter in the political campaign: Searching for intermedia agenda-setting effects in the presidential primary. *Journal of Computer-Mediated Communication*, 20(4):363–380, 2015. (Cited on pages 3 and 12).

35. Riley Crane and Didier Sornette. Robust dynamic classes revealed by measuring the response function of a social system. *Proceedings of the National Academy of Sciences*, 105(41):15649–15653, 2008. (Cited on pages 13, 14, 15, and 65).

36. Aron Culotta. Towards detecting influenza epidemics by analyzing Twitter messages. In *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, pages 115–122, New York, NY, USA, 2010. ACM. (Cited on page 5).

37. George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314, December 1989. (Cited on page 37).

38. Clayton Allen Davis, Onur Varol, Emilio Ferrara, Alessandro Flammini, and Filippo Menczer. Botornot: A system to evaluate social bots. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, pages 273–274, 2016. (Cited on pages 89 and 108).

39. Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of*

the 30th International Conference on Neural Information Processing Systems, NIPS'16, pages 3844–3852, USA, 2016. Curran Associates Inc. (Cited on page 36).

40. Anna Deluca and Álvaro Corral. Fitting and goodness-of-fit test of non-truncated and truncated power-law distributions. *Acta Geophysica*, 61(6):1351–1394, Dec 2013. (Cited on page 90).

41. Bhuwan Dhingra, Zhong Zhou, Dylan Fitzpatrick, Michael Muehl, and William Cohen. Tweet2Vec: Character-based distributed representations for social media. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 269–274. Association for Computational Linguistics, 2016. (Cited on pages 26, 29, 138, and 140).

42. Yi Ding and Xue Li. Time weight collaborative filtering. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, pages 485–492, New York, NY, USA, 2005. (Cited on pages 19 and 93).

43. Peter Sheridan Dodds and D. J. Watts. A generalized model of social and biological contagion. *J. Theor. Biol.*, 232:587–604, 2005. (Cited on page 14).

44. Christian Doerr, Norbert Blenn, and Piet Van Mieghem. Lognormal infection times of online information spread. *PLoS ONE*, 8(5):1–6, 05 2013. (Cited on pages 17 and 93).

45. Pietro Ducange and Michela Fazzolari. Social sensing and sentiment analysis: Using social media as useful information source. In *2017 International Conference on Smart Systems and Technologies (SST)*, pages 301–306, Oct 2017. (Cited on page 1).

46. Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Number 57 in Monographs on Statistics and Applied Probability. Chapman & Hall/CRC, Boca Raton, Florida, USA, 1993. (Cited on page 40).

47. V. A. Epanechnikov. Non-Parametric estimation of a multivariate probability density. *Theory of Probability and Its Applications*, 14(1):153–158, 1969. (Cited on page 114).

48. Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages

226–231. AAAI Press, 1996. (Cited on page 119).

49. Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the Internet topology. *SIGCOMM Computer Communication Review*, 29(4):251–262, August 1999. (Cited on pages 16 and 84).

50. Orhan Firat, Kyunghyun Cho, Baskaran Sankaran, Fatos T. Yarman Vural, and Yoshua Bengio. Multi-way, multilingual neural machine translation. *Computer Speech and Language*, 45(C):236–252, September 2017. (Cited on page 36).

51. Friend Or Follow. Twitter: Most followers. `http://friendorfollow.com/twitter/most-followers/`, 2016. Accessed: 2016-03-15. (Cited on page 81).

52. Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007. (Cited on page 27).

53. Peter Gabrovsek, Darko Aleksovski, Igor Mozetic, and Miha Grcar. Twitter sentiment around the earnings announcement events. *PLoS ONE*, 12(2):1–21, 02 2017. (Cited on page 5).

54. Ryan J. Gallagher, Andrew J. Reagan, Christopher M. Danforth, and Peter Sheridan Dodds. Divergent discourse between protests and counter-protests: #blacklivesmatter and #alllivesmatter. *CoRR*, abs/1606.06820, 2016. (Cited on page 112).

55. Yerali Gandica, João Carvalho, Fernando Sampaio dos Aidos, Renaud Lambiotte, and Timoteo Carletti. Stationarity of the inter-event power-law distributions. *PLoS ONE*, 12(3):1 – 10, 2017. (Cited on page 21).

56. Shuai Gao, Jun Ma, and Zhumin Chen. Modeling and predicting retweeting dynamics on microblogging platforms. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 107–116, New York, NY, USA, 2015. ACM. (Cited on pages 32, 68, and 78).

57. M. Gavrilov, D. Anguelov, P. Indyk, and R. Motwani. Mining the stock market: Which measure is best. In *Proceedings of the sixth ACM SIGKDD international conference on knowledge discovery and data mining*, 2000. (Cited on page 24).

58. Daniel Gayo-Avello. "I Wanted to Predict Elections with Twitter and all I got was this Lousy Paper" – A Balanced Survey on Election Prediction using Twitter Data. *ArXiv e-prints 1204.6441*, April 2012. (Cited on page 6).

59. Mehreen Gillani, Muhammad U. Ilyas, Saad Saleh, Jalal S. Alowibdi, Naif Aljohani, and Fahad S. Alotaibi. Post Summarization of Microblogs of Sporting Events. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, pages 59–68, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee. (Cited on pages 3, 24, 25, 110, and 117).

60. Ashish Goel, Kamesh Munagala, Aneesh Sharma, and Hongyang Zhang. A note on modeling retweet cascades on Twitter. In *Proceedings of the 12th International Workshop on Algorithms and Models for the Web Graph - Volume 9479*, WAW 2015, pages 119–131, New York, NY, USA, 2015. Springer-Verlag New York, Inc. (Cited on pages 1, 8, and 21).

61. Yoav Goldberg and Graeme Hirst. *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers, 2017. (Cited on page 28).

62. Clive Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–38, 1969. (Cited on pages 35 and 58).

63. Clive W.J. Granger. Time Series Analysis, Cointegration, and Applications. *American Economic Review*, 94(3):421–425, June 2004. (Cited on page 35).

64. Caitlin Gray, Lewis Mitchell, and Matthew Roughan. Super-blockers and the effect of network structure on information cascades. In *Companion Proceedings of the Web Conference 2018*, WWW '18, pages 1435–1441, Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee. (Cited on page 20).

65. Josh Haner. The data behind Trump's Twitter takeover. `https://www.politico.com/magazine/story/2016/04/donald-trump-2016-twitter-takeover-213861`, 2018. Accessed: 2018-02-21. (Cited on page 68).

66. David Harris and Sarah Harris. *Digital Design and Computer Architecture, Second Edition*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition, 2012. (Cited on page 34).

67. Zellig Harris. Distributional structure. *Word*, 10(23):146–162, 1954. (Cited on page 29).

68. Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR,

Upper Saddle River, NJ, USA, 2nd edition, 1998. (Cited on pages 36 and 61).

69. Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, and Joaquin Quiñonero Candela. Practical lessons from predicting clicks on ads at Facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, ADKDD'14, pages 5:1–5:9, New York, NY, USA, 2014. ACM. (Cited on page 2).

70. Nathan Oken Hodas and Kristina Lerman. How visibility and divided attention constrain social contagion. In *Proceedings of the 2012 ASE/IEEE International Conference on Social Computing and 2012 ASE/IEEE International Conference on Privacy, Security, Risk and Trust*, SOCIALCOM-PASSAT '12, pages 249–257, Washington, DC, USA, 2012. IEEE Computer Society. (Cited on pages 3, 8, and 15).

71. William Hoiles, Anup Aprem, and Vikram Krishnamurthy. Engagement and popularity dynamics of YouTube videos and sensitivity to meta-data. *IEEE Transactions on Knowledge and Data Engineering*, 29(7):1426–1437, July 2017. (Cited on page 2).

72. Mengdie Hu, Shixia Liu, Furu Wei, Yingcai Wu, John Stasko, and Kwan-Liu Ma. Breaking news on Twitter. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 2751–2754, New York, NY, USA, 2012. ACM. (Cited on page 43).

73. Rob J. Hyndman and Anne B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006. (Cited on pages 23, 33, 34, and 104).

74. Nitin Indurkhya and Fred J. Damerau. *Handbook of Natural Language Processing*. Chapman & Hall/CRC, 2nd edition, 2010. (Cited on page 28).

75. InternetLiveStats. Twitter usage statistics. `http://www.internetlivestats.com/twitter-statistics/`, March 2018. Accessed: 2018-03-16. (Cited on page 2).

76. Yannis M. Ioannides and Henry G. Overman. Zipf's law for cities: an empirical examination. *Regional Science and Urban Economics*, 33(2):127 – 137, 2003. (Cited on page 18).

77. Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988. (Cited on page 27).

78. Zhi-Qiang Jiang, Wen-Jie Xie, Ming-Xia Li, Boris Podobnik, Wei-Xing Zhou, and H. Eugene Stanley. Calling patterns in human communication dynamics. *Proceedings of the National Academy of Science*, 110:1600–1605, January 2013. (Cited on page 31).

79. Wahiba Ben Abdessalem Karaa and Nidhal Gribâa. Information retrieval with Porter Stemmer: A new version for English. In Dhinaharan Nagamalai, Ashok Kumar, and Annamalai Annamalai, editors, *Advances in Computational Science, Engineering and Information Technology*, pages 243–254, Heidelberg, 2013. Springer International Publishing. (Cited on page 28).

80. Eamonn Keogh and Chotirat Ann Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386, Mar 2005. (Cited on page 24).

81. Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, pages 1137–1143, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. (Cited on page 38).

82. Michel Krieger and David Ahn. TweetMotif: Exploratory search and topic summarization for Twitter. In *Proc. of AAAI Conference on Weblogs and Social*, 2010. (Cited on pages 4 and 24).

83. Andrey Kupavskii, Liudmila Ostroumova, Alexey Umnov, Svyatoslav Usachev, Pavel Serdyukov, Gleb Gusev, and Andrey Kustarev. Prediction of retweet cascade size over time. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 2335–2338, New York, NY, USA, 2012. (Cited on pages 1, 8, 21, and 101).

84. Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is Twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 591–600, New York, NY, USA, 2010. ACM. (Cited on pages 11 and 43).

85. David Lazer, Ryan Kennedy, Gary King, and Alessandro Vespignani. The parable of Google flu: Traps in big data analysis. *Science*, 343(14 March):1203–1205, 2014. (Cited on page 5).

86. Quoc Le and Tomas Mikolov. Distributed representations of sentences and doc-

uments. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1188–II–1196. JMLR.org, 2014. (Cited on page 29).

87. Angela M. Lee. Social media and speed-driven journalism: Expectations and practices. *International Journal on Media Management*, 17(4):217–239, 2015. (Cited on page 1).

88. Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 497–506. ACM New York, NY, USA, 2009. (Cited on pages 1 and 14).

89. Chenliang Li, Aixin Sun, and Anwitaman Datta. Twevent: Segment-based event detection from tweets. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 155–164, New York, NY, USA, 2012. ACM. (Cited on page 25).

90. Lei Li, Li Zheng, Fan Yang, and Tao Li. Modeling and broadening temporal user interest in personalized news recommendation. *Expert Systems with Applications*, 41(7):3168–3177, June 2014. (Cited on pages 1, 19, 93, and 116).

91. Nan-Nan Li, Ning Zhang, and Tao Zhou. Empirical analysis on temporal statistics of human correspondence patterns. *Physica A: Statistical Mechanics and its Applications*, 387(25):6391 – 6394, 2008. (Cited on page 17).

92. Shuang Li, Huchuan Lu, Zhe Lin, Xiaohui Shen, and Brian Price. Adaptive metric learning for saliency detection. *IEEE Transactions on Image Processing*, 24(11):3321–3331, Nov 2015. (Cited on page 36).

93. Yao Li, Lingqiao Liu, Chunhua Shen, and Anton Van Hengel. Mining mid-level visual patterns with deep CNN activations. *International Journal of Computer Vision*, 121(3):344–364, February 2017. (Cited on page 36).

94. Xiaomo Liu, Quanzhi Li, Armineh Nourbakhsh, Rui Fang, Merine Thomas, Kajsa Anderson, Russ Kociuba, Mark Vedder, Steven Pomerville, Ramdev Wudali, Robert Martin, John Duprey, Arun Vachher, William Keenan, and Sameena Shah. Reuters tracer: A large scale system of detecting and verifying real-time news events from Twitter. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 207–216, New

York, NY, USA, 2016. ACM. (Cited on pages 7, 12, 91, 109, and 134).

95. James O. Lloyd-Smith, Sebastian J. Schreiber, P. E. Kopp, and Wayne M. Getz. Superspreading and the effect of individual variation on disease emergence. *Nature*, 438(7066):355–359, November 2005. (Cited on page 20).

96. Edward Loper and Steven Bird. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. (Cited on pages 28 and 118).

97. Yao Lu, Peng Zhang, Yanan Cao, Yue Hu, and Li Guo. On the frequency distribution of retweets. *Procedia Computer Science*, 31:747 – 753, 2014. (Cited on pages 1, 8, 14, and 21).

98. Hans Peter Luhn. Key word-in-context index for technical literature (kwic index). *American Documentation*, 11(4):288–295, 1960. (Cited on page 28).

99. James Macqueen. Some methods for classification and analysis of multivariate observations. In *5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967. (Cited on page 24).

100. Hosam Mahmoud. *Polya Urn Models*. Chapman & Hall/CRC, 1st edition, 2008. (Cited on page 18).

101. Luís Marujo, Wang Ling, Isabel Trancoso, Chris Dyer, Alan W. Black, Anatole Gershman, David Martins de Matos, João Paulo da Silva Neto, and Jaime G. Carbonell. Automatic keyword extraction on Twitter. In *ACL*, 2015. (Cited on page 4).

102. Peter Mathews, Caitlin Gray, Lewis Mitchell, Giang T. Nguyen, and Nigel G. Bean. SMERC: Social media event response clustering using textual and temporal information. In *IEEE International Conference on Big Data, Big Data 2018, Seattle, WA, USA, December 10-13, 2018*, pages 3695–3700, 2018. (Cited on page 110).

103. Peter Mathews, Lewis Mitchell, Giang Nguyen, and Nigel Bean. The nature and origin of heavy tails in retweet activity. In *The 26th International Conference on World Wide Web Companion*, pages 1493–1498, 2017. (Cited on pages 8, 21, and 66).

104. Michael Mathioudakis and Nick Koudas. Twittermonitor: Trend detection over the Twitter stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, pages 1155–1158, New York, NY, USA, 2010. ACM. (Cited on page 25).

105. Yasuko Matsubara, Yasushi Sakurai, B. Aditya Prakash, Lei Li, and Christos Faloutsos. Rise and fall patterns of information diffusion: Model and implications. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 6–14, New York, NY, USA, 2012. ACM. (Cited on page 15).

106. Tyler H. McCormick, Hedwig Lee, Nina Cesare, Ali Shojaie, and Emma S. Spiro. Using Twitter for demographic and social science research: Tools for data collection and processing. *Sociological Methods & Research*, 46(3):390–421, 2017. (Cited on page 5).

107. Matus Medo, Giulio Cimini, and Stanislao Gualdi. Temporal effects in the growth of networks. *Physical Review Letters*, 107(23):238701, December 2011. (Cited on page 20).

108. Donald Metzler, Susan Dumais, and Christopher Meek. Similarity measures for short segments of text. In *Proceedings of the 29th European Conference on IR Research*, ECIR'07, pages 16–27, Berlin, Heidelberg, 2007. Springer-Verlag. (Cited on page 26).

109. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013. (Cited on pages 26, 29, and 118).

110. Michael Mitzenmacher. A brief history of generative models for power law and lognormal distributions. *Internet Mathematics*, 1:226–251, 2004. (Cited on pages 17 and 127).

111. Alfredo J. Morales, Vaibhav Vavilala, Rosa M. Benito, and Yaneer Bar-Yam. Global patterns of synchronization in human communications. *Journal of The Royal Society Interface*, 14(128), 2017. (Cited on pages 5 and 49).

112. Juergen Mueller and Gerd Stumme. Gender inference using statistical name characteristics in Twitter. In *Proceedings of the 3rd Multidisciplinary International*

Social Networks Conference on SocialInformatics 2016, Data Science 2016, MISNC, SI, DS 2016, pages 47:1–47:8, New York, NY, USA, 2016. ACM. (Cited on page 5).

113. Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, USA, 2010. Omnipress. (Cited on page 37).

114. Surya Nepal, Uma Srinivasan, and Graham Reynolds. Automatic detection of 'goal' segments in basketball videos. In *Proceedings of the Ninth ACM International Conference on Multimedia*, MULTIMEDIA '01, pages 261–269, New York, NY, USA, 2001. ACM. (Cited on page 110).

115. Mark E. J. Newman. Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*, 46(5):323–351, 2005. (Cited on pages 17 and 18).

116. Huy Nguyen and Rong Zheng. A data-driven study of influences in twitter communities. *CoRR*, abs/1307.4264, 2013. (Cited on page 13).

117. Chris Norval and Tristan Henderson. Contextual consent: Ethical mining of social media for health research. *ArXiv e-prints 1701.07765*, January 2017. (Cited on page 6).

118. Andre Nyembwe. *Development and International Economics: Essays in Memory of Michel Norro*. Hors collection. Presses Universitaires de Louvain, 2008. (Cited on page 35).

119. Sei Onishi, Yuto Yamaguchi, and Hiroyuki Kitagawa. Real-time relevance matching of news and tweets. In Christophe Debruyne, Hervé Panetto, Robert Meersman, Tharam Dillon, Georg Weichhart, Yuan An, and Claudio Agostino Ardagna, editors, *On the Move to Meaningful Internet Systems: OTM 2015 Conferences*, pages 109–126, Cham, 2015. Springer International Publishing. (Cited on page 12).

120. Venkata S. Pagolu, Kamal N. Reddy, Ganapati Panda, and Babita Majhi. Sentiment analysis of Twitter data for predicting stock market movements. In *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, pages 1345–1350, Oct 2016. (Cited on page 5).

121. Romualdo Pastor-Satorras, Claudio Castellano, Piet Van Mieghem, and Alessan-

dro Vespignani. Epidemic processes in complex networks. *Reviews of Modern Physics*, 87:925–979, Aug 2015. (Cited on page 13).

122. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort., Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. (Cited on pages 29, 60, 114, and 119).

123. Saša Petrović, Miles Osborne, and Victor Lavrenko. Using paraphrases for improving first story detection in news and Twitter. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 338–346, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. (Cited on pages 3 and 12).

124. Real Clear Politics. Latest 2016 GOP Presidential Primary Polls. `https://www.realclearpolitics.com/epolls/latest_polls/gop_pres_primary/`, 2018. Accessed: 2018-02-21. (Cited on page 68).

125. William Rand, Jeffrey Herrmann, Brandon Schein, and Neza Vodopivec. An agent-based model of urgent diffusion in social media. *Journal of Artificial Societies and Social Simulation*, 18(2):1, 2015. (Cited on page 31).

126. William J. Reed. The Pareto, Zipf and other power laws. *Economics Letters*, 74(1):15 – 19, 2001. (Cited on pages 18 and 72).

127. Albert Reuther, Chansup Byun, William Arcand, David Bestor, Bill Bergeron, Matthew Hubbell, Michael Jones, Peter Michaleas, Andrew Prout, Antonio Rosa, and Jeremy Kepner. Scalable system scheduling for HPC and big data. *CoRR*, abs/1705.03102, 2017. (Cited on page 107).

128. Matthew Rosenberg, Nicholas Confessore, and Carole Cadwalladr. How Trump consultants exploited the Facebook data of millions. `https://www.nytimes.com/2018/03/17/us/politics/cambridge-analytica-trump-campaign.html`, March 2018. Accessed: 2018-03-22. (Cited on page 6).

129. P. Rosin. The laws governing the fineness of powdered coal. *Journal of the Institute of Fuel*, pages 29–36, 1933. (Cited on page 31).

130. Yong Rui, Anoop Gupta, and Alex Acero. Automatically extracting highlights for TV baseball programs. In *Proceedings of the Eighth ACM International Conference on Multimedia*, Multimedia '00, pages 105–115, New York, NY, USA, 2000. ACM. (Cited on page 110).

131. Arif Mohaimin Sadri, Samiul Hasan, Satish V. Ukkusuri, and Manuel Cebrián. Understanding information spreading in social media during Hurricane Sandy: User activity and network properties. *CoRR*, abs/1706.03019, 2017. (Cited on pages 15 and 82).

132. Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. In *Proceedings of the Nineteenth International WWW Conference (WWW2010). ACM*, 2010. (Cited on pages 30 and 109).

133. Gerard Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989. (Cited on pages 29 and 118).

134. Abraham Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, July 1964. (Cited on pages 39 and 78).

135. M. Angeles Serrano, Alessandro Flammini, and Filippo Menczer. Modeling statistical properties of written text. *PLoS ONE*, 4(4):1 – 8, 2009. (Cited on page 30).

136. Beaux Sharifi, Mark-Anthony Hutton, and Jugal Kalita. Automatic summarization of Twitter topics. In *National Workshop on Design and Analysis of Algorithms*, 2010. (Cited on pages 4, 23, and 24).

137. Deepika Sharma. Article: Stemming algorithms: A comparative study and their analysis. *International Journal of Applied Information Systems*, 4(3):7–12, September 2012. Published by Foundation of Computer Science, New York, USA. (Cited on page 28).

138. Kim Sneppen and Mark E.J. Newman. Coherent noise, scale invariance and intermittency in large systems. *Physica D: Nonlinear Phenomena*, 110(3):209 – 222, 1997. (Cited on page 18).

139. Pawel Sobkowicz, Mike Thelwall, Kevan Buckley, Georgios Paltoglou, and An-

toni Sobkowicz. Lognormal distributions of user post lengths in internet discussions - a consequence of the Weber-Fechner law? *EPJ Data Science*, 2(1):2, Feb 2013. (Cited on page 30).

140. Connie St Louis and Gozde Zorlu. Can Twitter predict disease outbreaks? *BMJ*, 344, 2012. (Cited on page 109).

141. Statista. Twitter: number of monthly active users 2010-2017. `https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/`, 2018. Accessed: 2018-04-02. (Cited on page 2).

142. Stefan Stieglitz, Milad Mirbabaie, Bjorn Ross, and Christoph Neuberger. Social media analytics - challenges in topic discovery, data collection, and data preparation. *International Journal of Information Management*, 39:156 – 168, 2018. (Cited on page 4).

143. Stephen D. Strowes. Diurnal and weekly cycles in IPv6 traffic. *CoRR*, abs/1607.05183, 2016. (Cited on page 31).

144. Eric Sun, Itamar Rosenn, Cameron Marlow, and Thomas Lento. Gesundheit! modeling contagion through Facebook news feed. *AAAI*, 2009. (Cited on pages 13, 23, and 65).

145. Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014. (Cited on page 36).

146. Taro Takaguchi. Analyzing dynamical social interactions as temporal networks. *IFAC-PapersOnLine*, 48(18):169 – 174, 2015. 4th IFAC Conference on Analysis and Control of Chaotic Systems CHAOS 2015. (Cited on page 15).

147. Yuki Takeichi, Kazutoshi Sasahara, Reiji Suzuki, and Takaya Arita. Concurrent bursty behavior of social sensors in sporting events. *PLoS ONE*, 10(12):1–13, 12 2015. (Cited on pages 3, 24, and 110).

148. Edson Tandoc and Erika Johnson. Most students get breaking news first from Twitter. *Newspaper Research Journal*, 37(2):153–166, 2016. (Cited on page 43).

149. Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *EMNLP*, 2015. (Cited on page 29).

150. Alex Thompson. Journalists and trump voters live in separate online bubbles, mit analysis shows. *Vice News*. (Accessed on 04/25/2018). (Cited on page 46).

151. Chris Tofallis. A better measure of relative prediction accuracy for model selection and model estimation. *Journal of the Operational Research Society*, 66(8):1352–1362, Aug 2015. (Cited on page 34).

152. Twingly. News articles from 180 countries. `https://www.twingly.com/news-data/`, March 2018. Accessed: 2018-03-16. (Cited on page 2).

153. Twitter. The Twitter API, Jan 2017. `https://dev.twitter.com/`. (Cited on pages 2, 43, 65, and 67).

154. Twitter. Twitter terms of service. `https://twitter.com/en/tos`, 2017. (Cited on page 4).

155. Sonja Utz. Is LinkedIn making you more successful? The informational benefits derived from public social media. *New Media & Society*, 18(11):2685–2702, 2016. (Cited on page 2).

156. Rik van Noord, Florian A. Kunneman, and Antal van den Bosch. *Predicting Civil Unrest by Categorizing Dutch Twitter Events*, pages 3–16. Springer International Publishing, Cham, 2017. (Cited on page 109).

157. Onur Varol, Emilio Ferrara, Clayton A. Davis, Filippo Menczer, and Alessandro Flammini. Online human-bot interactions: Detection, estimation, and characterization. In *ICWSM*, 2017. (Cited on page 4).

158. Alexei Vázquez, João Gama Oliveira, Zoltán Dezsö, Kwang-Il Goh, Imre Kondor, and Albert-László Barabási. Modeling bursts and heavy tails in human dynamics. *Physical Review E*, 73:036127, March 2006. (Cited on pages 18 and 19).

159. Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018. (Cited on page 21).

160. Soroush Vosoughi, Prashanth Vijayaraghavan, and Deb Roy. Tweet2Vec: Learning tweet embeddings using character-level CNN-LSTM encoder-decoder. In *SIGIR*, 2016. (Cited on pages 4, 26, and 29).

161. Jakob Voss. Measuring Wikipedia. In *Proceedings International Conference of the International Society for Scientometrics and Informetrics: 10th*, 2005. (Cited on page 30).

162. Dong Wang, Tarek Abdelzaher, and Lance Kaplan. Front matter. In *Social Sensing*, pages i – ii. Morgan Kaufmann, Boston, 2015. (Cited on page 4).

163. Karol Wegrzycki, Piotr Sankowski, Andrzej Pacuk, and Piotr Wygocki. Why do cascade sizes follow a power-law? *CoRR*, abs/1702.05913, 2017. (Cited on page 17).

164. Xuetao Wei, Nicholas Valler, B. Aditya Prakash, Iulian Neamtiu, Michalis Faloutsos, and Christos Faloutsos. Competing memes propagation on networks: A case study of composite networks. *SIGCOMM Computer Communication Review*, 42(5):5–12, September 2012. (Cited on page 14).

165. David Weigel. Cruz: National Enquirer story is 'garbage' from 'Donald Trump and his henchmen', March 2016. `https://www.washingtonpost.com/news/post-politics/wp/2016/03/25/cruz-national-enquirer-story-is-garbage-from-donald-trump/`. (Cited on page 54).

166. Shirley A. Williams, Melissa M. Terras, and Claire Warwick. What do people study when they study Twitter? Classifying Twitter related academic papers. *Journal of Documentation*, 69(3):384–410, 2013. (Cited on page 2).

167. Cort J. Willmott and Kenji Matsuura. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, 30:79–82, 2005. (Cited on page 33).

168. Jiyoung Woo, Jaebong Son, and Hsinchun Chen. An SIR model for violent topic diffusion in social media. In *Proceedings of 2011 IEEE International Conference on Intelligence and Security Informatics*, pages 15–19, July 2011. (Cited on page 13).

169. Bo Wu, Wen-Huang Cheng, Yongdong Zhang, and Tao Mei. Time matters: Multi-scale temporalization of social media popularity. In *Proceedings of the 2016 ACM on Multimedia Conference*, MM '16, pages 1336–1344, New York, NY, USA, 2016. ACM. (Cited on page 20).

170. Bo Wu and Haiying Shen. Analyzing and predicting news popularity on Twitter. *International Journal of Information Management*, 35(6):702–711, December

2015. (Cited on pages 11 and 23).

171. Shaomei Wu, Jake M. Hofman, Winter A. Mason, and Duncan J. Watts. Who says what to whom on Twitter. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 705–714, New York, NY, USA, 2011. (Cited on page 13).

172. Hao Xiaoming, Wen Nainan, and Cherian George. The impact of online news consumption on young people's political participation. *International Journal of Information Management*, 5(2):16–31, April 2014. (Cited on page 43).

173. Wei Xie, Feida Zhu, Jing Jiang, Ee-Peng Lim, and Ke Wang. TopicSketch: Real-time bursty topic detection from Twitter. *IEEE Transactions on Knowledge & Data Engineering*, 28(8):2216–2229, Aug. 2016. (Cited on pages 12 and 25).

174. Dongkuan Xu and Yingjie Tian. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193, Jun 2015. (Cited on pages 27, 119, and 160).

175. Xin Yan and Xiao Gang Su. *Linear Regression Analysis: Theory and Computing*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2009. (Cited on page 36).

176. Jaewon Yang and Jure Leskovec. Modeling information diffusion in implicit networks. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pages 599–608, Washington, DC, USA, 2010. IEEE Computer Society. (Cited on pages 13, 23, and 65).

177. Jaewon Yang and Jure Leskovec. Patterns of temporal variation in online media. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 177–186, New York, NY, USA, 2011. ACM. (Cited on page 24).

178. George Udny Yule. A mathematical theory of evolution, based on the conclusions of Dr. J. C. Willis, F.R.S. *Royal Society of London Philosophical Transactions Series B*, 213:21–87, 1925. (Cited on page 18).

179. Tieling Zhang and Min Xie. Failure data analysis with extended Weibull distribution. *Communications in Statistics - Simulation and Computation*, 36(3):579–592, 2007. (Cited on page 31).

180. Qingyuan Zhao, Murat A. Erdogdu, Hera Y. He, Anand Rajaraman, and Jure Leskovec. SEISMIC: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 1513–1522, New York, NY, USA, 2015. (Cited on pages 1, 3, 8, 15, 21, 22, 91, 98, 102, 105, and 139).

181. Siqi Zhao, Lin Zhong, Jehan Wickramasuriya, and Venu Vasudevan. Human as Real-Time Sensors of Social and Physical Events: A Case Study of Twitter and Sports Games. *ArXiv e-prints 1106.4300*, June 2011. (Cited on pages 3, 24, 30, and 110).

182. Changtao Zhong, Dmytro Karamshuk, and Nishanth Sastry. Predicting Pinterest: Automating a distributed human computation. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 1417–1426, Republic and Canton of Geneva, Switzerland, 2015. International World Wide Web Conferences Steering Committee. (Cited on page 2).

183. Zhenkun Zhou, Jichang Zhao, and Ke Xu. Can online emotions predict the stock market in China? In Wojciech Cellary, Mohamed F. Mokbel, Jianmin Wang, Hua Wang, Rui Zhou, and Yanchun Zhang, editors, *Web Information Systems Engineering – WISE 2016*, pages 328–342, Cham, 2016. Springer International Publishing. (Cited on page 5).

184. Linhong Zhu and Kristina Lerman. Attention inequality in social media. *ArXiv e-prints 1601.07200*, January 2016. (Cited on pages 20 and 98).

185. Yin Zhu, Erheng Zhong, Sinno Jialin Pan, Xiao Wang, Minzhe Zhou, and Qiang Yang. Predicting user activity level in social networks. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pages 159–168, New York, NY, USA, 2013. ACM. (Cited on page 21).